Albert Bifet · Tomas Krilavičius ·
Ioanna Miliou · Slawomir Nowaczyk (Eds.)

LNAI 14949

Machine Learning and Knowledge Discovery in Databases

Applied Data Science Track

European Conference, ECML PKDD 2024
Vilnius, Lithuania, September 9–13, 2024
Proceedings, Part IX

9 Part IX

ECML
PKDD
2024

Springer

MOREMEDIA

Lecture Notes in Computer Science

Lecture Notes in Artificial Intelligence 14949

Founding Editor

Jörg Siekmann

Series Editors

Randy Goebel, *University of Alberta, Edmonton, Canada*
Wolfgang Wahlster, *DFKI, Berlin, Germany*
Zhi-Hua Zhou, *Nanjing University, Nanjing, China*

The series Lecture Notes in Artificial Intelligence (LNAI) was established in 1988 as a topical subseries of LNCS devoted to artificial intelligence.

The series publishes state-of-the-art research results at a high level. As with the LNCS mother series, the mission of the series is to serve the international R & D community by providing an invaluable service, mainly focused on the publication of conference and workshop proceedings and postproceedings.

Albert Bifet · Tomas Krilavičius · Ioanna Miliou ·
Slawomir Nowaczyk
Editors

Machine Learning and Knowledge Discovery in Databases

Applied Data Science Track

European Conference, ECML PKDD 2024
Vilnius, Lithuania, September 9–13, 2024
Proceedings, Part IX

Springer

Editors
Albert Bifet
LTCI
Télécom Paris
Palaiseau Cedex, France

Tomas Krilavičius
Faculty of Informatics
Vytautas Magnus University
Akademija, Lithuania

Ioanna Miliou
University of Tartu
Kista, Sweden

Slawomir Nowaczyk
School of Information Technology
Halmstad University
Halmstad, Sweden

ISSN 0302-9743 ISSN 1611-3349 (electronic)
Lecture Notes in Artificial Intelligence
ISBN 978-3-031-70377-5 ISBN 978-3-031-70378-2 (eBook)
https://doi.org/10.1007/978-3-031-70378-2

LNCS Sublibrary: SL7 – Artificial Intelligence

© The Editor(s) (if applicable) and The Author(s), under exclusive license
to Springer Nature Switzerland AG 2024

This work is subject to copyright. All rights are solely and exclusively licensed by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.
The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.
The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

If disposing of this product, please recycle the paper.

Preface

The 2024 edition of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD 2024) was held in Vilnius, Lithuania, from September 9 to 13, 2024.

The annual ECML PKDD conference acts as a world-wide platform showcasing the latest advancements in machine learning and knowledge discovery in databases. Held jointly since 2001, ECML PKDD has established itself as the leading European Machine Learning and Data Mining conference. It offers researchers and practitioners an unparalleled opportunity to exchange knowledge and ideas about the latest technical advancements in these disciplines. Moreover, the conference appreciates the synergy between foundational advances and groundbreaking data science and hence strongly welcomes contributions about how Machine Learning and Data Mining is being employed to solve real-world challenges.

The conference continues to evolve reflecting evolving technological developments and societal needs. For example, in the Research Track this year there has been an increase in submissions on generative AI, especially LLMs, and various aspects of responsible AI.

We received 826 submissions for the Research Track and 224 for the Applied Data Science Track. The Research track accepted 202 papers (out of 826, 24.5%) and the Applied Data Science Track accepted 56 (out of 224, 24.5%). In addition, 31 papers from the Journal Track (accepted out of 65 submissions) and 14 Demo Track papers (accepted out of 30 submissions).

The papers presented over the three main conference days were organized into five distinct tracks:

Research Track: This track featured research and methodology papers spanning all branches within Machine Learning, Knowledge Discovery, and Data Mining.
Applied Data Science Track: Papers in this track focused on novel applications of machine learning, data mining, and knowledge discovery to address real-world challenges, aiming to bridge the gap between theory and practical implementation.
Journal Track: This track included papers that had been published in special issues of the journals *Machine Learning* and *Data Mining and Knowledge Discovery.*
Demo Track: Short papers in this track introduced new prototypes or fully operational systems that leverage data science techniques, demonstrated through working prototypes.
Nectar Track: Concise presentations of recent scientific advances published in related conferences or journals. It aimed to disseminate important research findings to a broader audience within the ECML PKDD community.

The conference featured five keynote talks on diverse topics, reflecting emerging needs like benchmarking and resource-awareness, as well as theoretical understanding and industrial needs.

- Gintarė Karolina Džiugaitė (Google DeepMind): *The Dynamics of Memorization and Unlearning.*
- Moritz Hardt (Max Planck Institute for Intelligent Systems): *The Emerging Science of Benchmarks.*
- Mounia Lalmas-Roelleke (Spotify): *Enhancing User Experience with AI-Powered Search and Recommendations at Spotify.*
- Patrick Lucey (Stats Perform): *How to Utilize (and Generate) Player Tracking Data in Sport.*
- Katharina Morik (TU Dortmund University): *Resource-Aware Machine Learning — a User-Oriented Approach.*

The ECML PKDD 2024 Organizing Committee supported Diversity and Inclusion by awarding some grants that enable early career researchers to attend the conference, present their research activities, and become part of the ECML PKDD community. We provided a total of 3 scholarships of €1000 to individuals that come from the developing countries and/or communities which are underrepresented in science and technology. The scholarships could be used for travel and accommodation. In addition 3 grants covering all of the registration fees were awarded to individuals who belong to underrepresented communities, based on gender and role/position, to attend the conference and present their research activities. The Diversity and Inclusion action also included the Women Networking event and Diversity and Inclusion Panel discussion. The Women Networking event aimed to create a safe and inclusive space for networking and reflecting on the experience of women in science. The event included a structured brainstorm/reflection on the role and experience of women in science and technology, which will be published in the conference newsletter. The Diversity and Inclusion Panel aimed to reach a wider audience and encourage the discussion on the need for diversity in tech, and challenges and solutions in achieving it.

We want to thank the authors, workshop and tutorial organizers, and participants whose scientific contributions make this such an exciting event. Moreover, putting together an outstanding conference program would also not be possible without the dedication and (substantial) time investments of the area chairs, program committee, and organizing committee. The event would not run smoothly without the many volunteers and sessions chairs. Finally, we want to extend a special thanks to all the local organizers – they dealt with all the little details that are needed to make the conference a memorable event.

We want to extend our heartfelt gratitude to our wonderful sponsors for their generous financial support. We also want to thank Springer for their continuous support and Microsoft for allowing us to use their CMT software for conference management and providing help throughout. We very much appreciate the advice and guidance provided by the ECML PKDD Steering Committee over the past two years. Finally, we thank the organizing institution, the Artificial Intelligence Association of Lithuania.

September 2024

Albert Bifet
Tomas Krilavičius
Eirini Ntoutsi
Indrė Žliobaitė
Jesse Davis
Meelis Kull
Ioanna Miliou
Slawomir Nowaczyk

Organization

General Chairs

Albert Bifet	IP Paris, France/University of Waikato, New Zealand
Tomas Krilavičius	Vytautas Magnus University, Lithuania

Research Track Program Chairs

Indrė Žliobaitė	University of Helsinki, Finland
Meelis Kull	University of Tartu, Estonia
Jesse Davis	KU Leuven, Belgium
Eirini Ntoutsi	University of the Bundeswehr Munich, Germany

Applied Data Science Track Program Chairs

Slawomir Nowaczyk	Halmstad University, Sweden
Ioanna Miliou	Stockholm University, Sweden

Journal Track Chairs

Panagiotis Papapetrou	Stockholm University, Sweden
Rita Ribeiro	University of Porto/LIAAD, Portugal
Myra Spiliopoulou	Otto-von-Guericke University Magdeburg, Germany
Šarūnas Girdzijauskas	KTH Royal Institute of Technology, Sweden

Local Chair

Linas Petkevičius	Vilnius University, Lithuania

Workshop and Tutorial Chairs

Mantas Lukoševičius	Kaunas University of Technology, Lithuania
Mykola Pechenizkiy	Technische Universiteit Eindhoven, the Netherlands

Demo Chairs

Povilas Daniušis	Vytautas Magnus University, Lithuania
Kai Puolamäki	University of Helsinki, Finland

Proceedings Chairs

Wouter Duivesteijn	Technische Universiteit Eindhoven, the Netherlands
Rianne Schouten	Technische Universiteit Eindhoven, the Netherlands

PhD Forum Chairs

Virginijus Marcinkevičius	Vilnius University, Lithuania
Simona Ramanauskaitė	Vilnius Tech, Lithuania

Discovery Track Chairs

Peter van der Putten	Universiteit Leiden, the Netherlands
Jan N. van Rijn	Universiteit Leiden, the Netherlands

Workshop Proceedings Chairs

Danguole Kalinauskaite	Vytautas Magnus University, Lithuania
Kristina Šutiene	Kaunas Technology University, Lithuania

Social Media and Web Chairs

Julija Vaitonytė	Tilburg University, the Netherlands
Kamilė Dementavičiūtė	Vilnius University, Lithuania

Sponsorship Chairs

Mariam Barry	BNP Paribas, France
Dalia Breskuvienė	Vilnius University, Lithuania
Daniele Apiletti	Politecnico di Torino, Italy

Diversity and Inclusion Chair

Rūta Binkytė-Sadauskienė	Inria, France

Industry Track Chairs

Pieter Van Hertum	ASML, the Netherlands
Bjoern Bringmann	Deloitte, Germany

Nectar Track Chairs

Heitor Murilo Gomes	Victoria University of Wellington, New Zealand
Jesse Read	École Polytechnique, France

Awards Chairs

Michele Sebag	CNRS, France
João Gama	University of Porto, Portugal

ECML PKDD Steering Committee

Tijl De Bie	Ghent University, Belgium
Francesco Bonchi	ISI Foundation, Italy
Albert Bifet	Télécom ParisTech, France

Andrea Passerini	University of Trento, Italy
Katharina Morik	TU Dortmund, Germany
Arno Siebes	Utrecht University, the Netherlands
Sašo Džeroski	Jožef Stefan Institute, Slovenia
Robert Jan van Wijk	ASML, the Netherlands
Ilaria Bordino	UniCredit, Italy
Siegfried Nijssen	Université catholique de Louvain, Belgium
Albrecht Zimmermann	University of Caen - Normandie, France
Annalisa Appice	University of Bari ‘Aldo Moro’, Italy
Tania Cerquitelli	Politecnico di Torino, Italy
Alípio Jorge	University of Porto, Portugal
Fernando Perez-Cruz	ETH Zurich, Switzerland
Massih-Reza Amini	University Grenoble Alpes, France
Peggy Cellier	INSA Rennes, IRISA, France
Tias Guns	KU Leuven, Belgium
Grigorios Tsoumakas	Aristote University of Thessaloniki, Greece
Elena Baralis	Politecnico di Torino, Italy
Claudia Plant	Universität Wien, Austria
Manuel Gomez Rodriguez	Max Planck Institute for Software Systems, Germany

Program Committees

Guest Editorial Board, Journal Track

Richard Allmendringer	University of Manchester, UK
Marie Anastacio	Leiden University, the Netherlands
Giuseppina Andresini	Università degli Studi di Bari ‘Aldo Moro’, Italy
Annalisa Appice	Università degli Studi di Bari ‘Aldo Moro’, Italy
Jaume Bacardit	Newcastle University, UK
Maria Bampa	Stockholm University, Sweden
Mitra Baratchi	LIACS - University of Leiden, the Netherlands
Szymon Bobek	Jagiellonian University, Poland
Claudio Borile	CENTAI Institute, Italy
Falko Brause	University of Vienna, Austria
Barbara Catania	University of Genoa, Italy
Michelangelo Ceci	University of Bari, Italy
Loïc Cerf	Universidade Federal de Minas Gerais, Brazil
Tianyi Chen	Boston University, USA
Filip Cornell	KTH Royal Institute of Technology, Sweden

Marco Cotogni	University of Pavia, Italy
Claudia Diamantini	Università Politecnica delle Marche, Italy
Sebastien Destercke	UTC, France
César Ferri	Universitat Politécnica Valéncia, Spain
Olga Fink	EPFL, Switzerland
Esther Galbrun	University of Eastern Finland, Finland
Joao Gama	INESC TEC - LIAAD, Portugal
Jose A. Gamez	Universidad de Castilla-La Mancha, Spain
Paolo Garza	Politecnico di Torino, Italy
Carolina Geiersbach	Weierstrass Institute Berlin, Germany
Riccardo Guidotti	University of Pisa, Italy
Francesco Gullo	UniCredit, Italy
Martin Holena	Institute of Computer Science, Czechia
Dino Ienco	INRAE, France
Georgiana Ifrim	University College Dublin, Ireland
Felix Iglesias	Technical University of Vienna, Austria
Angelo Impedovo	University of Bari 'Aldo Moro', Italy
Matthias Jacobs	Technical University Dortmund, Germany
Szymon Jaroszewicz	Polish Academy of Sciences, Poland
Yifei Jin	Ericsson Research/KTH Royal Institute of Technology, Sweden
Panagiotis Karras	University of Copenhagen, Denmark
Mehdi Kaytoue	Infologic R&D, France
Dragi Kocev	Josef Stefan Institute, Slovenia
Helge Langseth	Norwegian Univ of Science and Technology, Norway
Thien Le	MIT, USA
Hsuan-Tien Lin	National Taiwan University, Taiwan
Marco Lippi	University of Modena and Reggio Emilia, Italy
Corrado Loglisci	Università degli Studi di Bari 'Aldo Moro', Italy
Brian Mac Namee	University College Dublin, Ireland
Sindri Magnusson	Stockholm University, Sweden
Giuseppe Manco	ICAR-CNR, Italy
Michael Mathioudakis	University of Helsinki, Finland
Ioanna Miliou	Stockholm University, Sweden
Olof Mogren	RISE Research Institutes, Sweden
Nuno Moniz	University of Notre Dame, France
Anna Monreale	University of Pisa, Italy
Alberto Montresor	University of Trento, Italy
Katharina Morik	Technical University Dortmund, Germany
Lia Morra	Politecnico di Torino, Italy
Amedeo Napoli	LORIA, Nancy, France

Andrea Paudice	University of Milan, Italy
Benjamin Noack	Otto-von-Guericke University Magdeburg, Germany
Slawomir Nowaczyk	Halmstad University, Sweden
Vincenzo Pasquadibisceglie	Università degli Studi di Bari 'Aldo Moro', Italy
Ruggero G. Pensa	University of Turin, Italy
Linas Petkevicius	Vilnius University, Finland
Marc Plantevit	EPITA, France
Kai Puolamäki	University of Helsinki, Finland
Jan Ramon	Inria, France
Matteo Riondato	Amherst College, USA
Isak Samsten	Stockholm University, Sweden
Shinichi Shirakawa	Yokohama National University, Japan
Amira Soliman	Halmstad University, Sweden
Fabian Spaeh	Boston University, USA
Gerasimos Spanakis	Maastricht University, the Netherlands
Mahito Sugiyama	National Institute of Informatics, Japan
Nikolaj Tatti	Helsinki University, Finland
Josephine Thomas	University of Kassel, Germany
Sebastian Stober	Otto-von-Guericke University Magdeburg, Germany
Genoveva Vargas-Solar	CNRS LIRIS, France
Bruno Veloso	University of Porto, Portugal
Pascal Welke	Technical University of Vienna, Austria
Marcel Wever	Ludwig-Maximilian-University Munich, Germany
Ye Zhu	Deakin University, Australia
Albrecht Zimmermann	Université de Caen Normandie, France
Blaz Zupan	University of Ljubljana, Slovenia

Area Chairs, Research Track

Leman Akoglu	CMU, USA
Anthony Bagnall	University of Southampton, UK
Gustavo Batista	UNSW, Australia
Jessa Bekker	KU Leuven, Belgium
Bettina Berendt	TU Berlin, Germany
Hendrik Blockeel	KU Leuven, Belgium
Henrik Bostrom	KTH Royal Institute of Technology, Sweden
Zied Bouraoui	CRIL CNRS & Univ Artois, France
Ulf Brefeld	Leuphana, Germany

Toon Calders	Universiteit Antwerpen, Belgium
Michelangelo Ceci	University of Bari, Italy
Fabrizio Costa	Exeter University, UK
Tijl De Bie	Ghent University, Belgium
Tom Diethe	AstraZeneca, UK
Kurt Driessens	Maastricht University, the Netherlands
Wouter Duivesteijn	TU Eindhoven, the Netherlands
Sebastijan Dumancic	TU Delft, the Netherlands
Tapio Elomaa	Tampere University, Finland
Stefano Ferilli	University of Bari, Italy
Cèsar Ferri	Universitat Politècnica València, Spain
Peter Flach	University of Bristol, UK
Elisa Fromont	Université Rennes 1, IRISA/Inria rba, France
Johannes Fürnkranz	JKU Linz, Austria
Esther Galbrun	University of Eastern Finland, Finland
Joao Gama	INESC TEC - LIAAD, Portugal
Aristides Gionis	KTH Royal Institute of Technology, Sweden
Bart Goethals	Universiteit Antwerpen, Belgium
Chen Gong	Nanjing University of Science and Technology, China
Dimitrios Gunopulos	University of Athens, Greece
Tias Guns	KU Leuven, Belgium
Barbara Hammer	CITEC, Bielefeld University, Germany
José Hernández-Orallo	Universitat Politècnica de València, Spain
Sibylle Hess	TU Eindhoven, the Netherlands
Andreas Hotho	University of Wuerzburg, Germany
Eyke Hüllermeier	University of Munich, Germany
Georgiana Ifrim	University College Dublin, Ireland
Manfred Jaeger	Aalborg University, Denmark
Szymon Jaroszewicz	Polish Academy of Sciences, Poland
George Karypis	University of Minnesota, Twin Cities, USA
Ioannis Katakis	University of Nicosia, Cyprus
Marius Kloft	TU Kaiserslautern, Germany
Dragi Kocev	Jožef Stefan Institute, Slovenia
Parisa Kordjamshidi	Michigan State University, USA
Lars Kotthoff	University of Wyoming, USA
Petra Kralj Novak	Central European University, Austria
Georg Krempl	Utrecht University, the Netherlands
Peer Kröger	Christian-Albrechts-Universität Kiel, Germany
Leo Lahti	University of Turku, Finland
Mark Last	Ben-Gurion University of the Negev, Israel
Jefrey Lijffijt	Ghent University, Belgium

Jessica Lin	George Mason University, USA
Michele Lombardi	University of Bologna, Italy
Donato Malerba	Università degli Studi di Bari ‘Aldo Moro’, Italy
Fragkiskos Malliaros	CentraleSupelec, France
Giuseppe Marra	KU Leuven, Belgium
Wannes Meert	KU Leuven, Belgium
Ernestina Menasalvas	Universidad Politècnica de Madrid, Spain
Pauli Miettinen	University of Eastern Finland, Finland
Dunja Mladenic	Jozef Stefan Institute, Slovenia
Emmanuel Müller	TU Dortmund, Germany
Siegfried Nijssen	Université catholique de Louvain, Belgium
Symeon Papadopoulos	Information Technologies Institute/Centre for Research & Technology - Hellas, Greece
Evangelos Papalexakis	UC Riverside, USA
Andrea Passerini	University of Trento, Italy
Jaakko Peltonen	Tampere University, Finland
Bernhard Pfahringer	University of Waikato, New Zealand
Claudia Plant	University of Vienna, Austria
Ricardo Prudencio	Universidade Federal de Pernambuco, Brazil
Milos Radovanovic	U. Novi Sad, Serbia
Chedy Raissi	Inria, France
Jesse Read	Ecole Polytechnique, France
Celine Robardet	INSA Lyon, France
Salvatore Ruggieri	University of Pisa, Italy
Steven Schockaert	Cardiff University, Wales, UK
Matthias Schubert	Ludwig-Maximilians-Universität München, Germany
Thomas Seidl	LMU Munich, Germany
Arno Siebes	Universiteit Utrecht, the Netherlands
Fabrizio Silvestri	Sapienza University of Rome, Italy
Jerzy Stefanowski	Poznan University of Technology, Poland
Nikolaj Tatti	Helsinki University, Finland
Evimaria Terzi	Boston University, USA
Grigorios Tsoumakas	Aristotle University of Thessaloniki, Greece
Charalampos Tsourakakis	Boston University, USA
Matthijs van Leeuwen	Leiden University, the Netherlands
Jan Van Rijn	LIACS, Leiden University, the Netherlands
Celine Vens	KU Leuven, Belgium
Jilles Vreeken	CISPA Helmholtz Center for Information Security, Germany
Willem Waegeman	Universiteit Gent, Belgium
Wei Ye	Tongji University, China

Wenbin Zhang	Florida International University, USA
Arthur Zimek	University of Southern Denmark, Denmark
Albrecht Zimmermann	Université de Caen Normandie, France

Area Chairs, Applied Data Science Track

Annalisa Appice	University of Bari ‘Aldo Moro’, Italy
Sahar Asadi	King (Microsoft), Sweden
Martin Atzmueller	Osnabrück University & DFKI, Germany
Michael R. Berthold	KNIME, Germany
Michelangelo Ceci	University of Bari, Italy
Peggy Cellier	INSA Rennes, IRISA, France
Nicolas Courty	IRISA, Université Bretagne-Sud, France
Bruno Cremilleux	Université de Caen Normandie, France
Tom Diethe	AstraZeneca, UK
Dejing Dou	BCG, USA
Olga Fink	EPFL, Switzerland
Elisa Fromont	Université Rennes 1, IRISA/Inria rba, France
Johannes Fürnkranz	JKU Linz, Austria
Sreenivas Gollapudi	Google, USA
Andreas Hotho	University of Wuerzburg, Germany
Alipio M. G. Jorge	INESC TEC/University of Porto, Portugal
George Karypis	University of Minnesota, Minneapolis, USA
Yun Sing Koh	University of Auckland, New Zealand
Parisa Kordjamshidi	Michigan State University, USA
Niklas Lavesson	Blekinge Institute of Technology, Sweden
Chuan Lei	Amazon, USA
Thomas Liebig	TU Dortmund Artificial Intelligence Unit, Germany
Tony Lindgren	Stockholm University, Sweden
Patrick Loiseau	Inria, France
Giuseppe Manco	ICAR-CNR, Italy
Gabor Melli	PredictionWorks, USA
Ioanna Miliou	Stockholm University, Sweden
Anna Monreale	University of Pisa, Italy
Luis Moreira-Matias	sennder, Germany
Jian Pei	Simon Fraser University, Canada
Fabio Pinelli	IMT Lucca, Italy
Zhiwei (Tony) Qin	Lyft, USA
Visvanathan Ramesh	Independent Researcher, Germany
Fabrizio Silvestri	Sapienza, University of Rome, Italy

Liang Sun	Alibaba Group, China
Jiliang Tang	Michigan State University, USA
Sandeep Tata	Google, USA
Yinglong Xia	Meta, USA
Fuzhen Zhuang	Institute of Artificial Intelligence, Beihang University, China
Albrecht Zimmermann	Université de Caen Normandie, France

Program Committee Members, Research Track

Zahraa Abdallah	University of Bristol, UK
Ziawasch Abedjan	TU Berlin, Germany
Koren Abitbul	Ben-Gurion University, Israel
Timilehin Aderinola	Insight SFI Research Centre for Data Analytics, University College Dublin, Ireland
Homayun Afrabandpey	Nokia Technologies, Finland
Reza Akbarinia	Inria, France
Esra Akbas	Georgia State University, USA
Cuneyt Akcora	University of Central Florida, USA
Youhei Akimoto	University of Tsukuba/RIKEN AIP, Japan
Ozge Alacam	University of Bielefeld, Germany
Amr Alkhatib	KTH Royal Institute of Technology, Sweden
Mari-Liis Allikivi	University of Tartu, Estonia
Ranya Almohsen	West Virginia University, USA
Jose Alvarez	Scuola Normale Superiore, Italy
Ehsan Aminian	INESC TEC, Portugal
Christos Anagnostopoulos	University of Glasgow, UK
James Anderson	Columbia University, USA
Thiago Andrade	INESC TEC/University of Porto, Portugal
Jean-Marc Andreoli	Naverlabs Europe, France
Giuseppina Andresini	University of Bari ‘Aldo Moro’, Italy
Simone Angarano	Politecnico di Torino, Italy
Akash Anil	Cardiff University, Wales, UK
Ekaterina Antonenko	Mines Paris - PSL, France
Alessandro Antonucci	IDSIA, Switzerland
Edward Apeh	Bournemouth University, UK
Nikhilanand Arya	Indian Institute of Technology, Patna, India
Saeed Asadi Bagloee	University of Melbourne, Australia
Ali Ayadi	University of Strasbourg, France
Steve Azzolin	University of Trento, Italy
Lilian Berton	Universidade Federal de Sao Paulo, Brazil

Florian Babl	Universität der Bundeswehr München, Germany
Michael Bain	University of New South Wales, Australia
Chandrajit Bajaj	University of Texas, Austin, USA
Bunil Balabantaray	NIT Meghalaya, India
Federico Baldo	University of Bologna, Italy
Georgia Baltsou	Information Technologies Institute/Centre for Research & Technology - Hellas, Greece
Hubert Baniecki	University of Warsaw, Poland
Mitra Baratchi	LIACS - University of Leiden, the Netherlands
Francesco Bariatti	Univ Rennes, CNRS, IRISA, France
Franka Bause	University of Vienna, Austria
Florian Beck	JKU Linz, Austria
Jacob Beck	LMU Munich, Germany
Rita Beigaite	VTT, Finland
Michael Beigl	Karlsruhe Institute of Technology, Germany
Diana Benavides Prado	University of Auckland, New Zealand
Andreas Bender	LMU Munich, Germany
Idir Benouaret	Epita Research Laboratory, France
Gilberto Bernardes	INESC TEC & University of Porto, Faculty of Engineering, Portugal
Jolita Bernatavičienė	Vilnius University, Lithuania
Cuissart Bertrand	University of Caen, France
Eva Besada-Portas	Universidad Complutense de Madrid, Spain
Jalaj Bhandari	Columbia University, USA
Monowar Bhuyan	Umea University, Sweden
Manuele Bicego	University of Verona, Italy
Przemyslaw Biecek	Warsaw University of Technology, Poland
Albert Bifet	Telecom Paris, France
Livio Bioglio	University of Turin, Italy
Anton Björklund	University of Helsinki, Finland
Szymon Bobek	Jagiellonian University, Poland
Ludovico Boratto	University of Cagliari, Italy
Stefano Bortoli	Huawei Research Center
Annelot Bosman	Universiteit Leiden, the Netherlands
Tassadit Bouadi	Université de Rennes, France
Hamid Bouchachia	Bournemouth University, UK
Jannis Brugger	TU Darmstadt, Germany
Dariusz Brzezinski	Poznan University of Technology, Poland
Maria Sofia Bucarelli	Sapienza University of Rome, Italy
Mirko Bunse	TU Dortmund University, Germany
Tomasz Burzykowski	Hasselt University, Belgium

Sebastian Buschjäger	TU Dortmund Artificial Intelligence Unit, Germany
Maarten Buyl	Ghent University, Belgium
Zaineb Chelly Dagdia	UVSQ, Paris-Saclay, France
Huaming Chen	University of Sydney, Australia
Xiaojun Chen	Institute of Information Engineering, CAS, China
Tobias Callies	Universtiät der Bundeswehr München, Germany
Xiaofeng Cao	University of Technology Sydney, Australia
Cécile Capponi	Aix-Marseille University, France
Lorenzo Cascioli	KU Leuven, Belgium
Guilherme Cassales	University of Waikato, New Zealand
Giovanna Castellano	University of Bari ‘Aldo Moro’, Italy
Andrea Cavallo	Delft University of Technology, the Netherlands
Remy Cazabet	Lyon, France
Antanas Čenys	Vilnius Gediminas Technical University, Lithuania
Mattia Cerrato	JGU Mainz, Germany
Ricardo Cerri	Federal University of Sao Carlos, Brazil
Prithwish Chakraborty	IBM Corporation
Harry Kai-Ho Chan	University of Sheffield, UK
Laetitia Chapel	IRISA, France
Victor Charpenay	Mines Saint-Etienne, France
Arthur Charpentier	UQAM, Canada
Chunchun Chen	Tongji University, China
Huiping Chen	University of Birmingham, UK
Jin Chen	Hong Kong University of Science and Technology, China
Kuan-Hsun Chen	University of Twente, the Netherlands
Lingwei Chen	Wright State University, USA
Minyu Chen	Shanghai Jiaotong University, China
Xuefeng Chen	Chongqing University, China
Ying Chen	RMIT University, Australia
Zheng Chen	Osaka University, Japan
Zhong Chen	Southern Illinois University, USA
Ziheng Chen	Walmart, USA
Zehua Cheng	University of Oxford, UK
Hua Chu	Xidian University, China
Oana Cocarascu	King’s College London, UK
Johanne Cohen	LISN-CNRS, France
Lidia Contreras-Ochando	Universitat Politècnica de València, Spain
Denis Coquenet	IRISA, France
Luca Corbucci	University of Pisa, Italy

Roberto Corizzo	American University, USA
Nathan Cornille	KU Leuven, Belgium
Baris Coskunuzer	University of Texas at Dallas, USA
Andrea Cossu	University of Pisa, Italy
Tiago Cunha	Expedia Group, Portugal
Florence d'Alché-Buc	Télécom Paris, France
Sebastian Dalleiger	KTH Royal Institute of Technology, Sweden
Robertas Damaševičius	Vytautas Magnus University, Lithuania
Xuan-Hong Dang	IBM T.J Watson Research Center, USA
Thi-Bich-Hanh Dao	University of Orleans, France
Paul Davidsson	Malmö University, Sweden
Jasper de Boer	KU Leuven, Belgium
Andre de Carvalho	USP, Brazil
Graziella De Martino	University of Bari 'Aldo Moro', Italy
Lennert De Smet	KU Leuven, Belgium
Marcilio de Souto	LIFO/Univ. Orleans, France
Julien Delaunay	Inria, France
Emanuele Della Valle	Politecnico di Milano, Italy
Pieter Delobelle	KU Leuven, Belgium
Vincent Derkinderen	KU Leuven, Belgium
Guillaume Derval	UCLouvain - ICTEAM, Belgium
Sebastien Destercke	UTC, France
Laurens Devos	KU Leuven, Belgium
Bhaskar Dhariyal	University College Dublin, Ireland
Davide Di Pierro	Università degli Studi di Bari, Italy
Yiqun Diao	National University of Singapore, Singapore
Lucile Dierckx	Université catholique de Louvain, Belgium
Anastasia Dimou	KU Leuven, Belgium
Jingtao Ding	Tsinghua University, China
Zifeng Ding	LMU Munich, Germany
Lamine Diop	EPITA, France
Christos Diou	Harokopio University of Athens, Greece
Alexander Dockhorn	Leibniz University Hannover, Germany
Stephan Doerfel	Kiel University of Applied Sciences, Germany
Hang Dong	University of Oxford, UK
Nanqing Dong	Shanghai Artificial Intelligence Laboratory, China
Emilio Dorigatti	LMU Munich, Germany
Haizhou Du	Shanghai University of Electric Power, China
Stefan Duffner	University of Lyon, France
Inês Dutra	University of Porto, Portugal
Anany Dwivedi	University of Waikato, New Zealand
Sofiane Ennadir	KTH Royal Institute of Technology, Sweden

Mark Eastwood	University of Warwick, UK
Vasilis Efthymiou	Harokopio University of Athens, Greece
Rémi Emonet	Unversité Saint-Etienne, France
Dominik Endres	Philipps-Universität Marburg, Germany
Eshant English	Hasso Plattner Institute, Germany
Bojan Evkoski	Central European University, Austria
Zipei Fan	University of Tokyo, Japan
Hadi Fanaee-T	Halmstad University, Germany
Fabio Fassetti	Universita della Calabria, Italy
Ad Feelders	Universiteit Utrecht, the Netherlands
Wenjie Feng	National University of Singapore, Singapore
Len Feremans	Universiteit Antwerpen, Belgium
Luca Ferragina	University of Calabria, Italy
Carlos Ferreira	INESC TEC, Portugal
Julien Ferry	LAAS-CNRS, France
Michele Fontana	Università di Pisa, Italy
Germain Forestier	University of Haute Alsace, France
Edouard Fouché	Karlsruhe Institute of Technology (KIT), Germany
Matteo Francobaldi	University of Bologna, Italy
Christian Frey	Fraunhofer IIS, Germany
Holger Froening	University of Heidelberg, Germany
Benoît Frénay	University of Namur, Belgium
Fabio Fumarola	Prometeia, Italy
Shanqing Guo	Shandong University, China
Claudio Gallicchio	University of Pisa, Italy
Shengxiang Gao	Kunming University of Science and Technology, China
Yifeng Gao	University of Texas Rio Grande Valley, USA
Manuel Garcia-Piqueras	Universidad de Castilla-La Mancha, Spain
Dario Garigliotti	University of Bergen, Norway
Damien Garreau	Université Côte d'Azur, France
Dominique Gay	Université de La Réunion, France
Alborz Geramifard	Meta, USA
Pierre Geurts	Montefiore Institute, University of Liège, Belgium
Alireza Gharahighehi	KU Leuven, Belgium
Siamak Ghodsi	Leibniz University of Hannover Free University Berlin, Germany
Shreya Ghosh	Penn State, USA
Vasilis Gkolemis	ATHENA RC, Greece
Dorota Glowacka	University of Helsinki, Finland
Heitor Gomes	Victoria University of Wellington, New Zealand

Wenwen Gong	Tsinghua University, China
Adam Goodge	I2R, A*STAR, Singapore
Anastasios Gounaris	Aristotle University of Thessaloniki, Greece
Brandon Gower-Winter	Utrecht University, the Netherlands
Michael Granitzer	University of Passau, Germany
Xinyu Guan	Xian Jiaotong University, China
Massimo Guarascio	ICAR-CNR, Italy
Riccardo Guidotti	University of Pisa, Italy
Dominique Guillot	University of Delaware, USA
Nuwan Gunasekara	AI Institute, University of Waikato, New Zealand
Thomas Guyet	Inria, Centre de Lyon, France
Vanessa Gómez-Verdejo	Universidad Carlos III de Madrid, Spain
Huong Ha	RMIT University, Australia
Benjamin Halstead	University of Auckland, New Zealand
Marwan Hassani	TU Eindhoven, the Netherlands
Yujiang He	University of Kassel, Germany
Edith Heiter	Ghent University, Belgium
Lars Hillebrand	Fraunhofer IAIS and University of Bonn, Germany
Martin Holena	Institute of Computer Science, Czechia
Mike Holenderski	Eindhoven University of Technology, the Netherlands
Hongsheng Hu	Data 61, CSIRO, Australia
Chao Huang	University of Hong Kong, China
Denis Huseljic	University of Kassel, Germany
Julian Höllig	University of the Bundeswehr Munich, Germany
Dimitrios Iliadis	UGENT, Belgium
Dino Ienco	INRAE, France
Roberto Interdonato	CIRAD, France
Omid Isfahani Alamdari	University of Pisa, Italy
Elvin Isufi	TU Delft, the Netherlands
Giulio Jacucci	University of Helsinki, Finland
Kuk Jin Jang	University of Pennsylvania, USA
Inigo Jauregi Unanue	University of Technology Sydney, Australia
Renhe Jiang	University of Tokyo, Japan
Pengfei Jiao	Hangzhou Dianzi University, China
Yilun Jin	Hong Kong University of Science and Technology, China
Rūta Juozaitienė	Vytautas Magnus University, Lithuania
Joonas Jälkö	University of Helsinki, Finland
Mira Jürgens	Ghent University, Belgium

Vana Kalogeraki	Athens University of Economics and Business, Greece
Toshihiro Kamishima	Independent Researcher, Japan
Nikos Kanakaris	University of Southern California, USA
Sevvandi Kandanaarachchi	CSIRO, Australia
Bo Kang	Ghent University, Belgium
Jurgita Kapočiūtė-Dzikienė	Tilde SIA, University of Latvia, Tilde IT, Vytautas Magnus University, Lithuania
Maiju Karjalainen	University of Eastern Finland, Finland
Panagiotis Karras	University of Copenhagen, Denmark
Gjergji Kasneci	TU Munich, Germany
Panagiotis Kasnesis	University of West Attica, Greece
Dimitrios Katsaros	University of Thessaly, Greece
Natthawut Kertkeidkachorn	Japan Advanced Institute of Science and Technology (JAIST), Japan
Stefan Kesselheim	Forschungszentrum Jülich, Germany
Jaleed Khan	University of Oxford, UK
Adem Kikaj	KU Leuven, Belgium
Nadja Klein	University Alliance Ruhr and TU Dortmund, Germany
Tomas Kliegr	University of Economics Prague, Czechia
Astrid Klipfel	CRIL - UMR 8188, France
Simon Koop	Technische Universiteit Eindhoven, the Netherlands
Frederic Koriche	Univ. d'Artois, CRIL CNRS UMR 8188, France
Grazina Korvel	Vilnius University, Lithuania
Ana Kostovska	Jožef Stefan Institute, Slovenia
Stefan Kramer	Johannes Gutenberg University Mainz, Germany
Emmanouil Krasanakis	CERTH, Greece
Anna Krause	Universität Würzburg, Germany
Nils Kriege	University of Vienna, Austria
Ričardas Krikštolaitis	Vytautas Magnus University, Lithuania
Amer Krivosija	TU Dortmund, Germany
Paweł Ksieniewicz	Wrocław University of Science and Technology, Poland
Janne Kujala	University of Turku, Finland
Nitesh Kumar	Cardiff University, UK
Vivek Kumar	Universität der Bundeswehr München, Germany
Olga Kurasova	Vilnius University, Institute of Data Science and Digital Technologies, Lithuania
Marius Köppel	Johannes Gutenberg University Mainz, Germany
Antti Laaksonen	University of Helsinki, Finland
Ville Laitinen	University of Turku, Finland

Carlos Lamuela Orta	University of Helsinki, Finland
Johannes Langguth	Simula Research Laboratory, Norway
Helge Langseth	Norwegian University of Science and Technology, Norway
Martha Larson	Radboud University, the Netherlands
Anton Lautrup	University of Southern Denmark, Denmark
Aonghus Lawlor	University College Dublin, Ireland
Tuan Le	New Mexico State University, USA
Erwan Le Merrer	Inria, France
Thach Le Nguyen	University College Dublin, Ireland
Tai Le Quy	IU International University of Applied Sciences, Germany
Mustapha Lebbah	Paris Saclay University-Versailles, France
Yeon-Chang Lee	Ulsan National Institute of Science and Technology (UNIST), South Korea
Zed Lee	Stockholm University, Sweden
Mathieu Lefort	Univ. Lyon, France
Vincent Lemaire	Orange Innovation
Daniel Lemire	University of Quebec (TELUQ), Canada
Florian Lemmerich	University of Passau, Germany
Daphne Lenders	University of Antwerp, Belgium
Carson Leung	University of Manitoba, Canada
Dan Li	Sun Yat-Sen University, China
Gang Li	Deakin University, Australia
Mark Junjie Li	Shenzhen University, China
Mingxio Li	KU Leuven, Belgium
Nian Li	Tsinghua University, China
Peiyan Li	Ludwig Maximilian University of Munich, Germany
Shuai Li	University of Cambridge, UK and University of Tokyo, Japan and Tsinghua University, China
Tong Li	HKUST, China
Xiang Li	East China Normal University, China
Yinsheng Li	Fudan University, China
Yong Li	Huawei European Research Center, Germany
Zhixin Li	Guangxi Normal University, China
Zhuoqun Li	Louisiana State University, USA
Yuxuan Liang	Hong Kong University of Science and Technology, China
Nick Lim	University of Waikato, New Zealand
Jason Lines	Independent Researcher, UK
Piotr Lipinski	Institute of Computer Science, University of Wroclaw, Poland

Arunas Lipnickas	Kaunas University of Technology, Lithuania
Marco Lippi	University of Florence, Italy
Bin Liu	Chongqing University of Posts and Telecommunications, China
Fenglin Liu	University of Oxford, UK
Junze Liu	University of California, Irvine, USA
Li Liu	Chongqing University, China
Xu Liu	National University of Singapore, Singapore
Zihan Liu	Zhejiang University & Westlake University, China
Corrado Loglisci	Università degli Studi di Bari ‘Aldo Moro’, Italy
Antonio Longa	University of Trento, Italy
Marco Loog	Radboud University, the Netherlands
Ana Carolina Lorena	ITA, Brazil
Beatriz López	University of Girona, Spain
Tuwe Löfström	Jönköping University, Sweden
Pingchuan Ma	HKUST, China
Ziqiao Ma	University of Michigan, USA
Henryk Maciejewski	Wrocław University of Science and Technology, Poland
Michael Madden	National University of Ireland Galway, Ireland
Sindri Magnusson	Stockholm University, Sweden
Ajay Mahimkar	AT&T, USA
Cedric Malherbe	AstraZeneca, UK
Giuseppe Manco	ICAR-CNR, Italy
Domenico Mandaglio	DIMES Dept., University of Calabria, Italy
Justina Mandravickaitė	Vytautas Magnus University, Lithuania
Silviu Maniu	Université Grenoble Alpes, France
Naresh Manwani	International Institute of Information Technology, Hyderabad, India
Alexandru Mara	Ghent University, Belgium
Virginijus Marcinkevičius	Vilnius University, Lithuania
Timo Martens	KU Leuven, Belgium
Linas Martišauskas	Vytautas Magnus University, Lithuania
Fernando Martínez-Plumed	Universitat Politècnica de València, Spain
Koji Maruhashi	Fujitsu Research, Fujitsu Limited
Rytis Maskeliūnas	Polsl, Poland
Florent Masseglia	Inria, France
Antonio Mastropietro	Università di Pisa, Italy
Sarah Masud	LCS2, IIIT-D, India
Dalius Matuzevicius	Vilnius Gediminas Technical University, Lithuania
Chandresh Maurya	IBM Research, India

Wolfgang Mayer	University of South Australia, Australia
Giacomo Medda	University of Cagliari, Italy
Nida Meddouri	LRE-EPITA, France
Stefano Melacci	University of Siena, Italy
Alessandro Melchiorre	Johannes Kepler University Linz, Austria
Marco Mellia	Politecnico di Torino, Italy
Joao Mendes-Moreira	University of Porto, Portugal
Engelbert Mephu Nguifo	Université Clermont Auvergne, CNRS, LIMOS, France
Fabio Mercorio	University of Milan-Bicocca, Italy
Henning Meyerhenke	Humboldt-Universität zu Berlin, Germany
Matthew Middlehurst	University of Southampton, UK
Jan Mielniczuk	Polish Academy of Sciences, Poland
Paolo Mignone	University of Bari 'Aldo Moro', Italy
Matej Mihelčić	University of Zagreb, Croatia
Tsunenori Mine	Kyushu University, Japan
Pierre Monnin	Université Côte d'Azur, Inria, CNRS, I3S, France
Carlos Monserrat-Aranda	Universitat Politècnica de València, Spain
Raha Moraffah	Arizona State University, USA
Thomas Mortier	Ghent University, Belgium
Frank Mtumbuka	Cardiff University, Wales, UK
Koyel Mukherjee	Adobe Research, India
Mario Andrés Muñoz	University of Melbourne, Australia
Nikolaos Mylonas	Aristotle University of Thessaloniki, Greece
Tommi Mäklin	University of Helsinki, Finland
Felipe Kenji Nakano	KU Leuven, Belgium
Géraldin Nanfack	University of Concordia, Canada
Mirco Nanni	CNR-ISTI Pisa, Italy
Francesca Naretto	University of Pisa, Italy
Fateme Nateghi Haredasht	Stanford University, USA
Benjamin Negrevergne	Université PSL – Paris Dauphine, France
Matti Nelimarkka	University of Helsinki, Finland
Kim Thang Nguyen	LIG, University Grenoble-Alpes, France
Shiwen Ni	Shenzhen Institute of Advanced Technology (SIAT), Chinese Academy of Sciences, China
Mikko Niemi	City of Helsinki, Finland
Nikolaos Nikolaou	University College London, UK
Simona Nisticò	University of Calabria, Italy
Hao Niu	KDDI Research, Inc., Japan
Andreas Nuernberger	Magdeburg University, Germany
Claire Nédellec	INRAE, MaIAGE, France
Barry O'Sullivan	University College Cork, Ireland

Makoto Onizuka	Osaka University, Japan
Jose Oramas	University of Antwerp, IMEC-IDLab, Belgium
Luis Ortega Andrés	Autonomous University of Madrid, Spain
Latifa Oukhellou	IFSTTAR, France
Agne Paulauskaite-Taraseviciene	KTU, Artificial Intelligence Centre, Lithuania
Massimo Piccardi	University of Technology Sydney, Australia
Marc Plantevit	EPITA, France
Andrei Paleyes	University of Cambridge, UK
Emmanouil Panagiotou	Freie Universität Berlin, Germany
George Panagopoulos	University of Luxembourg
Pance Panov	Jozef Stefan Institute, Slovenia
Apostolos Papadopoulos	Aristotle University of Thessaloniki, Greece
Panagiotis Papapetrou	Stockholm University, Sweden
Francesco Parisi	University of Calabria, Italy
Abigail Parker	University of Helsinki, Finland
Antonio Parmezan	University of São Paulo, Brazil
Vincenzo Pasquadibisceglie	University of Bari ‘Aldo Moro’, Italy
Tatiana Passali	Aristotle University of Thessaloniki, Greece
Eliana Pastor	Politecnico di Torino, Italy
Anand Paul	Louisiana State University HSC, USA
Mykola Pechenizkiy	TU Eindhoven, the Netherlands
Yulong Pei	TU Eindhoven, the Netherlands
Nikos Pelekis	University of Piraeus, Greece
Leonardo Pellegrina	University of Padova, Italy
Charlotte Pelletier	Université de Bretagne du Sud, France
Antonio Pellicani	Università degli Studi di Bari ‘Aldo Moro’, Italy
Frédéric Pennerath	CentraleSupélec - LORIA, France
Ruggero Pensa	University of Torino, Italy
Lucas Pereira	Interactive Technologies Institute, LARSyS, Técnico Lisboa, Portugal
Pedro Pereira Rodrigues	University of Porto, Portugal
Miquel Perello-Nieto	University of Bristol, UK
Lorenzo Perini	KU Leuven, Belgium
Linas Petkevicius	Vilnius University, Lithuania
Ninh Pham	University of Auckland, New Zealand
Nico Piatkowski	Fraunhofer IAIS, Germany
Francesco Piccialli	Independent Researcher, Italy
Martin Pilát	Charles University, Czechia
Gianvito Pio	University of Bari, Italy
Darius Plonis	Vilnius Gediminas Technical University, Lithuania
Marco Podda	University of Pisa, Italy

Mirko Polato	University of Turin, Italy
Marco Polignano	Università di Bari, Italy
Giovanni Ponti	ENEA, Italy
Alexandru Popa	University of Bucharest, Romania
Fabrice Popineau	CentraleSupélec/LISN, France
Cedric Pradalier	GeorgiaTech Lorraine, France
Paul Prasse	University of Potsdam, Germany
Mahardhika Pratama	University of South Australia, Australia
Bardh Prenkaj	Sapienza University of Rome, Italy
Steven Prestwich	University College Cork, Ireland
Giulia Preti	CENTAI, Italy
Philippe Preux	Inria, France
Danil Provodin	TU Eindhoven, the Netherlands
Chiara Pugliese	ISTI Institute of National Research Council University of Pisa, Italy
Simon Puglisi	University of Helsinki, Finland
Andrea Pugnana	University of Pisa, Italy
Erasmo Purificato	Otto von Guericke University Magdeburg, Germany
Peter van der Putten	Leiden University, the Netherlands
Abdulhakim Qahtan	Utrecht University, the Netherlands
Kun Qian	Amazon, USA
Kallol Roy	University of Tartu, Estonia
Dimitrios Rafailidis	University of Thessaly, Greece
Muhammad Rajabinasab	University of Southern Denmark, Denmark
Chang Rajani	University of Helsinki, Finland
Simona Ramanauskaitė	Vilnius Gediminas Technical University, Lithuania
Jan Ramon	Inria, France
M. José Ramírez-Quintana	Technical University of Valencia, Spain
Rajeev Rastogi	Amazon, USA
Domenico Redavid	University of Bari, Italy
Luis Rei	Jožef Stefan Institute, Slovenia
Christoph Reinders	Leibniz University Hannover, Germany
Qianqian Ren	Heilongjiang University, China
Mina Rezaei	LMU Munich, Germany
Rita Ribeiro	Porto, Portugal
Matteo Riondato	Amherst College, USA
Simon Rittel	University of Vienna, Austria
Giuseppe Rizzo	Niuma s.r.l, Italy
Pieter Robberechts	KU Leuven, Belgium

Christophe Rodrigues	DVRC pôle universitaire Léonard de Vinci, France
Federica Rollo	UNIMORE, Italy
Luca Romeo	University of Macerata, Italy
Nicolas Roque dos Santos	University of São Paulo, Brazil
Céline Rouveirol	LIPN Univ. Sorbonne Paris Nord, France
Arjun Roy	Freie Universität Berlin, Germany
Krzysztof Rudaś	Institute of Computer Science, Polish Academy of Sciences, Poland
Allou Same	Université Gustave Eiffel, France
Oswaldo Solarte-Pabon	Universidad del Valle, Spain
Amal Saadallah	TU Dortmund, Germany
Matthia Sabatelli	University of Groningen, the Netherlands
Chafik Samir	CNRS-UCA, France
Ramses Sanchez	University of Bonn, Germany
Ioannis Sarridis	Information Technologies Institute/Centre for Research & Technology - Hellas, Greece
Milos Savic	University of Novi Sad, Serbia
Nripsuta Saxena	University of Southern California, USA
Alexander Schiendorfer	Technische Hochschule Ingolstadt, Germany
Christian Schlauch	Humboldt-Universität zu Berlin, Germany
Rainer Schlosser	Hasso Plattner Institute, Germany
Johannes Schneider	University of Liechtenstein, Liechtenstein
Rianne Schouten	Technische Universiteit Eindhoven, the Netherlands
Andreas Schwung	Fachhochschule Südwestfalen, Germany
Patrick Schäfer	Humboldt-Universität zu Berlin, Germany
Kristen Scott	KU Leuven, Belgium
Marian Scuturici	LIRIS, France
Raquel Sebastião	ESTGV-IPV & IEETA-UA
Nina Seemann	University of the Bundeswehr, Germany
Artūras Serackis	Vilnius Tech, Lithuania
Giuseppe Serra	Goethe University Frankfurt, Germany
Mattia Setzu	University of Pisa, Italy
Manali Sharma	Samsung, USA
Shubhranshu Shekhar	Brandeis University, USA
Qiang Sheng	Institute of Computing Technology, Chinese Academy of Sciences, China
John Sheppard	Montana State University, USA
Bin Shi	Xi'an Jiaotong University, China
Jimeng Shi	Florida International University, USA
Paula Silva	INESC TEC - LIAAD, Portugal

Telmo Silva Filho	University of Bristol, UK
Esther-Lydia Silva-Ramírez	Universidad de Cádiz, Spain
Raivydas Šimėnas	Vilnius University, Lithuania
Kuldeep Singh	Cerence GmbH, Germany
Andrzej Skowron	University of Warsaw, Poland
Carlos Soares	University of Porto, Portugal
Dennis Soemers	Maastricht University, the Netherlands
Andy Song	RMIT University, Australia
Liyan Song	Harbin Institute of Technology, China
Zixing Song	Chinese University of Hong Kong, China
Sucheta Soundarajan	Syracuse University, USA
Fabian Spaeh	Boston University, USA
Myra Spiliopoulou	Otto-von-Guericke-University Magdeburg, Germany
Dimitri Staufer	TU Berlin, Germany
Kostas Stefanidis	Tampere University, Finland
Pavel Stefanovič	Vilnius Tech, Lithuania
Julian Stier	University of Passau, Germany
Giovanni Stilo	Università of L'Aquila, Italy
Michiel Stock	Ghent University, Belgium
Luca Stradiotti	KU Leuven, Belgium
Lukas Struppek	Technical University of Darmstadt, Germany
Maximilian Stubbemann	University of Hildesheim, Germany
Nikolaos Stylianou	Information Technologies Institute, Greece
Jinyan Su	University of Electronic Science and Technology of China, China
Peijie Sun	Tsinghua University, China
Weiwei Sun	Shandong University, China
Swati Swati	Universität der Bundeswehr München, Germany
Panagiotis Symeonidis	University of the Aegean, Greece
Maryam Tabar	University of Texas at San Antonio, USA
Shazia Tabassum	INESC TEC, Portugal
Andrea Tagarelli	DIMES - UNICAL, Italy
Martin Takac	Mohamed bin Zayed University of Artificial Intelligence, UAE
Acar Tamersoy	NortonLifeLock Research Group, USA
Chang Wei Tan	Monash University, Australia
Xing Tang	Tencent, China
Enzo Tartaglione	Télécom Paris - Institut Polytechnique de Paris, France
Romain Tavenard	Univ. Rennes, LETG/IRISA, France
Gustaf Tegnér	KTH Royal Institute of Technology, Lithuania

Paweł Teisseyre	Warsaw University of Technology, Poland
Alexandre Termier	Université Rennes, France
Stefano Teso	University of Trento, Italy
Surendrabikram Thapa	Virginia Tech, USA
Martin Theobald	University of Luxembourg, Luxembourg
Maximilian Thiessen	TU Wien, Austria
Steffen Thoma	FZI Research Center for Information Technology, Germany
Matteo Tiezzi	University of Siena, Italy
Matteo Tiezzi	SAILab, DIISM, University of Siena, Italy
Gabriele Tolomei	Sapienza University of Rome, Italy
Paulina Tomaszewska	Warsaw University of Technology, Poland
Dinh Tran	King Fahd University of Petroleum & Minerals, Saudi Arabia
Isaac Triguero	Nottingham University, UK
Andre Tättar	University of Tartu, Estonia
Evaldas Vaičiukynas	Kaunas University of Technology, Lithuania
Jente Van Belle	KU Leuven, Belgium
Fabio Vandin	University of Padova, Italy
Aparna S. Varde	Montclair State University, USA
Bruno Veloso	INESC TEC & FEP-UP, Portugal
Dmytro Velychko	University of Oldenburg, Germany
Sreekanth Vempati	Myntra, India
Gabriele Venturato	KU Leuven, Belgium
Michela Venturini	KU Leuven, ITEC, Belgium
Mathias Verbeke	KU Leuven, Belgium
Théo Verhelst	Université libre de Bruxelles, Belgium
Rosana Veroneze	LBiC, UK
Gennaro Vessio	University of Bari 'Aldo Moro', Italy
Paul Viallard	Inria Rennes, France
Herna Viktor	University of Ottawa, Canada
Joao Vinagre	Joint Research Centre - European Commission, Spain
Jean-Noël Vittaut	Sorbonne Université, CNRS, LIP6, France
Maximilian von Zastrow	Southern Denmark University, Denmark
Tomasz Walkowiak	Wrocław University of Science and Technology, Poland
Beilun Wang	Southeast University, China
Huandong Wang	Tsinghua University, China
Hui (Wendy) Wang	Stevens Institute of Technology, USA
Jianwu Wang	University of Maryland, Baltimore County, USA
Jiaqi Wang	Penn State University, USA

Suhang Wang	Pennsylvania State University, USA
Yanhao Wang	East China Normal University, China
Yimu Wang	University of Waterloo, Canada
Yue Wang	Microsoft Research
Zhaonan Wang	University of Illinois Urbana-Champaign, USA
Zichong Wang	Florida International University, USA
Zifu Wang	KU Leuven, Belgium
Zijie J. Wang	Georgia Tech, USA
Roger Wattenhofer	ETH Zurich, Germany
Tonio Weidler	Maastricht University, the Netherlands
Jörg Wicker	University of Auckland, New Zealand
Alicja Wieczorkowska	Polish-Japanese Academy of Information Technology, Poland
Michael Wilbur	Vanderbilt University, USA
David Winkel	LMU Munich, Germany
Moritz Wohlstein	Leuphana Universität Lüneburg, Germany
Szymon Wojciechowski	Wrocław University of Science and Technology, Poland
Bin Wu	Zhengzhou University, China
Chenwang Wu	University of Science and Technology of China, China
Di Wu	Chongqing Institute of Green and Intelligent Technology, Chinese Academy of Sciences, China
Wei Wu	Ben Gurion University of the Negev, Israel
Yongkai Wu	Clemson University, USA
Zhiwen Xiao	Southwest Jiaotong University, China
Cheng Xie	Yunnan University, China
Yaqi Xie	Carnegie Mellon University, USA
Huanlai Xing	Southwest Jiaotong University, China
Xing Xing	Tongji University, China
Ning Xu	Southeast University, China
Weifeng Xu	Weifeng Xu, USA
Ziqi Xu	CSIRO, Australia
Yexiang Xue	Purdue University, USA
Yan Yan	Carleton University, Canada
Yu Yan	School of Information and Cyber Security, People's Public Security University of China, China
Lincen Yang	Leiden University, the Netherlands
Shaofu Yang	Southeast University, China
Muchao Ye	Pennsylvania State University, USA
Kalidas Yeturu	Indian Institute of Technology Tirupati, India

Jaemin Yoo	KAIST, South Korea
Kristina Yordanova	University of Greifswald, Germany
Hang Yu	Shanghai University, China
Jidong Yuan	Beijing Jiaotong University, China
Xiaoyong Yuan	Clemson University, USA
Klim Zaporojets	Aarhus University, Denmark
Claudius Zelenka	Kiel University, Germany
Akka Zemmari	Univ. Bordeaux, France
Guoxi Zhang	Beijing Institute of General Artificial Intelligence, China
Hao Zhang	Fudan University, China
Teng Zhang	Huazhong University of Science and Technology, China
Tianlin Zhang	University of Manchester, UK
Xiang Zhang	National University of Defense Technology, China
Xiao Zhang	Shandong University, China
Xiaoming Zhang	Beihang University, China
Yaqian Zhang	University of Waikato, New Zealand
Yin Zhang	University of Electronic Science and Technology of China
Zhiwen Zhang	University of Tokyo, Japan
Lingxiao Zhao	Carnegie Mellon University, USA
Tongya Zheng	Hangzhou City University, China
Wenhao Zheng	Shopee, Singapore
Yu Zheng	Tsinghua University, China
Yujia Zheng	CMU, USA
Zhengyang Zhou	University of Science and Technology of China, China
Jing Zhu	University of Michigan, Ann Arbor, USA
Ye Zhu	Deakin University, Australia
Yichen Zhu	Midea Group, China
Zirui Zhuang	Beijing University of Posts and Telecommunications, China
Tommaso Zoppi	University of Florence, Italy
Pedro Zuidberg Dos Martires	Örebro University, Sweden
Meiyun Zuo	Renmin University of China, China

Program Committee Members, Applied Data Science Track

Ziawasch Abedjan	TU Berlin, Germany
Shahrooz Abghari	Blekinge Institute of Technology, Sweden
Christian M. Adriano	Hasso-Plattner Institute, Germany
Haluk Akay	KTH Royal Institute of Technology, Lithuania
Fahed Alkhabbas	Malmo University, Sweden
Mohammed Ghaith Altarabichi	Högskolan i Halmstad, Sweden
Evelin Amorim	INESC TEC, Portugal
Giuseppina Andresini	University of Bari 'Aldo Moro', Italy
Sunil Aryal	Deakin University, New Zealand
Awais Ashfaq	Region Halland, Sweden
Asma Atamna	Ruhr-University Bochum, Germany
Berkay Aydin	Georgia State University, USA
Mehdi Bahrami	Fujitsu Research of America, USA
Hareesh Bahuleyan	Zalando, Sweden
Michael Bain	University of New South Wales, Australia
Hubert Baniecki	University of Warsaw, Poland
Enda Barrett	University of Galway, Ireland
Michele Bernardini	Università Politecnica delle Marche, Ancona, Italy
Lilian Berton	Universidade Federal de Sao Paulo, Brazil
Antonio Bevilacqua	Meetecho, Italy
Szymon Bobek	Jagiellonian University, Poland
Veselka Boeva	Blekinge Institute of Technology, Sweden
Martin Boldt	Blekinge Institute of Technology, Sweden
Anton Borg	Blekinge Institute of Technology, Sweden
Cecile Bothorel	IMT Atlantique, France
Mohamed Reda Bouadjenek	Deakin University, New Zealand
Axel Brando	Barcelona Supercomputing Center (BSC) and Universitat de Barcelona (UB), Spain
Stefan Byttner	Halmstad University, Sweden
Ece Calikus	KTH Royal Institute of Technology, Lithuania
Shilei Cao	Tencent, China
Yixuan Cao	Institute of Computing Technology, CAS, China
Hau Chan	University of Nebraska-Lincoln, USA
Chung-Chi Chen	National Taiwan University, Taiwan
Lei Chen	Hong Kong University of Science and Technology, China
Wei-Peng Chen	Fujitsu Research of America, USA
Zhiyu Chen	Amazon, USA
Dawei Cheng	Tongji University, China

Wei Cheng	NEC Laboratories America
Farhana Choudhury	University of Melbourne, Australia
Lingyang Chu	McMaster University, Canada
Zhendong Chu	University of Virginia, USA
Paolo Cintia	Kode srl, Italy
Pablo José Del Moral Pastor	Ekkono.ai, Sweden
Yushun Dong	University of Virginia, USA
Antoine Doucet	La Rochelle Université, France
Farzaneh Etminani	Halmstad University and Region Halland, Sweden
Michael Faerber	KIT, Germany
Yuantao Fan	Halmstad University, Sweden
Yixiang Fang	Chinese University of Hong Kong, China
Damien Fay	INFOR Logicblox, USA
Dayne Freitag	SRI International, USA
Erik Frisk	Linköping University, Sweden
Yanjie Fu	Arizona State University, USA
Ariel Fuxman	Google, USA
Xiaofeng Gao	Shanghai Jiaotong University, China
Yunjun Gao	Zhejiang University, China
Lluis Garcia-Pueyo	Meta, USA
Mariana-Iuliana Georgescu	Helmholtz Munich, Germany
Aakash Goel	Amazon, USA
Markus Götz	Karlsruhe Institute of Technology (KIT), Germany
Håkan Grahn	Blekinge Institute of Technology, Sweden
Francesco Guerra	University of Modena e Reggio Emilia, Italy
Nuno RPS Guimarães	INESC TEC & University of Porto, Portugal
Huifeng Guo	Huawei Noah's Ark Lab, Canada
Vinayak Gupta	University of Washington Seattle, USA
Jinyoung Han	Sungkyunkwan University, South Korea
Shuchu Han	Stellarcyber, USA
Julia Handl	University of Manchester, UK
Atiye Sadat Hashemi	Halmstad University, Sweden
Aron Henriksson	Stockholm University, Sweden
Andreas Holzinger	University of Natural Resources and Life Sciences Vienna, Austria
Sebastian Hönel	Linnaeus University, Sweden
Ping-Chun Hsieh	National Yang Ming Chiao Tung University, Taiwan
Zhengyu Hu	HKUST, China
Chao Huang	University of Notre Dame, USA

Hong Huang	Huazhong University of Science and Technology, China
Yizheng Huang	York University, UK
Yu Huang	University of Florida, USA
Angelo Impedovo	Niuma s.r.l., Italy
Radu Tudor Ionescu	University of Bucharest, Romania
Wei Jin	Emory University, USA
Xiaobo Jin	Xi’an Jiaotong-Liverpool University, China
Xiaolong Jin	Institute of Computing Technology, CAS, China
Pinar Karagoz	Middle East Technical University (METU), Turkey
Saeed Karami Zarandi	Halmstad University, Sweden
Thomas Kober	Zalando, Germany
Elizaveta Kopacheva	LNU, Sweden
Christos Koutras	TU Delft, the Netherlands
Adit Krishnan	University of Illinois at Urbana-Champaign, USA
Rafal Kucharski	Jagiellonian University, Poland
Niraj Kumar	Fujitsu, India
Krzysztof Kutt	Jagiellonian University, Poland
Susana Ladra	University of A Coruña, Spain
Matthieu Latapy	CNRS, France
Niklas Lavesson	Blekinge Institute of Technology, Sweden
Roy Ka-Wei Lee	Singapore University of Technology and Design, Singapore
Alessandro Leite	Inria, France
Daniel Lemire	University of Quebec (TELUQ), Canada
Chang Li	Apple, USA
Daifeng Li	Sun Yat-Sen University, China
Haifang Li	Baidu Inc., China
Junxuan Li	Microsoft, USA
Lei Li	Hong Kong University of Science and Technology, China
Shijun Li	University of Science and Technology of China
Shuai Li	University of Cambridge, UK and University of Tokyo, Japan and Tsinghua University, China
Wei Li	Harbin Engineering University, China
Xiang Lian	Kent State University, USA
Guojun Liang	Halmstad University, Sweden
Zhaohui Liang	National Library of Medicine, NIH, USA
Kwan Hui Lim	Singapore University of Technology and Design, Singapore
Adi Lin	Didi, China

Bang Liu	University of Montreal, Canada
Dugang Liu	Guangdong Laboratory of Artificial Intelligence and Digital Economy (SZ), Shenzhen University, China
Jingjing Liu	MD Anderson Cancer Center, USA
Li Liu	Chongqing University, China
Qing Liu	Zhejiang University, China
Xueyan Liu	Jilin University, China
Yongchao Liu	Ant Group, China
Andreas Lommatzsch	TU Berlin, Germany
Ping Luo	Chinese Academy of Sciences, China
Guixiang Ma	Intel Labs, USA
Zongyang Ma	York University, UK
Saulo Martiello Mastelini	Volt Robotics, Brazil
Elio Masciari	University of Naples, Italy
Nédra Mellouli	LIASD, France
Zoltan Miklos	University of Rennes, France
Mihaela Mitici	Utrecht University, the Netherlands
Martin Mladenov	Google, Brazil
Ahmed K. Mohamed	Meta, USA
Seung-Hoon Na	Jeonbuk National University, South Korea
Sepideh Nahali	York University, UK
Mirco Nanni	CNR-ISTI Pisa, Italy
Richi Nayak	Queensland University of Technology, Brisbane, Australia
Wee Siong Ng	Institute for Infocomm Research, Singapore
Le Nguyen	University of Oulu, Finland
Thanh Thi Nguyen	Monash University, Australia
Slawomir Nowaczyk	Halmstad University, Sweden
Tomas Olsson	RISE SICS, Sweden
Panagiotis Papadakos	FORTH-ICS, Greece
Manos Papagelis	York University, UK
Panagiotis Papapetrou	Stockholm University, Sweden
Luca Pappalardo	ISTI, Italy
Sepideh Pashami	Halmstad University, Sweden
Vincenzo Pasquadibisceglie	University of Bari ‘Aldo Moro’, Italy
Leonardo Pellegrina	University of Padova, Italy
Pop Petrica	Technical University of Cluj-Napoca, Romania
Pablo Picazo-Sanchez	Halmstad University, Sweden
Srijith PK	IIT, Hyderabad, India
Buyue Qian	Xi’an Jiaotong University, China
Enayat Rajabi	Halmstad University, Sweden

Yanghui Rao	Sun Yat-sen University, China
Salvatore Rinzivillo	KDDLab - ISTI - CNR, Italy
Riccardo Rosati	Università Politecnica delle Marche, Ancona, Italy
Stefan Rueping	Fraunhofer IAIS, Germany
Snehanshu Saha	BITS Pilani Goa Campus, India
Lou Salaün	Nokia Bell Labs, France
Isak Samsten	Stockholm University, Sweden
Eric Sanjuan	Avignon University, France
Johannes Schneider	University of Liechtenstein, Liechtenstein
Wei Shao	Data61, CSIRO, Australia
Nasrullah Sheikh	IBM Research, USA
Jun Shen	University of Wollongong, Australia
Jingwen Shi	Michigan State University, USA
Yue Shi	Meta, USA
Carlos N. Silla	Pontifical Catholic University of Parana (PUCPR), Brazil
Gianmaria Silvello	University of Padova, Italy
Yang Song	Apple, USA
Shafiullah Soomro	Linnaeus University, Sweden
Efstathios Stamatatos	University of the Aegean, Greece
Ting Su	Imperial College London, UK
Gan Sun	South China University of Technology, China
Munira Syed	Procter & Gamble, USA
Zahra Taghiyarrenani	Halmstad University, Sweden
Liang Tang	Google, USA
Xing Tang	Tencent, China
Junichi Tatemura	Google, USA
Joe Tekli	Lebanese American University, Lebanon
Mingfei Teng	Amazon, USA
Sofia Tolmach	Amazon, USA
Gabriele Tolomei	Sapienza University of Rome, Italy
Ismail Hakki Toroslu	METU, Turkey
Md Zia Ullah	Edinburgh Napier University, UK
Maurice Van Keulen	University of Twente, the Netherlands
Ranga Raju Vatsavai	North Carolina State University, USA
Bruno Veloso	INESC TEC & FEP-UP, Portugal
Chang-Dong Wang	Sun Yat-sen University, China
Chengyu Wang	Alibaba Group, China
Kai Wang	Shanghai Jiao Tong University, China
Pengyuan Wang	University of Georgia, USA
Sen Wang	University of Queensland, USA

Senzhang Wang	Central South University, China
Sheng Wang	Wuhan University, China
Wei Wang	Tsinghua University, China
Wentao Wang	Michigan State University, USA
Xiaoli Wang	Xiamen University, China
Yang Wang	University of Science and Technology of China, China
Yu Wang	Vanderbilt University, USA
Zhibo Wang	Zhejiang University, China
Paweł Wawrzyński	IDEAS NCBR, Poland
Hua Wei	Arizona State University, USA
Shi-ting Wen	Ningbo Tech University, China
Zeyi Wen	Hong Kong University of Science and Technology, China
Avani Wildani	Emory University, USA
Fangzhao Wu	MSRA, China
Jun Wu	University of Illinois at Urbana–Champaign, USA
Wentao Wu	Microsoft Research, USA
Xianchao Wu	NVIDIA, Japan
Haoyi Xiong	Baidu, Inc., China
Guandong Xu	University of Technology Sydney, Australia
Yu Yang	City University of Hong Kong, China
Lina Yao	University of New South Wales, Australia
Fanghua Ye	University College London, UK
Dongxiao Yu	Shandong University, China
Haomin Yu	Aalborg University, Denmark
Ran Yu	DSIS Research Group, University of Bonn, Germany
Erik Zeitler	Stream Analyze, Sweden
Chunhui Zhang	Dartmouth College, USA
Denghui Zhang	Rutgers University, USA
Li Zhang	University of Sheffield, UK
Mengxuan Zhang	Australian National University, Australia
Kaiping Zheng	National University of Singapore
Yucheng Zhou	University of Macau, China
Yuanyuan Zhu	Wuhan University, China
Ziwei Zhu	George Mason University, USA
Vasileios Zografos	sennder, Germany

Program Committee Members, Demo Track

Bijaya Adhikari	University of Iowa, USA
Andrius Budrionis	Norwegian Centre for E-health Research, Norway
Luca Cagliero	Politecnico di Torino, Italy
Tania Cerquitelli	Politecnico di Torino, Italy
Gintautas Daunys	Vilnius University, Lithuania
Katharina Dost	University of Auckland, New Zealand
Sourav Dutta	Huawei Research Centre, Ireland
Françoise Fessant	Orange, France
Christelle Godin	CEA, France
Anil Goyal	Amazon, India
Maciej Grzenda	Warsaw University of Technology, Poland
Marius Gudauskis	Institute of Mechatronics, KTU, Lithuania
Thomas Guyet	Inria, Centre de Lyon, France
Andreas Henelius	Independent Researcher, Finland
Rokas Jurevicius	Scandit AG, Lithuania/Switzerland
Pawan Kumar	IIIT, Hyderabad, India
Olga Kurasova	Vilnius University, Institute of Data Science and Digital Technologies, Lithuania
Moreno La Quatra	Kore University of Enna, Italy
Jan Lemeire	Vrije Universiteit Brussel (VUB), Belgium
Martin Luckner	Warsaw University of Technology, Poland
Hoang Phuc Hau Luu	University of Helsinki, Finland
Jarmo Mäkelä	CSC - IT Center for Science Ltd, Finland
Michael Mathioudakis	University of Helsinki, Finland
Darius Miniotas	Vilnius Gediminas Technical University, Lithuania
Michalis Mountantonakis	FORTH-ICS, and CS Department - University of Crete, Greece
Raj Nath Patel	Huawei Ireland Research Center, Ireland
Darius Plikynas	Vilnius Gediminas Technical University, Lithuania
Alexandre Reiffers	IMT Atlantique, France
Marina Reyboz	Univ. Grenoble Alpes, CEA, LIST, France
Yuya Sasaki	Osaka University, Japan
Ines Sousa	Fraunhofer AICOS, Portugal
Jerzy Stefanowski	Poznan University of Technology, Poland
Guoxin Su	University of Wollongong, Australia
Lu-An Tang	NEC Labs America, USA
Michael C. Thrun	Philipps-Universität Marburg, Germany

Yannis Tzitzikas	FORTH-ICS and Computer Science Department, University of Crete, Greece
Aleksandras Voicikas	Vilnius University, Lithuania
Jörg Wicker	University of Auckland, New Zealand
Hao Xue	University of New South Wales, Australia

Sponsors

CENTAI

eDF

ASML

Vinted

Google

AstraZeneca

BNP PARIBAS
The bank for a changing world

ARTIFICIAL INTELLIGENCE ASSOCIATION OF LITHUANIA

forest 4.0

Go Vilnius

Open for Innovation
KNIME

VYTAUTAS MAGNUS UNIVERSITY
Faculty of Informatics

NOVIAN

Faculty of Mathematics and Informatics

ALTEN

Invited Talks Abstracts

The Dynamics of Memorization and Unlearning

Gintarė Karolina Džiugaitė

Google DeepMind

Abstract. Deep learning models exhibit a complex interplay between memorization and generalization. This talk will begin by exploring the ubiquitous nature of memorization, drawing on prior work on "data diets", example difficulty, pruning, and other empirical evidence. But is memorization essential for generalization? Our recent theoretical work suggests that eliminating it entirely may not be feasible. Instead, I will discuss strategies to mitigate unwanted memorization by focusing on better data curation and efficient unlearning mechanisms. Additionally, I will examine the potential of pruning techniques to selectively remove memorized examples and explore their impact on factual recall versus in-context learning.

Biography: Gintarė is a senior research scientist at Google DeepMind, based in Toronto, an adjunct professor in the McGill University School of Computer Science, and an associate industry member of Mila, the Quebec AI Institute. Prior to joining Google, Gintarė led the Trustworthy AI program at Element AI/ServiceNow, and obtained her Ph.D. in machine learning from the University of Cambridge, under the supervision of Zoubin Ghahramani. Gintarė was recognized as a Rising Star in Machine Learning by the University of Maryland program in 2019. Her research combines theoretical and empirical approaches to understanding deep learning, with a focus on generalization, memorization, unlearning, and network compression.

The Emerging Science of Benchmarks

Moritz Hardt

Max Planck Institute for Intelligent Systems

Abstract. Benchmarks have played a central role in the progress of machine learning research since the 1980s. Although there's much researchers have done with them, we still know little about how and why benchmarks work. In this talk, I will trace the rudiments of an emerging science of benchmarks through selected empirical and theoretical observations. Looking back at the ImageNet era, I'll discuss what we learned about the validity of model rankings and the role of label errors. Looking ahead, I'll talk about new challenges to benchmarking and evaluation in the era of large language models. The results we'll encounter challenge conventional wisdom and underscore the benefits of developing a science of benchmarks.

Biography: Hardt is a director at the Max Planck Institute for Intelligent Systems, Tübingen. Previously, he was Associate Professor for Electrical Engineering and Computer Sciences at the University of California, Berkeley. His research contributes to the scientific foundations of machine learning and algorithmic decision making with a focus on social questions. He co-authored Fairness and Machine Learning: Limitations and Opportunities (MIT Press) and Patterns, Predictions, and Actions: Foundations of Machine Learning (Princeton University Press).

Enhancing User Experience with AI-Powered Search and Recommendations at Spotify

Mounia Lalmas-Roelleke

Spotify

Abstract. This talk will explore the pivotal role of search and recommendation systems in enhancing the Spotify user experience. These systems serve as the gateway to Spotify's vast audio catalog, helping users navigate millions of music tracks, podcasts, and audiobooks. Effective search functionality allows users to quickly find specific content, whether it is a favorite song, a trending podcast, or an informative audiobook, while also satisfying broader search needs. Meanwhile, recommendation systems suggest new and relevant content that users might not have thought to search for, while ensuring their current needs for familiar content are met. This encourages exploration and discovery of new artists, genres, and shows, enriching the overall listening experience and keeping users engaged with the platform. Achieving this dual objective of precision and discovery requires sophisticated technology. It involves a deep understanding of representation learning, where both content and user preferences are accurately modeled. Advanced AI techniques, including machine learning and generative AI, play a crucial role in this process. These technologies enable the creation of highly personalized recommendations by understanding complex user behaviors and preferences. Generative AI, for instance, allows us to create personalized playlists, thereby enhancing the user experience with innovative features. This presentation is based on the collective research and publications of numerous contributors at Spotify.

Biography: Mounia is a Senior Director of Research at Spotify and the Head of Tech Research in Personalization, where she leads an interdisciplinary team of research scientists. She also holds an honorary professorship at University College London and serves as a Distinguished Research Fellow at the University of Amsterdam. Previously, Mounia was a Director of Research at Yahoo, overseeing a team focused on advertising quality and collaborating on user engagement projects related to news, search, and user-generated content. Before her tenure at Yahoo, Mounia held a Microsoft Research/RAEng Research Chair at the School of Computing Science, University of Glasgow, and before that was a Professor of Information Retrieval at the Department of Computer Science at Queen Mary, University of London. She is a prominent figure in the research community, regularly serving as a senior program committee member at major conferences such as WSDM, KDD, WWW, and SIGIR. She was also a program co-chair for SIGIR 2015, WWW 2018, WSDM 2020, and CIKM 2023. Mounia is widely recognized for her contributions as a speaker and author, with over 250 published papers and appearances on platforms like ACM ByteCast and the AI Business Podcasts series. She was nominated for the VentureBeat Women in AI Awards for Research in both 2022 and 2023.

How to Utilize (and Generate) Player Tracking Data in Sport

Patrick Lucey

Stats Perform

Abstract. Even though player tracking data in sports has been around for 25 years, it still poses as one of the most interesting and challenging datasets in machine learning due to its fine-grained, multi-agent, team-based, and adversarial nature. Despite these challenges, it is also extremely valuable as it is (relatively) low-dimensional, interpretable, and interactive, allowing us to measure performance and answer questions we couldn't objectively address before. In this talk, I will first give a brief history of tracking data in sports, then highlight the challenges associated with utilizing it. I will then show that by obtaining a permutation invariant representation, we can not only measure aspects of sports that couldn't be done before, but also interact with and simulate plays akin to a video game via our "visual search" and "ghosting" technology. Finally, I will show how we can use both tracking and event data to create a multimodal foundation model, which enables us to generate player tracking data at scale and achieve our goal of "digitizing every game of professional sport." Throughout the talk, I will utilize examples from top-tier basketball, soccer, and tennis.

Biography: Patrick Lucey is currently the Chief Scientist at sports data giant Stats Perform, leading the AI team with the goal of maximizing the value of the company's extensive sports data. He has studied and worked in the fields of machine learning and computer vision for the past 20 years, holding research positions at Disney Research and the Robotics Institute at Carnegie Mellon University, as well as spending time at IBM's T.J. Watson Research Center while pursuing his Ph.D. Patrick originally hails from Australia, where he received his BEng(EE) from the University of Southern Queensland and his doctorate from Queensland University of Technology, which focused on multimodal speech modeling. He has authored more than 100 peer-reviewed papers and has been a co-author on papers in the MIT Sloan Sports Analytics Conference Best Research Paper Track for 11 of the last 13 years, winning best paper in 2016 and runner-up in 2017 and 2018. Additionally, he has won best paper awards at INTERSPEECH and WACV international conferences. His main research interests are in artificial intelligence and interactive machine learning in sporting domains, as well as AI education. He has recently piloted a course on "AI in Sport," which aims to give students intuition behind AI methods using the interactive and visual nature of sports data.

Website: www.patricklucey.com

Resource-Aware Machine Learning—A User-Oriented Approach

Katharina Morik

TU Dortmund University

Abstract. Machine Learning (ML) has become integrated into several processes, ranging from medicine, manufacturing, logistics, smart cities, sales, recommendations and advertisements to entertainment and many more business and private processes. The applications together consume a considerable amount of energy and emit CO2. ML research investigates how to make models smaller and faster through pruning and quantization. Also the use of more energy-efficient hardware is an encouraging field. Research on ML under resource constraints is an active field proposing novel algorithms and scenarios. The aim is that for each application a variety of implementations is offered from which customers and the different types of users may choose the most thrifty one. This, in turn, would push tech providers to focus on the production of economical systems. However, if the customers, users, stakeholders do not know which of the models offers the best tradeoff between performance and energy-efficiency, they cannot select the most frugal one. Hence, testing implementations of learning and inference needs to be developed. They should be easy to use, produce visualizations that are mass-tailored for specific user groups. Automatized testing is difficult due to the diversity of models, computing architectures, training and evaluation data, and the fast rate of changes. The talk will illustrate work on resource-aware ML and advocate to pay more attention to the role of users in the development of scenarios, models, and tests.

Biography: Katharina Morik received her doctorate from the University of Hamburg in 1981 and her habilitation from the TU Berlin in 1988. In 1991, she established the chair of Artificial Intelligence at the TU Dortmund. She retired in 2023. She is a pioneer of bringing machine learning and computing architectures together so that machine learning models may be executed or even trained on resource restricted devices. In 2011, she acquired the Collaborative Research Center CRC 876 "Providing Information by Resource-Constrained Data Analysis" consisting of 12 projects and a graduate school. After the longest possible funding period of 12 years, the CRC ended with the publication of 3 books on Resource-Constrained Machine Learning (De Gruyter). She has participated in numerous European research projects and has been the coordinator of one. She was a founding member and Program Chair of the conference series IEEE International Conference on Data Mining (ICDM) and is a member of the steering committee of ECML PKDD. She is a co-founder of the Lamarr Institute for Machine Learning and Artificial Intelligence. Prof. Morik is a member of the Academy of Technical Sciences and of the North Rhine-Westphalian Academy of Sciences and Arts. She was made a Fellow of the German Society of Computer Science GI e.V. in 2019.

Contents – Part IX

Applied Data Science Track

VulEXplaineR: XAI for Vulnerability Detection on Assembly Code 3
Samaneh Mahdavifar, Mohd Saqib, Benjamin C. M. Fung, Philippe Charland, and Andrew Walenstein

Guiding Catalogue Enrichment with User Queries 21
Yupei Du, Jacek Golebiowski, Philipp Schmidt, and Ziawasch Abedjan

PeersimGym: An Environment for Solving the Task Offloading Problem with Reinforcement Learning 38
Frederico Metelo, Cláudia Soares, Stevo Racković, and Pedro Ákos Costa

Robust Interaction-Based Relevance Modeling for Online e-Commerce Search 55
Ben Chen, Huangyu Dai, Xiang Ma, Wen Jiang, and Wei Ning

Learning Optimal Linear Precoding for Cell-Free Massive MIMO with GNN 72
Benjamin Parlier, Lou Salaün, and Hong Yang

Multiple Hypergraph Learning for Ephemeral Group Recommendation 89
Rui Zhao, Beihong Jin, Yimin Lv, Yiyuan Zheng, and Weijiang Lai

Spoofing Transaction Detection with Group Perceptual Enhanced Graph Neural Network 106
Le Kang, Tai-Jiang Mu, and XiaoDong Ning

Self-SLAM: A Self-supervised Learning Based Annotation Method to Reduce Labeling Overhead 123
Alfiya M. Shaikh, Hrithik Nambiar, Kshitish Ghate, Swarnali Banik, Sougata Sen, Surjya Ghosh, Vaskar Raychoudhury, Niloy Ganguly, and Snehanshu Saha

Multi-intent Driven Contrastive Sequential Recommendation 141
Yiyuan Zheng, Beibei Li, Beihong Jin, Rui Zhao, Weijiang Lai, and Tao Xiang

KAT5: Knowledge-Aware Transfer Learning with a Text-to-Text Transfer Transformer 157
Mohammad Golam Sohrab and Makoto Miwa

Asymmetric Graph-Based Deep Reinforcement Learning for Portfolio Optimization 174
Haoyu Sun, Xin Liu, Yuxuan Bian, Peng Zhu, Dawei Cheng, and Yuqi Liang

Code Summarization with Project-Specific Features 190
Yu Wang, Xin Liu, Xuesong Lu, and Aoying Zhou

Unraveling Anomalies in Time: Unsupervised Discovery and Isolation of Anomalous Behavior in Bio-Regenerative Life Support System Telemetry 207
Ferdinand Rewicki, Jakob Gawlikowski, Julia Niebling, and Joachim Denzler

Long-Term Fairness in Ride-Hailing Platform 223
Yufan Kang, Jeffrey Chan, Wei Shao, Flora D. Salim, and Christopher Leckie

A Merge Sort Based Ranking System for the Evaluation of Large Language Models 240
Chenchen Li, Linfeng Shi, Chunyi Zhou, Zhaoxin Huan, Chengfu Tang, Xiaolu Zhang, Xudong Wang, Jun Zhou, and Song Liu

Enhancing HVAC Control Efficiency: A Hybrid Approach Using Imitation and Reinforcement Learning 256
Kevlyn Kadamala, Des Chambers, and Enda Barrett

Synthesis of Standard 12-Lead ECG from Single-Lead ECG Using Shifted Diffusion Models 271
Jingwei Liu, Hongyan Li, and Shenda Hong

SAGS-DynamicBio: Integrating Semantic-Aware and Graph Structure-Aware Embedding for Dynamic Biological Data with Knowledge Graphs 287
Yao Liu, Yongfei Zhang, and Xin Wang

Graph Machine Learning for Fast Product Development from Formulation Trials 303
Manuel Dileo, Raffaele Olmeda, Margherita Pindaro, and Matteo Zignani

HPExplorer: XAI Method to Explore the Relationship Between Hyperparameters and Model Performance 319
Yulia Grushetskaya, Mike Sips, Reyko Schachtschneider, Mohammadmehdi Saberioon, and Akram Mahan

Boosting Patient Representation Learning via Graph Contrastive Learning 335
Zhenhao Zhang, Yuxi Liu, Jiang Bian, Antonio Jimeno Yepes, Jun Shen, Fuyi Li, Guodong Long, and Flora D. Salim

Time Series Clustering for Enhanced Dynamic Allocation in A/B Testing 351
Emmanuelle Claeys, Myriam Maumy-Bertrand, and Pierre Gançarski

Reinforcement Learning Meets Microeconomics: Learning to Designate Price-Dependent Supply and Demand for Automated Trading 368
Łukasz Lepak and Paweł Wawrzyński

Spatial Transfer Learning for Estimating $PM_{2.5}$ in Data-Poor Regions 385
Shrey Gupta, Yongbee Park, Jianzhao Bi, Suyash Gupta, Andreas Züfle, Avani Wildani, and Yang Liu

Leveraging Foundation Models for Multi-modal Federated Learning with Incomplete Modality 401
Liwei Che, Jiaqi Wang, Xinyue Liu, and Fenglong Ma

Contrastive Learning Enhanced Diffusion Model for Improving Tropical Cyclone Intensity Estimation with Test-Time Adaptation 418
Ziheng Zhou, Haojia Zuo, Ying Zhao, and Wenguang Chen

BESTMVQA: A Benchmark Evaluation System for Medical Visual Question Answering 435
Xiaojie Hong, Zixin Song, Liangzhi Li, Xiaoli Wang, and Feiyan Liu

AeroINR: Meta-learning for Efficient Generation of Aerodynamic Geometries 452
Tom Bamford, David Toal, and Andy Keane

Author Index 469

Applied Data Science Track

VulEXplaineR: XAI for Vulnerability Detection on Assembly Code

Samaneh Mahdavifar[1](✉), Mohd Saqib[1], Benjamin C. M. Fung[1],
Philippe Charland[2], and Andrew Walenstein[3]

[1] Data Mining and Security Lab, McGill University, Montreal H3A 1X1, Canada
`samaneh.mahdavifar@affiliate.mcgill.ca`, `mohd.saqib@mail.mcgill.ca`,
`ben.fung@mcgill.ca`
[2] Defence Research and Development Canada, Ottawa, Canada
`philippe.charland@drdc-rddc.gc.ca`
[3] BlackBerry Limited, Waterloo, Canada
`walenste@ieee.org`

Abstract. Software vulnerabilities have posed significant threats to on-premise as well as cloud servers and applications. So far, numerous studies have focused on identifying and addressing software vulnerabilities at the binary level. Traditional approaches often involve highly complicated static and dynamic analysis techniques. Current intelligent methods are not explainable to reverse engineers, making them incapable of validating the detected vulnerabilities. In this paper, we propose VulEXplaineR, an XAI method for vulnerability detection based on assembly code. It employs BERT for block embedding, augmented with TFIDF of blocks and operand types information, to provide an effective vulnerability detection/explanation framework. VulEXplaineR takes a trained GCNN and its predictions and returns an explanation in the form of a small subgraph of the input graph. It is based on PGExplainer, a perturbation-based global explanation model for GNNs. We augment edge distribution with the edge feature in the form of intra-function jumps between blocks or inter-function calls between functions. The experimental results on the NDSS2018 and Juliet Test datasets demonstrate that VulEXplaineR outperforms the current state-of-the-art baselines in vulnerability detection. Unlike other baseline models, VulEXplaineR provides a high level of explainability as a complementary aid to a reverse engineer, for a more accurate function analysis. We measure fidelity to demonstrate how much two predictions from the extracted subgraph and the original graph match. Furthermore, we conduct a case study to show that VulEXplaineR not only identifies functions and basic blocks that cause the vulnerability, but also highlights interdependencies between those functions and blocks.

Keywords: Vulnerability · explainability · assembly code · BERT · block embedding · graph neural network · subgraph · TFIDF

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-70378-2_1.

© Crown 2024
A. Bifet et al. (Eds.): ECML PKDD 2024, LNAI 14949, pp. 3–20, 2024.
https://doi.org/10.1007/978-3-031-70378-2_1

1 Introduction

A software vulnerability refers to a weakness in system design, implementation, or operational management that, if exploited, can result in various attacks or system crashes. Software vulnerabilities are released through the CVE database. When it comes to vulnerability detection methods, traditional approaches, such as reverse engineering, are manually intensive and time-consuming, making them impractical for mitigating zero-day vulnerabilities. Using deep learning and machine learning, we can automate the process of identifying software vulnerabilities and streamlining the process. However, adding intelligence to vulnerability detection has never been an effortless task. There exists a multitude of challenges on the route of designing an efficient effective detector that is capable of identifying zero-day vulnerabilities. The existing methods do not explain why a binary is vulnerable. To date, several Artificial Intelligence (AI)-based binary vulnerability detection methods have been proposed at both the source code and binary level [9,17]. However, they are not fully explainable to reverse engineers. If they rarely exist, the explanations are low-level and not abstract enough. Thus, reverse engineers cannot benefit from the transparency these methods provide to dissect the observable factors and characteristics of the assembly code. If we can add a human-friendly explanation of the learning representations and the input assembly code, we would enhance the trustworthiness and reliability of the vulnerability detection method we provide. One of the most straightforward explanation approaches is to highlight those parts of the input file that have led to our decision. In this case, the reverse engineer does not need to review all parts of the code instruction by instruction to find vulnerable functions. He can only focus on those parts of the code we have identified by our explainable model. As a result, it increases accuracy and saves time, energy, and resources. From the detection perspective, there have been some attempts to use Graph Neural Networks (GNNs) for training on vulnerability datasets [5,10]. However, existing graph-based detection methods do not consider edge types in the underlying Control Flow Graph (CFG) and treat all jumps and calls equally. This assumption prevents the GNN from modeling the edge distributions correctly and fails to detect unknown vulnerabilities. We believe that knowing whether an edge represents a jump between two blocks in a function or a call between two functions in a CFG has an impact on identifying some specific vulnerabilities.

The primary objective of this research is to create and implement an eXplainable AI (XAI) model to detect binary vulnerabilities. The key aims are twofold: (1) providing evidence and predictions for reverse engineers and (2) identifying vulnerable behaviors rather than solely focusing on features. To achieve this, we introduce VulEXplaineR, a framework designed to identify vulnerabilities by analyzing assembly code. The initial step involves extracting CFG from the assembly code, followed by using BERT for block embedding, enriched with TFIDF incorporating block information and operand types. This integrated approach aims to establish an effective and scalable framework for vulnerability detection and explanation. Furthermore, we offer explainability through a subgraph of Graph Convolutional Neural Networks (GCNNs), ensuring a high level of comprehension for reverse engineers and an accurate representation of the relationships and interdependencies between blocks. Graph visualization provides a high level of explainability and is very much understandable to humans. Also, it provides an intermediary means for rule explanation. The main contributions of this paper are as follows:

- To the best of our knowledge, this is the first work that provides explainability for vulnerability detection in terms of subgraph of the CFG based on a graph explanation model called PGExplainer. This is also the first work to use block jumps and function calls as the edge distribution for the GCNN, unlike previous methods that treat all jumps and calls equally. The edge distribution provides information about how these connections are distributed throughout the graph.
- We use operand type frequency and TFIDF that provide a lightweight feature vector for detecting and explaining vulnerability. TFIDF and operand types alone demonstrate promising results on datasets with less sophisticated vulnerabilities, such as the Juliet Test Suite.
- We augment operand type frequency and TFIDF with BERT for block embedding. Experimental results show that VulExplaineR provides almost the same classification performance as the best performing baseline model, while providing a high level of explainability.
- We evaluate explainability in terms of fidelity, provide a case study to analyze the extracted graph explanation, and validate it using expert knowledge.

The remainder of the paper is organized as follows: we review related work in Sect. 2. The VulExplaineR design is described in Sect. 3. Section 4 provides the experimental results. Finally, Sect. 5 concludes the paper.

2 Related Work

2.1 Vulnerability Detection

Researchers employ machine learning and deep learning techniques to detect vulnerabilities, utilizing source or binary code analysis. In source code-based vulnerability detection, researchers analyze the software code and extract relevant features. For example, Li et al. [12] propose converting the source code into a numeric vector using representation learning, with the aim of reducing false negatives. Harer et al. [7] apply machine learning to detect vulnerabilities in C and C++ programs. In contrast, Cao et al. [3] utilize deep learning with Bidirectional GNN (BGNN) to capture syntax and semantic information for vulnerability detection. Another study by Garcia et al. [6] employs representation learning, specifically principal component analysis, to generate informative representations for the C source code. Similarly, in the case of Java code, Pang et al. [16], and Hovsepyan et al. [8] represent the code using n-grams and synthetic features, before applying a Support Vector Machine (SVM) for vulnerability detection. Russell et al. [17] explore the use of CNNs and Recurrent Neural Networks (RNNs) on real-world C/C++ code datasets. Although these approaches achieve good evaluation metrics, they heavily rely on access to source code, which is only sometimes available. Therefore, researchers have begun to explore alternative approaches based on binary-level vulnerability detection. Binary-level vulnerability detection has gained popularity in research, due to its advantage of using a standard representation for programs across different programming languages. Scholars have hypothesized that assembly code shares similarities with natural language processing by focusing on the assembly code obtained through disassembly. For example, Dahl et al. [4] employed representation learning on assembly code and utilized RNNs to detect vulnerabilities. Similarly, Lee et al. [9] used CNNs to process binary code by converting instructions into vectors.

While these approaches have shown promising results, they can only sometimes capture the semantic relationships among code blocks. Researchers have shifted towards graph-based detection methods to address this limitation, aiming for greater accuracy and precision. For example, Diwan et al. [5] employed representation learning using RoBERTa [13] to encode code blocks into vectors and applied message-passing neural networks to process the entire CFG. Previously, Diwan et al. [5] also explored graph-based techniques at the source code level, as demonstrated in [20]. The research conducted by Diwan et al. [5] achieved high accuracy, but one of the challenges of binary-level detection is the difficulty in interpreting the results. Reverse engineers still need help understanding the underlying reasons for the detected vulnerabilities. Furthermore, deep learning algorithms such as GNNs, RNNs, or CNNs are often perceived as black boxes, further complicating the task for reverse engineers. XAI techniques have been considered as potential solutions to explain why black-box models classify software as vulnerable.

2.2 XAI for Vulnerability Detection

Integrating XAI techniques into vulnerability detection has been a relatively understudied area within the existing literature. Most available XAI algorithms have been predominantly developed and tailored for real-world data domains, such as text and images. However, the unique characteristics of vulnerability data require modifications and adaptations of these XAI algorithms to suit this specific domain. To address this gap, researchers have proposed custom algorithms specifically designed to explain source code-level vulnerabilities. For example, Zou et al. [21] introduced a heuristic searching-based tree generation approach to explain vulnerability detection outcomes. Similarly, Li et al. [11] presented a detailed interpretation framework incorporating subgraphs from the Program Dependency Graph (PDG) containing critical statements related to the identified vulnerabilities. Notably, Li et al. [10] made a significant contribution by pioneering the exploration of binary-level vulnerability detection and explanation. However, their approach relied on attention-based graph classification, which may need to possess the sensitivity for local-level explainability in vulnerability detection. In contrast, our proposed algorithm overcomes this limitation by leveraging power gradient analysis and providing individual explanations tailored to vulnerability detection scenarios.

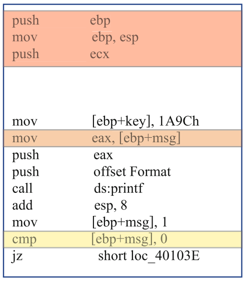
3 Methodology

Explainability is defined as the capacity to convey information clearly to a human audience. It is argued that efficient explanations should be discerning, requiring the choice of 'one or two causalities' from a potentially vast array of causes [15].

3.1 Method of Communication for Vulnerability Detection

In the literature, there are different ways of organizing how an explanation is communicated to an audience, out of which the three following methods could be tailored to the area of vulnerability detection.

- *Input feature explanation* could be based on the whole instruction or tokens (Fig. 1). Although it is highly adaptable to different problems, it is limited to providing abstract explanations and controlling the flow of the program.
- *Rule explanations* attempt to explain the model by a simple set of rules. Rules offer the highest level of abstract explanation and are powerful at approximating non-linear decision boundaries. Although they are a local approximation-based explanation method, they can be generalized to the entire dataset and provide global explanations. However, they are complex when applied directly to the assembly code and have high time complexity.
- *Graph visualization* matches the nature of the graphs of assembly code from two perspectives: (1) the structure and semantic information of the assembly code and (2) the relationships and dependencies between blocks. From the visual interpretation viewpoint, graphs are more human understandable and are an intermediary for rule explanation.



```
push        ebp
mov         ebp, esp
push        ecx


mov         [ebp+key], 1A9Ch
mov         eax, [ebp+msg]
push        eax
push        offset Format
call        ds:printf
add         esp, 8
mov         [ebp+msg], 1
cmp         [ebp+msg], 0
jz           short loc_40103E
```

Fig. 1. Input feature explanation

3.2 Graph Explanation

GNNs have been noticeably employed for representation learning in applications [19] that involve graph-structured data, such as social network data and genomic data. The main idea of learning a representation of graph-structured data is to use a message-passing scheme to enable each node in the graph to capture the feature vector of the neighbor nodes. This way, the GNN can capture both the node features and the topology of the graph. GNNs could be used for node classification, link prediction, graph classification, and graph generation. However, GNNS, like other deep neural networks, suffer from not being explainable to humans, because of their black-box nature. In terms of their application to cybersecurity, the lack of explainability hinders security experts to comprehend how complex decisions are made, leading to decreased levels of trust and reliability in the system. A few attempts have been made in the literature to explain graph predictions based on important subgraphs and sets of features. GNNExplainer [18] is the first general model-agnostic-based approach for explaining GNNs. GNNExplainer explains in terms of a compact subgraph and a small subset of node features that are essential for predicting a specific instance, i.e., a node or a graph in a GNN. However, since GNNExplainer focuses on providing local explainability for a single instance individually, this makes it difficult for the explanations to be generalized to other nodes. Therefore, this approach would not be suitable for comprehending the trained model globally and because of looking at each instance independently, this may generate sub-optimal generalized explanations.

3.3 VulEXplaineR

PGExplainer [14] parametrizes the generation process for explanations that explain multiple instances collectively with a global view of the GNN model. Therefore, it benefits from a higher generalization power. The PGExplainer model enables the inference of explanations for unexplained nodes in an inductive setting, without requiring the retraining of the explanation model. PGExplainer extracts G_s as the explanatory graph, which is the underlying subgraph that makes important contributions to the predictions of a GNN. We have adopted PGExplainer as the base model and exploited the concept of edge distribution to take into account the type of edge in the CFG. For the sake of simplicity, we assume that G_s follows a Gilbert random graph model. In this model, the selection of edges from the original input graph G_0 is treated as conditionally independent of each other. Let V denotes the node set and $E \in V \times V$ be the edge set of Graph G. $e_{ij} \in V \times V$ represents the binary variable that indicates whether the edge is selected, with $e_{ij} = 1$ if the edge (i, j) is selected and 0 otherwise. Further, assume e'_{ij} is the binary variable representing whether the edge is an inter-function call, with $e'_{ij} = 1$ if the edge (i, j) is an inter-function call, and 0 if it is an intra-function call (between blocks). Given G be the random graph variable, based on the above assumptions, the probability of a graph G is factorized as follows:

$$P(G) = \Pi_{(i,j) \in E} P(e_{ij}).P(e'_{ij}) \tag{1}$$

Probabilities $P(e_{ij})$ and $P(e'_{ij})$ could be modeled as Bernoulli distributions, where $e_{ij} \sim Bern(\theta_{ij})$ and $e'_{ij} \sim Bern(\theta'_{ij})$. $P(e_{ij}) = \theta_{ij}$ and $P(e'_{ij}) = \theta'_{ij}$

indicate the probability that edge (i, j) exists in G and is an inter-function call, respectively. The objective function in the PGExplainer algorithm is described as follows [14]:

$$\min_{G_s} H(Y_0|G = G_s) = \min_{G_s} \mathbb{E}_{G_s} \left[H(Y_0|G = G_s)\right] \approx \min_{\Theta} \mathbb{E}_{G_s \sim q(\Theta)} \left[H(Y_0|G = G_s)\right], \quad (2)$$

where $q(\Theta)$ is the distribution of the explanatory graph parameterized by θ and θ'. We apply a relaxation technique to the discrete variable G_s where edge weights, initially binary, are relaxed to continuous variables in the range $(0, 1)$. A reparameterization method is then employed to optimize the objective function efficiently using gradient-based methods. The approach involves using a deterministic function with parameters $\varOmega$, temperature τ, and an independent random variable ϵ to approximate the sampling process $G_s \sim q(\theta)$. Therefore, G_s is approximated by $\hat{G}_s = f_{\varOmega}(G_0, \tau, \epsilon)$ and the weight $\hat{e}_{ij} \in (0, 1)$ of edge (i, j) in $\hat{G}_s$ is calculated by:

$$\hat{e}_{ij} = \sigma((log\epsilon - log(1 - \epsilon) + \omega_{ij})/\tau), \quad \epsilon \sim \text{Uniform}(0, 1), \quad (3)$$

where $\sigma(.)$ is the Sigmoid function, and $\omega_{ij} \in \mathbb{R}$ is the parameter. Thus, with the reparametrization technique, the objective equation in Eq. 2 becomes:

$$\min_{\varOmega} \mathbb{E}_{\epsilon \sim \text{Uniform}(0,1)} H(Y_0|G = G_s). \quad (4)$$

Figure 2 depicts the explanation process. It takes an input graph G_0 to compute $\varOmega$ which serves as the latent variable in edge distributions, essentially representing explanations. To extract the explanatory subgraph, the latent variables are employed to select the highest-rank edges. Then a random graph $\hat{G}_s$ is drawn from the edge distributions and fed to the trained GNN to obtain the prediction $\hat{Y}_s$. Finally, the parameters in the explanation network are optimized by minimizing the cross-entropy between the original prediction Y_0 and the updated prediction $\hat{Y}_s$ [14].

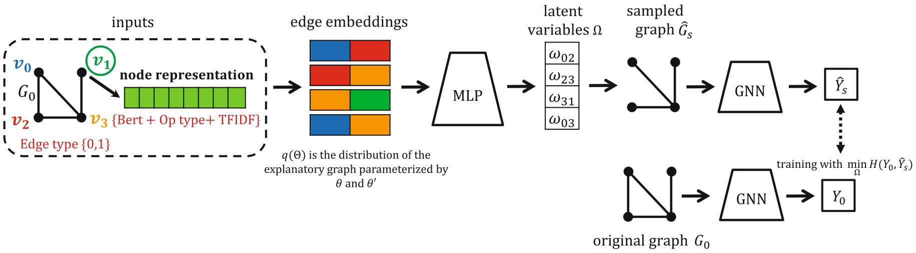
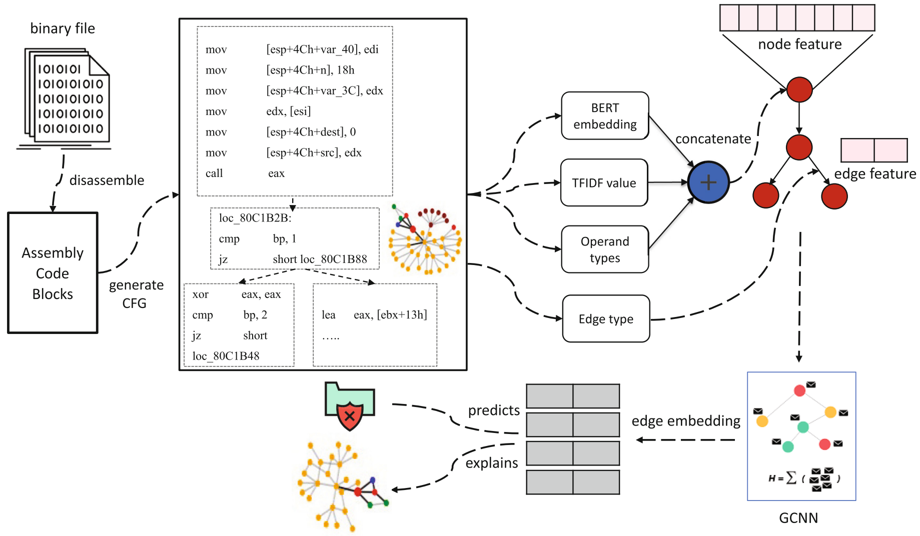

inputs
v_0
v_1
G_0
node representation
v_2
v_3 {Bert + Op type+ TFIDF}
Edge type {0,1}
edge embeddings
$q(\Theta)$ is the distribution of the explanatory graph parameterized by θ and θ'
MLP
latent variables Ω
ω_{02}
ω_{23}
ω_{31}
ω_{03}
sampled graph $\hat{G}_s$
GNN
$\hat{Y}_s$
training with $\min_{\Omega} H(Y_0, \hat{Y}_s)$
GNN
Y_0
original graph G_0

Fig. 2. Extracting explanation graph

Edge distribution is often used to describe how features are propagated between nodes through the edges of the graph. The edge distribution can define how neighboring nodes influence each other during message passing in a GNN. The distribution may determine how much weight or importance is assigned to the information coming from each neighbor. For example, a uniform edge distribution may imply that each neighbor contributes equally to the node's updated representation, while a learned distribution may assign different weights to different neighbors based on the model's learning. In a CFG, the probability distribution of the edges is very important to capture meaningful dependencies between blocks in the graph and therefore, detect zero-day vulnerabilities. To augment the edge distribution in a CFG, we concatenate the edge type as an entry in the edge embedding from which latent variables are computed. Edge types are either intra-function calls between blocks, indicated as zero, or inter-function calls between functions, indicated as one in the edge embedding. Figure 3 shows the overview of the VulEXplaineR framework to predict and explain vulnerability in terms of a subgraph of a GCNN that represents a CFG. First, the input binary files are disassembled into assembly code blocks using a disassembler. Then, we extract the CFG of the program that represents the graphical representation of its different execution paths. Each node in a CFG stands for a basic block and an edge of the graph connects these basic blocks that shows the flow of the program execution. We generate the CFG of the entire assembly file by connecting the functions through the basic blocks of each function that call one another. Using this representation, we can discover risky program execution paths of a file.



```
mov     [esp+4Ch+var_40], edi
mov     [esp+4Ch+n], 18h
mov     [esp+4Ch+var_3C], edx
mov     edx, [esi]
mov     [esp+4Ch+dest], 0
mov     [esp+4Ch+src], edx
call    eax

loc_80C1B2B:
cmp     bp, 1
jz      short loc_80C1B88

xor     eax, eax
cmp     bp, 2
jz      short
loc_80C1B48

lea     eax, [ebx+13h]
.....
```

Fig. 3. A model overview of the VulEXplaineR framework

After that, we need to create the embeddings for the nodes (blocks) and the edges (connectivity between blocks) [5] of the CFG and arrange them in a structure to serve as an input to a GCNN. The node embeddings consist of several components, including the TFIDF value, the BERT embedding, and the frequency of operand types in a basic block. All of these feature vectors are concatenated and serve as block embedding in each node. The edge feature would be the call type (intra-function or inter-function) which is incorporated into the edge embedding for the explanation part. The GCNN employs a message-passing mechanism to generate an enhanced binary file representation of the input graph and then predicts whether the input file is vulnerable or not. In a GCNN, each node which represents a block receives information from its neighbors and accumulates the vectors to create a block embedding.

Finally, the node embeddings and the edge embeddings are used to create the edge distributions and create an explanatory subgraph. Using the edge distribution, we can determine the weight of the information coming from each neighbor. To compute the TFIDF value of each block, we assume each block is a word and employ TFIDF on the whole binary file. Using this approach, we can capture the semantic relationship between the blocks, and similar blocks would have a TFIDF value at a proximity. BERT embedding is applied to the instructions in a block and then an average is taken between the embeddings of the instructions. Using pre-trained BERT methods, we can ensure to preserve the semantics between different tokens in assembly instructions so that similar basic blocks have embeddings close to each other. For the operand types, we calculate the frequency of each distinct operand type and create a bag-of-words type of feature vector. There exist seven operand types: (0) No Operand, (1) General Register (al, ax, es, ds...) (reg), (2) Direct Memory Reference (DATA) (addr), (3) Memory Ref [Base Reg + Index Reg] (phrase), (4) Memory Reg [Base Reg + Index Reg + Displacement] (phrase + addr), (5) Immediate Value (value), (6) Immediate Far Address (CODE) (addr), and (7) Immediate Near Address (CODE) (addr).

4 Experimental Results

4.1 Dataset

The NDSS18 dataset originated from the National Institute of Standards and Technology (NIST) and the Software Assurance Reference Dataset (SARD) project. It includes a total of 32,281 binary files, which can be categorized into Windows and Linux platforms and are associated with two specific Common Weakness Enumerations (CWEs), CWE-119 and CWE-399. Notably, the dataset is well balanced in terms of vulnerability labels both the Windows and Linux platforms. CWE-399 is related to a specific type of software vulnerability, known as “Resource Management Error” and CWE-119 is related to improper restriction of operations within the bounds of a memory buffer. The Juliet Test Suite is the second dataset we use that contains 118 distinct CWEs. This synthetic dataset comprises a total of 83,624 binary files, with an equal distribution of labels between vulnerable and non-vulnerable files. We analyzed our model on CWE-121 and CWE-190 from the Juliet Test Suite and on CWE-119 from the NDSS18 dataset as benchmarks. CWE-121 (Stack-based Buffer Overflow) is related to situations where excessive data is written to a stack buffer, potentially leading to a buffer overflow. Moreover, CWE-190 (Integer Overflow or Wraparound) involves errors in the calculation of integer values, leading to overflow or wraparound. A graph structure proves to be more appropriate for identifying and understanding these vulnerabilities, due to multiple functions. Table 1 shows the distribution of both datasets in terms of vulnerable and non-vulnerable samples. To normalize the assembly instructions, the following steps have been done on the instructions in each block: (1) removing the heading memory address, (2) replacing constants with 'const', e.g., '0Bh' is replaced with the word 'const', and (3) replacing effective addresses with 'addr', e.g., $[ebp + Var\_C]$ is replaced with the word 'addr'. Skipping the instruction normalization step would result in generating a huge list of distinct vocabularies for the language modeling task and not allowing the BERT algorithm to extract well-represented block embeddings.

Table 1. Dataset distributions

Dataset	# Vulnerable samples	# Non-vulnerable samples
NDSS18	8978	8999
Juliet Test Suite	7060	7060

4.2 Experimental Setup

The experiments were performed on an AMD Ryzen Threadripper PRO 3975WX 32-Core 3.50 GHz with 512 GB RAM running Windows Server 2022. VulEXplaineR was implemented using Python and Pytorch. We used the PGExplainer Github code repository to implement VulEXplaineR and revised the code[1] for pre-processing of the binary files (instruction normalization, creating block embedding, generating CFGs, generating node embedding), adding edge distributions, and computing fidelity metrics. To disassemble the binary files, we used IDA Pro version 8.2.230124. For implementing BERT models, 'bert-mini' from Hugging Face was employed, which is one of the smaller Pytorch pre-trained BERT variants. The graph convolutional layers were implemented using the GCNConv library from 'torch_geometric.nn'.

To evaluate the performance of our model, we split our dataset into training (80%), validation (10%), and testing (10%). The vulnerability detection task is a binary classification that classifies binary files as vulnerable (positive) or non-vulnerable (negative). We add explainability on top of the classification model by generating subgraphs of the initial graph. We use accuracy, recall, precision, and F1 score as evaluation metrics to calculate the classification performance. To assess explainability, we compute the classification performance of the generated

---

[1] https://github.com/Sam-Mah/VulExplainer.

subgraphs. Also, we use the fidelity metric to show the ratio of test examples that both GCNN and VulEXplaineR can agree on for the classification result.

Hyperparameter Fine-Tuning. We used cross-entropy loss for the loss function and Adam as the optimization algorithm for GCNN. We managed to fine-tune the learning rate in different ranges of [0.1, 0.01, 0.001, 0.0001]. More fine-tuning was conducted for the batch size [64, 100, 150, 200] and the number of epochs [200, 400, 600, 1000]. We chose the best setting that resulted in the best accuracy in the validation set. Thus, 0.001, 100, and 600 were selected as the best settings for the learning rate, batch size, and number of epochs, respectively. We needed to fine-tune the architecture of the GCNN including the convolutional layers. The best model consists of three stacked GCN layers followed by a linear layer. Max and Mean pooling is then used between the GCN outputs and the linear layers.

Models to Compare. We compare two variants of our model with two different combinations of node features with GCNN as the base GNN for both variants, VulEXplaineR$_{FS1}$ and VulEXplaineR$_{FS2}$. The first group of node features (FS1) includes BERT embedding, operand types, and TFIDF Value, while the second group (FS2) comprises operand types and TFIDF Value. The baseline models are VulGCNN$_{FS1}$ which is a vulnerability detector using a GCNN and employs BERT embedding, operand types, and TFIDF Value as the node features, VulTFIDF$_{RF}$ that uses Random Forest (RF) as a classifier and TFIDF embeddings of all blocks as the feature vector, VDGraph2Vec [5] with GCNN, VulANalyzeR [10], and i2v-TCNN [5] which uses Instruction2Vec [9] for embedding the assembly instructions and TextCNN for classifying the samples into benign and vulnerable.

4.3 Results

In Tables 2 and 3, we compare both variants of VulEXplaineR with the baseline algorithms VulGCNN$_{FS1}$, VulTFIDF$_{RF}$, VDGraph2Vec, i2v-TCNN, and VulANalyzeR on the NDSS18 and Juliet Test datasets, respectively. For the baseline algorithms, namely VDGraph2Vec, i2v-TCNN, and VulANalyzeR, we chose the setting with the best results in the papers. Both variants of VulEXplaineR are the only models that provide full explainability for the vulnerability detection task. VulANalyzeR provides a local-level explanation at the instruction level, while VulEXplaineR provides a global-level explanation, which preserves the local fidelity. As shown in Table 2, VulEXplaineR$_{FS1}$ achieves an accuracy of 96.02% and precision 97.31%, 0.54% higher accuracy and 1.66% higher precision acquired from the best performing baseline algorithm, i.e., VDGraph2Vec. As for other measures such as recall and F1, there is a negligible difference of $\approx$ 0.2% and 0.4%. Considering the fact that VDGraph2Vec does not provide any level of explainability, we can conclude that VulEXplaineR is highly efficient in terms of all measures. Similarly, in Table 3, VulEXplaineR provides nearly the same performance of the best performing baseline model, VDGraph2Vec, which lacks explainability. Overall, the results are representative of the fact that VulEXplaineR is capable of reproducing the accuracy of the GNNs from which it was derived.

Table 2. Performance results on the NDSS18 dataset

Models	Accuracy	Recall	Precision	F1
VulEXplaineR$_{FS1}$	**96.02**	**96.02**	**97.31**	**95.51**
VulEXplaineR$_{FS2}$	78.75	78.75	79.67	78.35
VulGCNN$_{FS1}$	93.02	93.02	95.81	92.99
VulTFIDF$_{RF}$	70.59	70.59	70.59	70.46
VDGraph2Vec	95.48	96.21	95.65	95.92
i2v-TCNN	81.41	82.50	83.72	83.11
VulANalyzeR	89.53	94.18	85.36	90.10

Table 3. Performance results on the Juliet Test dataset

Models	Accuracy	Recall	Precision	F1
VulEXplaineR$_{FS1}$	**99.80**	**99.80**	**99.81**	**99.80**
VulEXplaineR$_{FS2}$	94.79	96.65	93.40	95.00
VulGCNN$_{FS1}$	98.55	98.55	98.56	98.55
VulTFIDF$_{RF}$	87.47	87.47	88.25	87.42
VDGraph2Vec	100	100	100	100
i2v-TCNN	92.81	93.1	93.59	93.31
VulANalyzeR	99.68	100	99.38	99.69

4.4 Graph Explanation

In this section, we present the results of two sets of experiments, namely qualitative and quantitative, to evaluate the quality of the graph explanations. In quantitative evaluations, we calculate the fidelity of the subgraphs to show how much it mimics the behaviour of the original graph it is extracted from. The qualitative evaluation is the ground truth analysis of the extracted subgraph with respect to the expert knowledge. In this task, a reverse engineer thoroughly examines the subgraph associated with the Buffer Overflow Vulnerability (CWE 121) to validate the extracted subgraph and assess its level of representation, compared to the original graph from which it is derived.

Fidelity Analysis. Fidelity refers to the degree to which the predictions of the extracted subgraphs align with the actual GNN. The higher the fidelity metric, the more faithful the extracted subgraph is to the original graph. Fidelity ($\mathcal{F}$) values range between zero and one, based on Eq. 5:

$$\mathcal{F} = cnt/len(\mathcal{O}_p) \ , \tag{5}$$

where cnt is the number of the times $(\mathcal{O}_p - \mathcal{M}_p)$ vector is zero. $\mathcal{O}_p$ are the prediction vectors based on the original graph and $\mathcal{M}_p$ are the prediction vectors based on the current explanation, on the test dataset. The fidelity values of VulEXplaineR$_{FS1}$ and VulEXplaineR$_{FS2}$ on the NDSS18 dataset and the Juliet Test Suite are depicted in Table 4. The results show that the extracted subgraph and the underlying GNN highly agree with the predictions they make, except for the scenario with the lightweight feature set *FS2* (operand type frequency and TFIDF) on the NDSS18 dataset that contains non-synthetic vulnerabilities.

Table 4. Fidelity of VulEXplaineR

Dataset	$VulEXplaineR_{FS1}$	$VulEXplaineR_{FS2}$
NDSS18	0.95	0.80
Juliet Test Suite	0.99	0.99

Ground Truth Analysis of Buffer Overflow Vulnerabilities. Buffer overflow vulnerabilities are a common type of security weakness in software programs that can be exploited by attackers to execute their own code or overwrite critical data. In this paper, we focus specifically on buffer overflow vulnerabilities caused by improper bounds checking, weak input validation, and lack of stack protection. In addition to detecting whether a program is vulnerable to buffer overflow attacks, we provide explainability by extracting a subgraph from the CFG and offering justifications for vulnerability. We also link each vulnerability to the CWE list, which is a community-developed list of software and hardware weaknesses.

Summary of Extracted Subgraph. VulEXplaineR extracted a subgraph of the main CFG of a vulnerable binary, as shown in Fig. 4[2]. In these explanations, bold black edges indicate top-k edges ranked by their importance weights, where k is set to 50 by hyper-parameter fine tuning. The disassembled code consists of several blocks of instructions, each labeled with an address in memory. There are function call names in the code blocks that indicate the vulnerability. We have removed these debug symbols before conducting experiments. The first block at memory address 4096 checks if a certain value is null and calls a function if

---

[2] Details about the subgraph and the code are provided in the supplementary data.

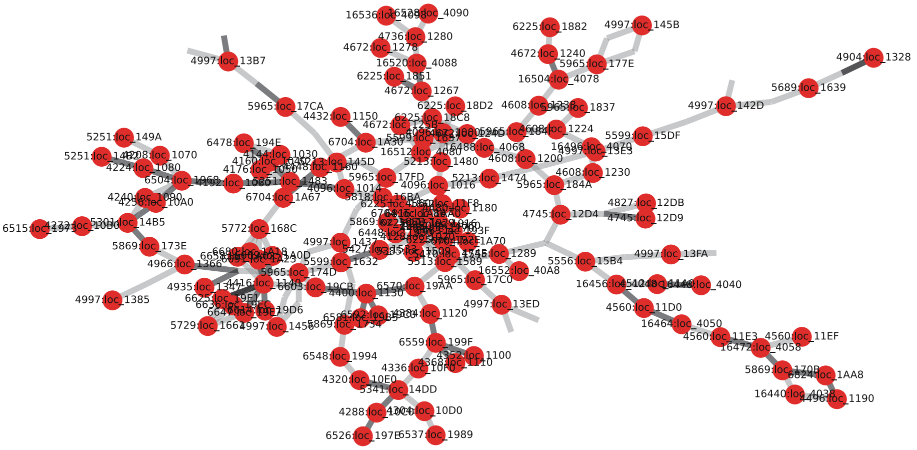


Fig. 4. Extracted explanations as a subgraph indicated by bold black edges calculated as top-*k* edges. Each node of the graph is a block represented as *function id: block name*. (Color figure online)

it is not. The next several blocks starting at memory address 4128 and ending at memory address 4496 seem to be setting up some constants. The block at memory address 4512 appears to be the main function, which initializes some values and calls other functions. The blocks at memory addresses 4560 and 4608 appear to be functions that manipulate the data in the BSS section of the program. The block at memory address 4672 appears to be a function that calls other functions to clean up the program. The block at memory address 4736 appears to be a function that registers some clones. Providing a justification or ground truth for a subgraph extracted from a model is challenging in the absence of complete code details. To determine the actual occurrences and functions that can serve as a backdoor for an attacker, it may be necessary to execute the code in a sandbox environment with the aid of an executable file. However, this approach is not always feasible. An alternative approach involves a manual reading of the extracted code to identify vulnerabilities. Using this method, we establish a ground truth for the explainability extracted from the model. A list of vulnerable functions along with their vulnerable basic blocks is described as follows:

```
4827:loc_12DB
['endbr64', 'push rbp', 'mov rbp rsp', 'sub rsp const', 'mov addr edi', 'mov addr rsi',
    'mov edi const', 'call _time', 'mov edi eax', 'call _srand', 'lea rdi acallingbad',
     'call println', 'mov eax const', 'call
    cwe121_stack_based_buffer_overflow__cwe805_struct_declare_loop_54_bad', 'lea rdi
    afinishedbad', 'call println', 'mov eax const', 'leave', 'retn']
```

This function block has instructions that push values onto the stack `'push rbp'`, `'sub rsp const'`, `'mov addr edi'`, `'mov addr rsi'`, `'lea rdi acallingbad'`, `'call println'`, `'mov eax const'` and call other functions that could write to the stack without checking the size of the buffer, such as ‘printline’. This leaves the program vulnerable to a stack-based buffer overflow attack. The function `cwe121_stack_based_buffer_overflow__cwe805_struct_declare_loop_54_bad` describes a weakness related to stack-based buffer overflows (CWE121), where a buffer overflow can occur if an attacker can cause more data to be written to a buffer than it can hold [2]. In this scenario, the weakness is caused by a loop that reads input into a structure, with no validation to ensure that the input does not exceed the size of the structure's buffer. This can lead to the overwriting of adjacent memory areas and cause unexpected behavior or crashes [2].

```
4935:loc_1347
[‘endbr64’, ‘push rbp’, ‘mov rbp rsp’, ‘sub rsp const’, ‘mov addr rdi’, ‘mov rax addr’,
     ‘mov rdi rax’, ‘call
     cwe121_stack_based_buffer_overflow__cwe805_struct_declare_loop_54d_badsink’, ‘nop’,
      ‘leave’, ‘retn’]
```

```
4904:loc_1328
[‘endbr64’, ‘push rbp’, ‘mov rbp rsp’, ‘sub rsp const’, ‘mov addr rdi’, ‘mov rax addr’,
     ‘mov rdi rax’, ‘call
     cwe121_stack_based_buffer_overflow__cwe805_struct_declare_loop_54c_badsink’, ‘nop’,
      ‘leave’, ‘retn’]
```

```
4966:loc_1366
[‘endbr64’, ‘push rbp’, ‘mov rbp rsp’, ‘sub rsp const’, ‘mov addr rdi’, ‘mov rax addr’,
     ‘mov rdi rax’, ‘call
     cwe121_stack_based_buffer_overflow__cwe805_struct_declare_loop_54e_badsink’, ‘nop’,
      ‘leave’, ‘retn’]
```

The above function blocks have instructions that push values onto the stack `‘push rbp’, ‘mov rbp rsp’, ‘sub rsp const’, ‘mov addr rdi’` and call `cwe121_stack_based_buffer_overflow__cwe805_struct_declare_loop_54c_bad` which could potentially write to the stack without checking the size of the buffer (CWE-805). This leaves the program vulnerable to a stack-based buffer overflow attack. The function `cwe121_stack_based_buffer_overflow__cwe805_struct_declare_loop_54c_badsink` describes a similar scenario where a bad sink is present, causing the same vulnerability (CWE-805) [1]. A sink is any place in a program where data is received from an untrusted source, and a bad sink is one that does not properly validate or sanitize the incoming data, allowing attackers to inject malicious input [1].

```
4096:loc_1000
[‘endbr64’, ‘sub rsp const’, ‘mov rax cs:__gmon_start___ptr’, ‘test rax rax’, ‘jz
     loc_1016’]
```

The above function has an instruction that jumps to `__gmon_start___ptr`, which could potentially be modified to point to an attacker-controlled address. If this happens, an attacker could execute arbitrary code by crafting a specially-crafted object file that causes the program to jump to the attacker-controlled address.

```
4672:loc_125B
[‘mov rdi cs:__dso_handle’, ‘call __cxa_finalize’]
```

This function has an instruction that calls `__cxa_finalize`, passing in a value from `__dso_handle`. If attackers can overwrite the value of `__dso_handle`, they could execute arbitrary code by crafting a specially-crafted shared library that causes the program to execute the attacker's code when `__cxa_finalize` is called.

```
4512:loc_11A0
[‘endbr64’, ‘xor ebp ebp’, ‘mov r9 rdx’, ‘pop rsi’, ‘mov rdx rsp’, ‘and rsp const’, ‘
     push rax’, ‘push rsp’, ‘lea r8 __libc_csu_fini’, ‘lea rcx __libc_csu_init’, ‘lea
     rdi main’, ‘call cs:__libc_start_main_ptr’, ‘hlt’]
```

The above function has instructions that push values onto the stack `push rsp, lea r8 __libc_csu_fini, lea rcx __libc_csu_init, lea rdi main` and call other functions that could write to the stack without checking the size of the buffer, such as printf, scanf, and wprintf. This leaves the program vulnerable to a stack-based buffer overflow attack.

Mainly, two functions `cwe121_stack_based_buffer_overflow__cwe805_struct_declare_loop_54_bad` and `cwe121_stack_based_buffer_overflow__cwe805_struct_declare_loop_54c_badsink` both belong to CWE lists [1,2] and describe similar weaknesses related to stack-based buffer overflows caused by inadequate input validation, with one involving a loop and the other involving a bad sink. To mitigate such vulnerabilities, it is essential to validate all user input to ensure that it does not exceed the size of the buffer and that it is correctly sanitized to remove any potentially harmful content.

5 Conclusion

In this paper, we propose VulEXplaineR, an XAI method for vulnerability detection in assembly code. Utilizing BERT and TFIDF, it offers an efficient framework to represent relationships between blocks and functions. Inspired by PGExplainer, VulEXplaineR produces explanations in the form of subgraphs of GCNNs, incorporating edge embeddings for enhanced accuracy. Experimental results on the NDSS2018 and Juliet Test datasets show that VulEXplaineR outperforms state-of-the-art baselines, providing high explainability that matches the graph nature of the assembly code and is valuable for reverse engineers. Qualitative and quantitative evaluations, including fidelity metrics, demonstrate the method's effectiveness. A case study highlights VulEXplaineR's ability to identify vulnerabilities and dependencies within the extracted subgraph. One of the directions of future work is to design motifs as the ground truth to conduct a quantitative evaluation of the extracted subgraph in the form of binary edge classification. Those edges that fall inside the motifs are positive edges, and those that fall outside the motifs are negative edges. Another line of research would be to model the underlying GNN as a directed graph. A directed graph imposes an ordering on a pair of nodes that is useful, as it further describes the relationship between the nodes.

Acknowledgments. This research is supported by BlackBerry Limited, Defence Research & Development Canada, and NSERC Alliance Grants (ALLRP 561035-20).

References

1. Cwe-805: Buffer access with incorrect length value. https://cwe.mitre.org. Accessed 2024
2. Cwe-121: Stack-based buffer overflow. https://cwe.mitre.org. Accessed 2024
3. Cao, S., Sun, X., Bo, L., Wei, Y., Li, B.: Bgnn4vd: constructing bidirectional graph neural-network for vulnerability detection. Inf. Softw. Technol. **136**, 106576 (2021)
4. Dahl, W.A., Erdodi, L., Zennaro, F.M.: Stack-based buffer overflow detection using recurrent neural networks. arXiv preprint arXiv:2012.15116 (2020)
5. Diwan, A., Li, M.Q., Fung, B.C.: Vdgraph2vec: vulnerability detection in assembly code using message passing neural networks. In: 21st International Conference on Machine Learning and Applications (ICMLA), pp. 1039–1046. IEEE (2022)
6. Garcia, F.D., de Koning Gans, G., Verdult, R.: Vulnerability extrapolation: assisted discovery of vulnerabilities using machine learning (2011)
7. Harer, J.A., et al.: Automated software vulnerability detection with machine learning. arXiv:1803.04497 (2018)
8. Hovsepyan, A., Scandariato, R., Joosen, W., Walden, J.: Software vulnerability prediction using text analysis techniques. In: 4th International Workshop on Security Measurements and Metrics, pp. 7–10 (2012)
9. Lee, Y., Kwon, H., Choi, S.H., Lim, S.H., Baek, S.H., Park, K.W.: Instruction2vec: efficient preprocessor of assembly code to detect software weakness with CNN. Appl. Sci. **9**(19), 4086 (2019)
10. Li, L., et al.: Vulanalyzer: explainable binary vulnerability detection with multi-task learning and attentional graph convolution. ACM Trans. Priv. Secur. **26**(3), 1–25 (2023)
11. Li, Y., Wang, S., Nguyen, T.N.: Vulnerability detection with fine-grained interpretations. In: 29th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, pp. 292–303 (2021)
12. Li, Z., et al.: Vuldeepecker: a deep learning-based system for vulnerability detection. arXiv:1801.01681 (2018)
13. Liu, Y., et al.: Roberta: a robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692 (2019)
14. Luo, D., et al.: Parameterized explainer for graph neural network. Adv. Neural. Inf. Process. Syst. **33**, 19620–19631 (2020)
15. Madsen, A., Reddy, S., Chandar, S.: Post-hoc interpretability for neural NLP: a survey. ACM Comput. Surv. **55**(8), 1–42 (2022)
16. Pang, Y., Xue, X., Namin, A.S.: Predicting vulnerable software components through n-gram analysis and statistical feature selection. In: 14th International Conference on Machine Learning and Applications (ICMLA), pp. 543–548. IEEE (2015)
17. Russell, R., et al.: Automated vulnerability detection in source code using deep representation learning. In: 17th International Conference on Machine Learning and Applications (ICMLA), pp. 757–762. IEEE (2018)
18. Ying, Z., Bourgeois, D., You, J., Zitnik, M., Leskovec, J.: Gnnexplainer: Generating explanations for graph neural networks. Adv. Neural Inf. Process. Syst. **32** (2019)

19. Zhou, J., et al.: Graph neural networks: a review of methods and applications. AI open **1**, 57–81 (2020)
20. Zhou, Y., Liu, S., Siow, J., Du, X., Liu, Y.: Devign: effective vulnerability identification by learning comprehensive program semantics via graph neural networks. Adv. Neural Inf. Process. Syst. **32** (2019)
21. Zou, D., Zhu, Y., Xu, S., Li, Z., Jin, H., Ye, H.: Interpreting deep learning-based vulnerability detector predictions based on heuristic searching. ACM Trans. Softw. Eng. Methodol. (TOSEM) **30**(2), 1–31 (2021)

Guiding Catalogue Enrichment with User Queries

Yupei Du[1](✉), Jacek Golebiowski[2], Philipp Schmidt[2], and Ziawasch Abedjan[3]

[1] Utrecht University, Utrecht, The Netherlands
`y.du@uu.nl`
[2] Amazon, Berlin, Germany
`{jacekgo,phschmid}@amazon.com`
[3] BIFOLD and TU Berlin, Berlin, Germany
`abedjan@tu-berlin.de`

Abstract. Techniques for knowledge graph (KGs) enrichment have been increasingly crucial for commercial applications that rely on evolving product catalogues. However, because of the huge search space of potential enrichment, predictions from KG completion (KGC) methods suffer from low precision, making them unreliable for real-world catalogues. Moreover, candidate facts for enrichment have varied relevance to users. While making correct predictions for incomplete triplets in KGs has been the main focus of KGC method, the relevance of when to apply such predictions has been neglected. Motivated by the product search use case, we address the angle of generating relevant completion for a catalogue using user search behaviour and the users property association with a product. In this paper, we present our intuition for identifying enrichable data points and use general-purpose KGs to show-case the performance benefits. In particular, we extract entity-predicate pairs from user queries, which are more likely to be correct and relevant, and use these pairs to guide the prediction of KGC methods. We assess our method on two popular encyclopedia KGs, DBPedia and YAGO 4. Our results from both automatic and human evaluations show that query guidance can significantly improve the correctness and relevance of prediction.

1 Introduction

Knowledge graphs (KGs) have become increasingly prevalent in commercial applications to provide accessible and structured representation of knowledge. For example, shopping websites like Amazon often use KGs to represent product catalogs [6], where product properties and taxonomies are captured in structures similar to the Resource Description Framework (RDF). For instance, "the color of blouse A is red" would be represented as the subject entity "blouse A" connecting with the object entity "red" via the predicate "color." These properties can then be used to offer users recommendations and navigation options during

---

Y. Du and Z. Abedjan—Work done while at Amazon.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
A. Bifet et al. (Eds.): ECML PKDD 2024, LNAI 14949, pp. 21–37, 2024.
https://doi.org/10.1007/978-3-031-70378-2_2

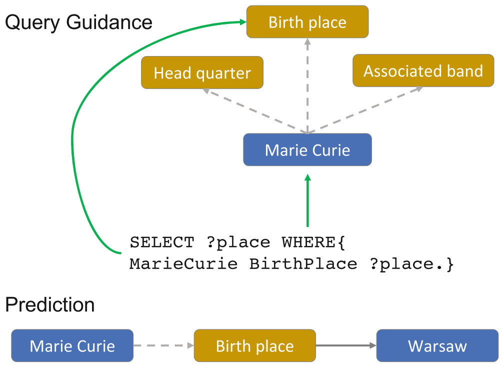


Fig. 1. An example of using query logs to guide prediction. In this example, we can make prediction on the entity "Marie Curie" using one of the predicates from "birthplace", "head quarter", and "associated band". Because the query selects the birthplace of Marie Curie, we make predictions from this entity-predicate pair.

product search (e.g., recommended categories and filtering widgets on the top and the left side of the Amazon product search page).

Despite the wide usage of KGs in industry, the dynamic nature of commercial applications leads to many practical problems. For example, sellers may frequently introduce new products without providing the necessary attributes or new markets might require different attributions than previous launched markets, impeding the maintenance of these KGs. One approach to remedy this issue is to automatically infer missing information. In KG management, this approach is known as KG completion (KGC), in which missing information refers to missing triplets. Different types of KGC methods have been proposed [30], including methods based on mining rules [7,8,14,15], embeddings [2,17,23,25], and neural networks [5,20,21]. Among these approaches, KG embedding (KGE) methods have shown good scalability and effectiveness [4,26].

Limitations in KGC: Despite the significant advances, all of the aforementioned methods suffer from two major issues when predicting missing triplets. First, there is a huge space of possible triplets when considering all possible combinations of entities and predicates. Formally, there are $|\mathcal{E}| \times |\mathcal{P}| \times |\mathcal{E}|$ possible triplets just by recombination of existing entities and properties, where $|\mathcal{E}|$ and $|\mathcal{P}|$ are respectively the number of entities and predicates in the KG. This huge search space makes missing triplet prediction on large-scale KGs usually show low precision [19]. Second, KGC methods mostly focus on ensuring the correctness of their predictions, by adopting maximum likelihood estimation style objectives. However, in real-world scenarios, different triplets are usually of different levels of relevance to users. With relevance, we refer to the appropriateness and importance of a triplet to describe the real-world. For example, although a blouse could have both a size and a manufacturing date, knowing its size is more useful for general users than knowing its manufacturing date. Therefore, it is also important to take into account which triples are more likely to be used, i.e., are more relevant for enrichment. *One strategy to mitigate both issues at the same time is to provide guidance during the prediction process.* For example, when already a correct and user-relevant pair of entity-predicates (e.g. blouse-color) are given, one can significantly reduce the search space (i.e., from $|\mathcal{E}| \times |\mathcal{P}| \times |\mathcal{E}|$ to $|\mathcal{E}|$).

In online retail systems, it is common to rely on behavioral signals from users to improve their experience. For instance, we can use users' clicks and purchases to infer their preferences towards various products. Regarding the prediction of missing triplets, the vast amount of user queries from the product search engine can be mined to extract preferences for product attributes. For the blouse example, we can compare the frequency of queries for "<color> blouse" against those for "blouse released on <date>". This data can help make grounded decisions about the relevance and correctness of possible triplets.

Contributions: In this paper, we propose to guide the missing triplet prediction process using user query log signals that express user interests to improve the correctness and relevance of the predicted triplets. Because commercial KGs and query logs are usually confidential (e.g., Amazon product catalogue), we show the suitability of our approach on public general-purpose KGs using their corresponding SPARQL query logs instead. User SPARQL queries usually search for information on general-purpose KGs that are relevant to the user and considered correct [1]. Thus, they exhibit information on how entities should relate to each other and what properties they should display. Figure 1 shows a real example from YAGO 4 [18], illustrating that the existence of queries can help as a heuristic to evaluate the correctness and relevance of properties: for the entity "Marie Curie", users often query for her "birthplace", instead of for her "headquarter" (incorrect) or for her "associated band" (taxonomically correct but less relevant for the specific entity "Marie Curie", because Marie Curie is famous for her scientific rather than musical contributions). Although we experiment with general-purpose KGs and query logs in this paper, our approach can be easily adapted to commercial KGs. Concretely, we make three contributions:

- We propose a simple and efficient method for guiding missing triplet prediction using user queries. We first develop a baseline without user guidance that relies on rejection sampling methods. We then present our query guidance approach for RDF-based KGs (Sect. 3). Our query guidance method can complement *any* KGC method that make predictions from entity-predicate pairs, which covers most popular KGC methods, including rule-based, KGE, and neural network methods. Our approach can also work with *any* RDF-based KGs, which covers most general encyclopedia KGs and product catalogues.
- We empirically illustrate the benefits of incorporating query guidance. Specifically, we compare our query-guidance method to three baselines: our own baseline that employs rejection sampling without guidance, as well as versions that incorporate two alternative types of guidance, namely KG metadata and KGE scores. This comparison is carried out on two popular general-purpose KGs: DBPedia [11] and YAGO 4 [18], using the popular RotatE

KGE model [23]. Our results from both automatic and human evaluation show that query guidance effectively benefits missing triplet prediction, by selecting entity-predicate pairs that are at least two times more likely to be correct, compared to our baseline without guidance (Table 2).
- We build and open-source a dataset consisting of 1600 entity-predicate pairs that are annotated with correctness and relevance scores (Sect. 4)[1].

2 Background and Related Work

In this section, we describe the relevant studies of KGs, KGE models, and rule-based KGC approaches. We also introduce RotatE, which is the KGE model used in this paper. We further include the notation system used in this paper.

Knowledge Graphs. KGs are structural representations of human knowledge in the form of triplets $\mathcal{G} = \{(h, r, t)\}$, where h, r and t are respectively the subject entity, predicate, and object entity [30]. For example, “Marie Curie was born in Warsaw” will be represented as (“Marie Curie”, “born in”, “Warsaw”). Different types of KGs exist, including encyclopedia KGs (e.g., DBPedia [11] and YAGO 4 [18]), domain-specific KGs (e.g., Drugbank [28] and semantic scholar [13]), and task-specific KGs (e.g., Amazon product graph [6]).

Knowledge Graph Embeddings and RotatE. Various KGE models are proposed in previous studies, including translation models [2,12,23], tensor decomposition models [10,25] and deep learning models [9,27]. These KGE models usually encode entities and predicates in KGs as dense vectors (i.e., embeddings), which can be used as prior knowledge for downstream tasks [22,24,31], or to predict missing triplets in KGs [2,5].

In this paper, we focus on RotatE [23] KGE model. For each triplet (h, r, t), RotatE measures the distance between h and t in the space of r with $d_r(\mathbf{h}, \mathbf{t}) = \|\mathbf{h} \circ \mathbf{r} - \mathbf{t}\|$, where $\mathbf{h}$, $\mathbf{r}$ and $\mathbf{t}$ are the embedding vectors of h, r and t, and $\circ$ is the element-wise product. Similar to other KGE models, RotatE adopts a margin loss with negative sampling (to facilitate convergence), to minimize the distance within existing triplets,

$$L = -\log \sigma \left(s_r(\mathbf{h}, \mathbf{t}) \right) - \sum_{i=1}^{n} \frac{1}{k} \log \sigma \left(-s_r \left(\mathbf{h}'_i, \mathbf{t}'_i \right) \right), \tag{1}$$

where $\mathbf{h}'_i$ and $\mathbf{t}'_i$ are the randomly-sampled negative samples, n and k are the number and weights of the negative samples respectively, and $s_r(\mathbf{h}, \mathbf{t}) = \gamma - d_r(\mathbf{h}, \mathbf{t})$ is the margin-based score function with a margin γ. One useful attrbute of this objective function is that $s_r(\mathbf{h}, \mathbf{t})$ is proportional to $\log p(\mathbf{h}, \mathbf{t}|\mathbf{r})$, which is illustrated in [16] and [29].

---

[1] Publicly available at https://github.com/LUH-DBS/KGEnrichment.

Rule-Based Knowledge Graph Completion. Besides KGE models, rule-based approaches are also popular in KGC, which usually mine compositional rules from statistical cues. For example, knowing that a person was born and raised in Amsterdam, while also having the knowledge that the official language of Amsterdam is Dutch, one can infer that likely this person speaks Dutch. Both top-down, which begins from general rule prototypes and specified with data, e.g., AMIE [7,8], and bottom-up, which begins from specific triplets and generalize to rules, e.g. AnyBURL [14,15], are commonly used. Compared with KGE models, rule-based models are more interpretable but *less scalable*, making them hard to apply to large-scale product catalogues.

3 Query-Guided Triplet Prediction

The goal of guided triplet prediction is to increase the utility and accuracy of KGC by focusing on triplets that are of interest to users and avoiding the generation of potentially irrelevant triplets. We first introduce a rejection-sampling-based baseline method for predicting missing triplets from KGE models. Afterwards, we propose a simple yet effective method for guiding the prediction of missing triplets with user query logs to obtain triplets of better correctness and relevance. In our experiments (Sect. 4), we apply our query guidance method on the baseline method and compare the prediction quality with and without guidance.

3.1 Prediction from KGE Using Rejection Sampling (RS)

Rejection sampling is a technique for drawing samples from a complex distribution, whose unnormalized probability can be expressed as a calculable score $z \cdot p(x)$, where $p(x)$ is the probability of a sample x, and z is a (mostly unknown) normalization factor. This method involves using a proposal distribution, denoted as $q(x)$, which is easy to sample from (e.g., a uniform distribution). Each sample drawn from $q(x)$ is then accepted with a probability

$$p(\text{accept}) = z \cdot p(x)/k \cdot q(x), \tag{2}$$

where k is a constant chosen such that $p(\text{accept}) \leq 1$ for all x. This choice ensures that $p(\text{accept})$ is well-defined.

Given a trained KGE model, because the embedding scores it assigns to triplets are proportional to the probabilities (i.e. $s_r(\mathbf{h}, \mathbf{t}) \propto \log p(\mathbf{h}, \mathbf{t}|\mathbf{r})$, [16,29]), one can predict missing triplets by sampling from the distribution of KGE scores. However, this sampling is not trivial due to that $s_r(\mathbf{h}, \mathbf{t})$ is not normalized. One direct fix is to sample entities and predicates uniformly, and filter out triplets with low $s_r(\mathbf{h}, \mathbf{t})$ by a pre-defined threshold (i.e., regarding triplets with high KGE scores as correct ones). However, this threshold can be difficult to determine, because the specific relationship between score and triplet quality is unclear.

As mentioned before, rejection sampling can be used to sample from complex distributions, as long as unnormalized probabilities are easy to compute, making it a good fit for sampling from KGE models. Specifically, we can take two steps to predict new triplets by rejection sampling. First, according the marginal distribution of predicates in the original KG, we sample a predicate r. Second, for this sampled predicate, we sample candidate entities subject h and object t from a uniform proposal distribution. We then accept the triplet (h, r, t) with probability $p(\text{accept}) = e^{s_r(\mathbf{h},\mathbf{t})}/e^{\gamma}$, where γ is the margin from the loss function of RotatE (c.f. Sect. 2). The rationales are that, 1) because $s_r(\mathbf{h},\mathbf{t})$ is proportional to $\log p(h, t|r)$, $e^{s_r(\mathbf{h},\mathbf{t})}$ is an estimation of the unnormalized probability $z \cdot p(h, t|r)$, and 2) since $s_r(\mathbf{h},\mathbf{t}) \leq \gamma$, e^{γ} can be seen as an unnormalized proposal uniform distribution whose value is greater or equal to $e^{s_r(\mathbf{h},\mathbf{t})}$ everywhere.[2]

3.2 Guided Prediction with Queries (QG)

Many KGs or catalogues are targets of exploratory search. Queries for exploratory search often reflect users' association with an entity, e.g., a product and its attributes. While a single user might not always hint at the correct signals, frequent appearances of certain queries are likely to mirror common expectations of the user base. Our intuition is to collect such queries and use them to identify gaps in the underlying dataset.

In this paper, we describe our methodology by referring to SPARQL language, which is a popular language for querying RDF data [3]. We make this choice because of the prevalent usage of SPARQL in querying general-purpose KGs, including the ones that we experimented with in this study.

SPARQL supports various functionalities, including `SELECT` (existing triplets), `CONSTRUCT` (new triplets), `ASK` (if a triplet exists), and `DESCRIBE` (an entity). Among them, `SELECT` queries are similar to actual queries appear in product search. `SELECT` queries usually consist of combinations of predicates and entities, where one connecting entity is missing as is queried for. For example, the query looking for the birthplace of Marie Curie, `SELECT ?place WHERE{MarieCurie BirthPlace ?place}`, would already include Marie Curie as the subject of the triplet and birthplace as its predicate. Based on our intuition, the existence of a query as such suggests that "Marie Curie" should have the attribute "birthplace", which is relevant to users. Similarly, product search users usually query for products of certain attributes, e.g., *red blouse.* This query suggests that all product items from the catalogue of type "blouse", should have a relevant attribute "color", knowing that *red* is a type of "color" (from named entity recognizers). Note that here we adopt pragmatic definitions for "correctness" and "relevance": queries show users' interests, and interests imply correctness and relevance (we will validate this heuristic in Sect. 4.3). Moreover, we observe that, for both KGs we use in this paper, **more than** 95% **of the**

---

[2] In practice, we can sample a large number of entities pairs and predicates simultaneously, and iterate until we accept the specified amount of triplets.

SPARQL queries in the query logs are SELECT queries. We therefore focus on using SELECT queries as guidance.

Specifically, based on our RS baseline, we perform three steps to predict new triplets with the guidance of `SELECT` queries. First, given a `SELECT` query, we extract all entity-predicate pairs from this query. Second, from a uniform proposal entity distribution, we sample the second entity for each entity-predicate pair Here, we focus on sampling the objects, because they are more relevant to the downstream use case of inferring product attributes. Third, we accept these sampled triplets based on their scores computed by the trained KGE model, following Eq. 2. Query guidance therefore help to reduce the prediction space from $|\mathcal{E}| \times |\mathcal{P}| \times |\mathcal{E}|$ to $|\mathcal{E}|$, which enhances prediction correctness and relevance, because they are within the scope of users' interests.

Comparison with Selecting Top-k Queries. Another approach of incorporating user query information is to select the top-k most frequent queries and make direct predictions from them. In contrast to this approach, our sampling-based approach additionally considers knowledge from the base KGC method, which is a representation of training KG information. Our approach can be extended for improved performance by considering additional aggregation or filtering of user queries. However, to illustrate our core idea, the effectiveness of user queries, we keep the simplest setting and leave further investigations to future work.

4 Evaluation and Results

In this section, we evaluate to what extent the guidance of user queries can help with missing triplet prediction. From the results of both automatic (Sect. 4.2) and human (Sect. 4.3) evaluations, we observe that query guidance can dramatically boost both the correctness and the relevance of the predicted triplets. Moreover, to better ground the impact of query guidance, we compare query guidance against two alternative types of guidance, namely KG metadata (i.e. taxonomy of entities and predicate types) and embedding scores from the KGE model (Sect. 4.4). We observe that, although these two types of guidance can both improve prediction quality, they are outperformed by query guidance.

4.1 Experimental Setup

We perform all our experiments using Amazon SageMaker, with a g5.16xlarge instance. We use Python 3.7, PyTorch 1.13, DGL 0.4.3, and DGLKE 0.1.2. We use RotatE [23] as the KGC model for prediction. It took around three GPU days (A10 Tensor Core GPU with 24 GB vRAM) to perform hyper-parameter optimization of the embedding models (20 times of random search on the validation set), and four GPU hours to produce all predictions (including the baselines).

KGs and Query Logs. We use two popular general-purpose RDF KGs for our experiments: DBPedia [11] English Wikipedia InfoBox 2020.07,[3] and the YAGO

---

[3] https://databus.dbpedia.org/dbpedia/mappings/mappingbased-objects/2020.07.01.

4 [18] English Wikipedia 2020.02.[4] Moreover, we use DBPedia SPARQL Query Logs from March 2021[5] and YAGO SPARQL Query Logs from 2022[6], which were the latest ones available at the time of experiments, and we removed the queries for entities and predicates that does not exist yet by 2020.

Table 1. Automatic evaluation results. #Hit Triples refers to the number of overlapping triplets between predictions and test sets, and Pair Precision is the precision score of the predicted entity-predicate pairs on test sets: we observe that query guidance (QG) drastically improve the quality of missing triplet prediction, compared with the rejection sampling baseline (RS).

	DBPedia900K		YAGO5M	
	#Hit Triplets	Pair Precision	#Hit Triplets	Pair Precision
RS	3	0.0106	0	0.0214
QG	**743**	**0.3467**	**21**	**0.1610**

Pre-processing of KGs and Query logs. We sanitize the KGs by removing entities containing only URLs and numbers, or are lists of other entities (e.g. list of all players of a soccer team). Beyond conventional pre-processing, we remove all the triplets in which both the predicate and at least one entity do not occur in the query logs, because these triplets are less relevant to our study. For example, for the triplet ("Marie Curie", "birthplace", "Warsaw"), if neither "birthplace" nor at least one of "Marie Curie" and "Warsaw" appear in the query logs, we will remove this triplet. As a result, we obtain 1.35 million triplets, 881649 entities, and 83 predicates from DBPedia, and 12.92 million triples, 4.82 million entities, and 124 predicates from YAGO, and will mention them as DBPedia900K and YAGO5M in the remainder of this paper. We then randomly split the KGs into train (70%), dev (10%) and test (20%) sets.

As mentioned before, most queries in the logs are SELECT queries (> 95% for both KGs). For example, the SELECT query in Fig. 1 aims to select the triplets that contain the entity-predicate pair ("Marie Curie", "birthplace"). Following the method described in Sect. 3.2, we extract all entity-predicate pairs from these queries and use them as guidance. Similar to the pre-processing of KGs, we only keep the pairs of which both the entity and the predicate exist in the processed KGs. As a result, we obtain 11960 entity-predicate pairs for DBPedia900K,[7] and 4.84 million entity-predicate pairs for YAGO5M.

---

[4] https://yago-knowledge.org/data/yago4/en/2020-02-24/.

[5] https://devhub.openlinksw.com/pub/Support/44aa7c1b-bd61-4d61-8fef-4075094f62ed/.

[6] https://yago-knowledge.org/assets/log_20221206_CoQlevVOXUyh.gz.

[7] Wikipedia InfoBox is only a small fraction of the whole DBPedia KG, so most items from the query log is not querying the part of KG that we use.

Comparisons. We primarily show the benefits of query guidance (QG) by comparing it with the rejection sampling (RS) baseline. We also compare our approach with two alternative types of guidance, namely KG metadata and embedding score, to better ground the impact of query guidance (details in Sect. 4.4). *For each method, we predict 10 million triplets that are not in the train or dev set.*

4.2 Automatic Evaluation

We first assess the benefits of adopting query guidance by automatic evaluation. Specifically, we compute the precision of predictions on the test sets, and compare the results for QG against those for the RS baseline. We exclude recall scores because the same amount of different triplets are predicted for each method (i.e. recall is fully dependent of precision). In particular, we first evaluate the predicted full triplets. Afterwards, we discuss the limitations of evaluating full triplets, and include a different setup to evaluate the precision of entity-predicate pairs extracted from these predicted triplets.

Automatic Evaluation of Triplets. To evaluate the prediction of full triplets, we assess the numbers of overlapping triplets (*#Hit Triplets*), i.e., triplets that appear in both predictions and test sets. We refrain from using the traditional precision score, because of two reasons. First, because we predict the same number of triplets for each method, the proportions between the number of overlapping triplets is the same as the precision scores. Second, as mentioned before, the search space of predictions, especially for the RS baseline, is huge (e.g., over 60 trillions for DBPedia900K). This undesirable attribute can lead to very small precision ratios. Considering our relatively small test sets that represent a closed world, such numbers might be misleading, for being more vulnerable to noises.

We show the results in the *#Hit Triplets* columns in Table 1, and make two observations. First, query guidance drastically increases the number of hit triplets, i.e., from 3 to 743 on DBPedia900K and from 0 to 21 on YAGO5M. The most likely reason for such huge improvements is the vast reduction of the search space size, from $|\mathcal{E}| \times |\mathcal{P}| \times |\mathcal{E}|$ to $|\mathcal{E}|$, where $|\mathcal{E}|$ and $|\mathcal{P}|$ are respectively the number of entities and predicates in KG: concretely, the search spaces of possible triplets decrease for 6.77 million and 597.68 million times for DBPedia900K and YAGO5M. We note that the larger numbers of overlapping triplets on DBPedia900K, compared with YAGO5M, the larger KG, may originate from the same reason: the search space of YAGO5M is approximately 137 times larger than DBPedia900K. Second, both methods have rather low numbers of overlapping triplets (at most hundreds compared to 10 million predicted triplets). This observation is consistent with our intuition that KGE models usually cannot make accurate predictions on large KGs, highlighting the importance of using query guidance.

Automatic Evaluation of Entity-Predicate Pairs. The evaluation of full triplets has two major limitations. First, the quality of predicted full triplets of QG

Table 2. Human evaluation results. Correct and relevant columns show the precision scores of predicted entity-predicate pairs regarding correctness and relevance. R/C shows the percentage of relevant triplets in all correct triplets. We observe that 1) consistent with automatic evaluation, query guidance greatly improves prediction quality over the RS baseline; 2) guidance of both KG metadata (KM) and embedding score from KGE models (ES) are beneficial, but outperformed by query guidance; and 3) query guidance can also improve the relevant ratio among correct triplets.

	DBPedia900K			YAGO5M		
	Correct	Relevant	R/C	Correct	Relevant	R/C
RS	0.345	0.220	63.8%	0.305	0.225	73.8%
QG	**0.950**	**0.895**	**94.2%**	**0.990**	**0.850**	**85.9%**
ES	0.425	0.355	83.5%	0.345	0.235	68.1%
KM	0.750	0.685	91.3%	0.920	0.760	82.6%

depends on two factors: the quality of entity-predicate pairs from user queries, and the performance of the KGE model in predicting the second entities. It is thus difficult to isolate the benefits of using entity-predicate pairs as guidance. Second, as observed in the previous experiment, the numbers of overlapping full triplets between predictions and test set can be very low for large-scale KGs, which makes such comparisons vulnerable to noises. For example, our KGE model produces 0 and 21 overlapping triplets on YAGO5M using RS and QG respectively: it is difficult to understand to which extent QG actually improves the prediction accuracy, because such proportions can be susceptible to randomness. Moreover, predicting entity-predicate pairs themselves is meaningful for improving user experience of product search as well: shopping websites can notify vendors which missing product attributes are relevant to users, so that such information can be manually added, which can then be used for search navigation and recommendation.

To accommodate the previous considerations, we focus on comparing the entity-predicate pairs extracted directly from user queries against the ones extracted from the predicted triplets of other methods. Specifically, we extract all entity-predicate pairs from the predictions of each method, and calculate the precision scores on the test sets, i.e., the percentage of entity-predicate pairs that consist of at least one triplet from the test sets. We show the results in the *Pair Precision* columns in Table 1. Consistent with our observation on the full triplets, the guidance of user queries significantly boosts the prediction accuracy, by at least ~ 8 times. Besides, we observe that the gap between QG and RS is smaller compared with the results from the evaluation of full triplets. This observation implies that KGE models predict the second entities more accurately based on entity-predicates extracted from user queries, compared with based on the ones that are randomly sampled. In other words, *QG not only offers more correct and relevant entity-predicate pairs, but also helps KGE models predict better.*

4.3 Human Evaluation

Automatic evaluation has the drawback of closed-world assumption: because KGs are not complete, the triplets in the test sets are only a small fraction of all possible (missing) triplets. In other words, entity-predicate pairs that are absent from the test set can still be correct and relevant. To address this issue, we also conduct a human evaluation of entity-predicate pairs. In particular, for each method on each KG, we randomly select 200 entity-predicate pairs, manually annotate their correctness and relevance, and compute the precision scores.

We use the following general guidelines for annotation: 1) In **correct** entity-predicate pairs, the entities should be able to logically possess the attribute or relationship described by the predicate. An incorrect counterexample is ("saw rock" - "birthplace"), because saw rock, which is a rock in South Atlantic Ocean, is an inanimate object; and inanimate objects cannot have attributes like "birthplace". 2) In **relevant** entity-predicate pairs, the predicates should provide pertinent information about the entity in the context of the knowledge domain that the entities belong to. In other words, annotators should evaluate whether an average user querying the KG would find the predicate's information beneficial or essential to their query purpose. For example ("Starsailor" - "band member") is a relevant entity-predicate pair, because "Starsailor" is a rock band, and users would likely want to know the members of a band they're looking up. In contrast, ("William Bayliss" - "associated band") should be annotated as correct but irrelevant, because although "William Bayliss", as a person, can associate with a band, he is known for his physiology contributions, not his musical affiliations.

We show our human evaluation results in Table 2 (the rows for RS and QG). Besides the precision scores of predicted entity-predicate pairs regarding both correctness and relevance, we also include the percentage of pairs that are annotated as relevant in all pairs that are annotated as correct (R/C). We make two observations. First, consistent from our observations in automatic evaluations, we observe that query guidance can improve both correctness and relevance of the predictions by a large margin (i.e. from < 0.35 to ≥ 0.95 for correctness, and < 0.25 to ≥ 0.85 for relevance). Notably, besides the absolute numbers of correct and relevant entity-predicate pairs, QG also achieves better R/C, indicating that *query guidance is beneficial for the relevance of predictions, beyond merely enhancing the fraction of correct predictions.* Second, compared with the precision scores from our automatic evaluation (Table 1), we observe much higher values in human evaluation. We believe that this result validates our previous analyses on the closed-world issue of automatic evaluation: because KGs are not complete and many correct and relevant predicted entity-predicate pairs are not included in the test set, precision scores from automatic evaluation are actually lower estimations than reality.

4.4 Comparison with Other Types of Guidance

Beside user queries, there exist other types of information that can help identify helpful entity-predicate pairs to guide the missing triplets prediction. To better

Table 3. Automatic evaluation of KG metadata compatible and incompatible entity-predicate pairs. #Pairs refers to the number of overlapping pairs between predictions and test sets, and Precision is their precision scores: KG metadata guidance can help prediction, because compatible groups show higher precision than incompatible groups.

	DBPedia900K		YAGO5M	
	#Pairs	Pair Precision	#Pairs	Pair Precision
Incompatible	3553456	0.0078	8216555	0.0209
Compatible	424526	0.0337	139554	0.0488

ground the impact of query guidance, we compare our approach to two alternative types of guiding information, namely KG metadata (KM) and embedding score (ES). Consistent with our previous experiments, we assess them using both automatic and human evaluations. Our results show that, although both types of guidance can improve prediction correctness and relevance, they are outperformed by QG, highlighting the relative advantage of using query guidance.

*KG Metadata Guidance (**KM**).* Both KGs used in this paper provide metadata used to construct them. Concretely, they retain the type of each entity, and the domain and range of each predicate, i.e., which types of entities that the predicate can accept as its subject and object. The combination of these two types of metadata can help filter out incompatible entity-predicate pairs. For example, knowing the metadata that the predicate "largest city" can only accept the entity type "place" as subject can help us filter out the pair ("Marie Curie", "largest city"), because "Marie Curie" is of type "person" not "place". We therefore divide entity-predicate pairs extracted from the predicted triplets of the RS baseline into (KG metadata) compatible and incompatible groups, and then compute the precision score of each group on the test sets.

It is worth noting that, likewise the incompleteness of the KGs themselves as we have discussed, KG metadata can also be incomplete. In this case, "incompatible" entity-predicate pairs can still be correct or relevant. For example, for a entity-predicate pair ("Germany", "largest city"), if we only know "Germany" is a "country", and we do not have the metadata that "country" is always "place", we will categorize this pair as incompatible.

*Embedding Score Guidance (**ES**).* We also investigate whether embedding scores computed by KGE models (i.e. s in Eq. 1) can help us select correct and relevant entity-predicate pairs. Different from KM, which directly divide entity-predicate pairs into two separate groups (i.e., compatible and incompatible), embedding scores are continuous values. For clearer evaluation, we divide all entity-predicate pairs predicted by the RS baseline into 50 bins, based on the highest embedding score from the triplets that include each pair. For instance, if an entity-predicate pair appears in 10 different predicted triplets, we use the triplet with the highest score to determine the bin for that pair. Similar to KM, we then compute the precision score of each group on the test sets.

Usage of KM and ES. In contrast to the query guidance approach, neither KM nor ES directly offer entity-predicate pairs for KGE models to make predictions on. Instead, they provide guidance in a post-hoc way, by either judging whether a predicted entity-predicate pair is compatible with KG metadata (KM), or assigning this pair a continuous embedding score (i.e. s in Eq. 1), whose quantity indicates how likely this pair is correct (ES). Therefore, we apply them on the 10 million triplets predicted by the RS baseline as filters to select more possible triplets and entity-predicate pairs.

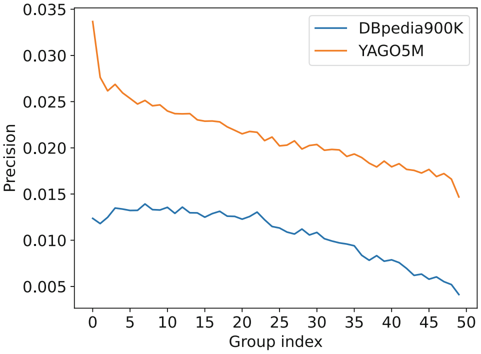


Fig. 2. Automatic evaluation of embedding score guidance. Y-axis is the precision score of each group, and X-axis shows the indices of the groups sorted by embedding scores, in which the larger is the group index the lower is the embedding score: embedding score guidance can help missing triplet prediction, but worse than query guidance.

Automatic Evaluation. We show the automatic evaluation results for KM in Table 3, where #Pairs and Pair Precision are the number of predicted entity-predicate pairs in this group (KG metadata compatible and incompatible) and their corresponding precision scores. We also show the automatic evaluation results for ES in Fig. 2, where x-axis is the group index, and larger group index indicates lower embedding score, which indicates lower quality (recall that we divide all predicted entity-predicate pairs into 50 bins based on their embedding scores); and y-axis is the precision score of this group.

We make three observations. First, the guidance of both KG metadata and embedding score can help prediction. This observation is supported by that 1) in Table 3, the precision scores of the compatible groups are >2 times higher than those of the incompatible groups; and 2) in Fig. 2, groups with higher embedding scores (i.e. smaller group indices) are of higher precision scores. Second, user queries still provide better guidance than both KG metedata and embedding scores, shown by that both 1) the precision scores of the compatible groups in Table 3 and 2) the group of the highest embedding score in Fig. 2 (i.e. leftmost) are outperformed by QG (i.e. Pair Precision in Table 1). Third, we observe that only a small portion of the predicted entity-predicate pairs are compatible with

KG metadata. Considering that KM works in a post-hoc way (i.e., it filters out incompatible ones after predictions are made), this result suggests the relatively low efficiency of KM compared with QG. The same concern applies to ES if we solely rely on the a few groups with the highest embedding scores.

Human Evaluation. We also conduct a human evaluation study to further compare the impact of these two types of guidance against query guidance. Consistent with Sect. 4.3, for each KG, we randomly select 200 entity-predicate pairs from both the compatible group in KM and the group of the highest embedding score in ES, and annotate their correctness and relevance.

Table 2 shows the results. We make similar observations as for the automatic evaluation: while the guidance through both KG metadata and embedding score achieve improvements over baseline, they are outperformed by QG.

5 Conclusions and Limitations

To improve the precision and relevance of KGC methods, we propose a user-driven approach based on explorative query logs. Our approach conceptually works for any type of query language where entities and properties can be defined. This includes explicit definition as RDF constructs in SPARQL, or implicitly through natural language queries "make-up for dark skin tone". The latter is particularly interesting for catching up with user-defined trends regarding product attributions. Because commercial KGs and queries are usually confidential, we perform our experiments with two popular general-purpose KGs, DBPedia and YAGO 4, and their SPARQL user queries. Specifically, we extract entity-predicate pairs from SELECT queries, and make predictions from KGE models from them, for they are likely to be correct and relevant to users. Our results from both automatic and human evaluations show that query guidance can significantly improve the correctness and relevance of predicted facts.

Our approach and its adaptation for open KGs opens up further avenues for the combined usage of KGs and query logs. In particular, future work can explore further aggregation and filtering of queries, and harvest more sophisticated structures from complex queries that suggest missing facts.

References

1. Arias, M., Fernández, J.D., Martínez-Prieto, M.A., de la Fuente, P.: An empirical study of real-world sparql queries. arXiv preprint arXiv:1103.5043 (2011)
2. Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: Advances in Neural Information Processing Systems, vol. 26. Curran Associates, Inc. (2013)

3. Brickley, D.: Rdf vocabulary description language 1.0: Rdf schema (2004). http://www.w3.org/TR/rdf-schema/
4. Chen, Z., Wang, Y., Zhao, B., Cheng, J., Zhao, X., Duan, Z.: Knowledge graph completion: a review. IEEE Access **8**, 192435–192456 (2020). https://doi.org/10.1109/ACCESS.2020.3030076
5. Dettmers, T., Minervini, P., Stenetorp, P., Riedel, S.: Convolutional 2d knowledge graph embeddings. In: Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence. AAAI 2018/IAAI 2018/EAAI 2018 (2018)
6. Dong, X.L., et al.: Autoknow: self-driving knowledge collection for products of thousands of types. In: KDD 2020 (2020). https://www.amazon.science/publications/autoknow-self-driving-knowledge-collection-for-products-of-thousands-of-types
7. Galárraga, L., Razniewski, S., Amarilli, A., Suchanek, F.M.: Predicting completeness in knowledge bases. In: Proceedings of the ACM International Conference on Web Search and Data Mining (WSDM), pp. 375–383 (2017). https://doi.org/10.1145/3018661.3018739
8. Galárraga, L.A., Teflioudi, C., Hose, K., Suchanek, F.: Amie: association rule mining under incomplete evidence in ontological knowledge bases. In: Proceedings of the International Conference on World Wide Web (WWW), pp. 413–422. Association for Computing Machinery, New York (2013). https://doi.org/10.1145/2488388.2488425
9. Jiang, X., Wang, Q., Wang, B.: Adaptive convolution for multi-relational learning. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, vol. 1 (Long and Short Papers), pp. 978–987. Association for Computational Linguistics, Minneapolis (2019). https://doi.org/10.18653/v1/N19-1103. https://aclanthology.org/N19-1103
10. Kazemi, S.M., Poole, D.: Simple embedding for link prediction in knowledge graphs. In: Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R. (eds.) Advances in Neural Information Processing Systems, vol. 31. Curran Associates, Inc. (2018)
11. Lehmann, J., et al.: Dbpedia-a large-scale, multilingual knowledge base extracted from wikipedia. Semant. web **6**(2), 167–195 (2015)
12. Lin, Y., Liu, Z., Sun, M., Liu, Y., Zhu, X.: Learning entity and relation embeddings for knowledge graph completion. In: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, pp. 2181–2187 (2015)
13. Lo, K., Wang, L.L., Neumann, M., Kinney, R., Weld, D.: S2ORC: the semantic scholar open research corpus. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 4969–4983. Association for Computational Linguistics, Online (2020). https://doi.org/10.18653/v1/2020.acl-main.447. https://www.aclweb.org/anthology/2020.acl-main.447
14. Meilicke, C., Chekol, M.W., Fink, M., Stuckenschmidt, H.: Reinforced anytime bottom up rule learning for knowledge graph completion. arXiv preprint arXiv:2004.04412 (2020)
15. Meilicke, C., Chekol, M.W., Ruffinelli, D., Stuckenschmidt, H.: Anytime bottom-up rule learning for knowledge graph completion. In: Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), IJCAI 2019, pp. 3137–3143 (2019)

16. Mnih, A., Teh, Y.W.: A fast and simple algorithm for training neural probabilistic language models. In: Proceedings of the 29th International Conference on Machine Learning, ICML 2012, Edinburgh, Scotland, UK, 26 June–1 July 2012. icml.cc/Omnipress (2012). http://icml.cc/2012/papers/855.pdf
17. Nickel, M., Tresp, V., Kriegel, H.P.: A three-way model for collective learning on multi-relational data. In: Proceedings of the International Conference on International Conference on Machine Learning (ICML), pp. 809–816. Omnipress, Madison (2011)
18. Pellissier Tanon, T., Weikum, G., Suchanek, F.: YAGO 4: a reason-able knowledge base. In: Harth, A., et al. (eds.) ESWC 2020. LNCS, vol. 12123, pp. 583–596. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-49461-2_34
19. Peng, C., Xia, F., Naseriparsa, M., Osborne, F.: Knowledge graphs: opportunities and challenges. Artif. Intell. Rev. 1–32 (2023)
20. Schlichtkrull, M., Kipf, T.N., Bloem, P., van den Berg, R., Titov, I., Welling, M.: Modeling relational data with graph convolutional networks. In: Gangemi, A., et al. (eds.) ESWC 2018. LNCS, vol. 10843, pp. 593–607. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-93417-4_38
21. Shang, C., Tang, Y., Huang, J., Bi, J., He, X., Zhou, B.: End-to-end structure-aware convolutional networks for knowledge base completion. In: Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence. AAAI 2019/IAAI 2019/EAAI 2019, AAAI Press (2019). https://doi.org/10.1609/aaai.v33i01.33013060
22. Sosa, D.N., Derry, A., Guo, M., Wei, E., Brinton, C., Altman, R.B.: A literature-based knowledge graph embedding method for identifying drug repurposing opportunities in rare diseases. In: Pacific Symposium on Biocomputing 2020, pp. 463–474. World Scientific (2019)
23. Sun, Z., Deng, Z.H., Nie, J.Y., Tang, J.: Rotate: Knowledge graph embedding by relational rotation in complex space. In: International Conference on Learning Representations (2019). https://openreview.net/forum?id=HkgEQnRqYQ
24. Sun, Z., et al.: Research commentary on recommendations with side information: a survey and research directions. Electron. Commer. Res. Appl. **37**, 100879 (2019). https://doi.org/10.1016/j.elerap.2019.100879. https://www.sciencedirect.com/science/article/pii/S1567422319300560
25. Trouillon, T., Welbl, J., Riedel, S., Gaussier, É., Bouchard, G.: Complex embeddings for simple link prediction. In: Balcan, M., Weinberger, K.Q. (eds.) Proceedings of the International Conference on Machine Learning (ICML), vol. 48, pp. 2071–2080. JMLR.org (2016). http://proceedings.mlr.press/v48/trouillon16.html
26. Wang, M., Qiu, L., Wang, X.: A survey on knowledge graph embeddings for link prediction. Symmetry **13**(3) (2021). https://doi.org/10.3390/sym13030485
27. Wang, Z., Zhang, J., Feng, J., Chen, Z.: Knowledge graph embedding by translating on hyperplanes. In: Proceedings of the AAAI Conference on Artificial Intelligence **28**(1) (2014). https://doi.org/10.1609/aaai.v28i1.8870. https://ojs.aaai.org/index.php/AAAI/article/view/8870
28. Wishart, D.S., et al.: Drugbank: a comprehensive resource for in silico drug discovery and exploration. Nucleic Acids Res. **34**(suppl_1), D668–D672 (2006)
29. Yang, Z., Ding, M., Zhou, C., Yang, H., Zhou, J., Tang, J.: Understanding negative sampling in graph representation learning. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2020, pp. 1666–1676 (2020). https://doi.org/10.1145/3394486.3403218

30. Zamini, M., Reza, H., Rabiei, M.: A review of knowledge graph completion. Information **13**(8), 396 (2022)
31. Zhou, S., et al.: Interactive recommender system via knowledge graph-enhanced reinforcement learning. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2020, pp. 179–188. Association for Computing Machinery, New York (2020). https://doi.org/10.1145/3397271.3401174

PeersimGym: An Environment for Solving the Task Offloading Problem with Reinforcement Learning

Frederico Metelo[1](✉), Cláudia Soares[1], Stevo Racković[2], and Pedro Ákos Costa[1]

[1] NOVA School of Science and Technology, Caparica, Portugal
{fc.metelo,pah.costa}@campus.fct.unl.pt, claudia.soares@fct.unl.pt
[2] Instituto Superior Técnico, Lisbon, Portugal
stevo.rackovic@tecnico.ulisboa.pt

Abstract. Task offloading, crucial for balancing computational loads across devices in networks such as the Internet of Things, poses significant optimization challenges, including minimizing latency and energy usage under strict communication and storage constraints. While traditional optimization falls short in scalability; and heuristic approaches lack in achieving optimal outcomes, Reinforcement Learning (RL) offers a promising avenue by enabling the learning of optimal offloading strategies through iterative interactions. However, the efficacy of RL hinges on access to rich datasets and custom-tailored, realistic training environments. To address this, we introduce PeersimGym, an open-source, customizable simulation environment tailored for developing and optimizing task offloading strategies within computational networks. PeersimGym supports a wide range of network topologies and computational constraints and integrates a *PettingZoo*-based interface for RL agent deployment in both solo and multi-agent setups. Furthermore, we demonstrate the utility of the environment through experiments with Deep Reinforcement Learning agents, showcasing the potential of RL-based approaches to significantly enhance offloading strategies in distributed computing settings. PeersimGym thus bridges the gap between theoretical RL models and their practical applications, paving the way for advancements in efficient task offloading methodologies.

Keywords: Task Offloading · Load Balancing · Peer-to-Peer Communication · Environment Simulator · Reinforcement Learning

1 Introduction

The proliferation of large networks of devices, such as the Internet of Things (IoT), has led to an exponential increase in data generation, requiring significant computational resources for processing. Traditionally, Cloud Computing

---

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-70378-2_3.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
A. Bifet et al. (Eds.): ECML PKDD 2024, LNAI 14949, pp. 38–54, 2024.
https://doi.org/10.1007/978-3-031-70378-2_3

provided the backbone for such computational demands. However, its limitations in latency and network traffic have become apparent with the growth of device networks [12]. Edge Computing emerged as a paradigm shift, extending the Cloud to bring processing capabilities closer to data sources, mitigating latency and traffic issues. In this evolving landscape, task offloading—the distribution of computational tasks across network participants—has gained prominence, particularly within the Fog and Multi-access Edge Computing (MEC) paradigms. Both paradigms aim to decentralize computing power, bringing it closer to end-users and alleviating the constraints of traditional Cloud Computing [15,28]. Despite distinctions between MEC and Fog computing, this paper treats them interchangeably, focusing on their shared goal of minimizing device-to-cloud distances [29]. Addressing the challenges of task offloading in such distributed environments involves balancing numerous factors, including task latency, energy consumption, and task completion reliability [31]. Conventional optimization methods often struggle to efficiently manage these complex systems. Multi-Agent Reinforcement Learning (MARL) offers promise in optimizing resource allocation and scheduling to maximize system efficiency or meet specific performance metrics [9,30]. In the task offloading problem, a reinforcement learning (RL) agent can serve as a decision maker to learn the optimal task-resource allocation strategy by interacting with the environment (i.e., the task and available resources). However, it is impractical to train agents in the real world. Therefore, a standardized, customizable, efficient, and user-friendly simulation tool is essential.

Contributions. Our contributions focus on advancing the field of MARL and Edge computing through the introduction of the PeersimGym environment and comprehensive experimental analysis. The **PeersimGym Environment** is a highly adaptable simulation platform tailored for the development, training, and evaluation of MARL strategies for task offloading challenges in Edge Computing systems. This environment, built on the synergy between an edge system simulator built with Peersim P2P simulator [14] and the PettingZoo API [23], allows detailed configuration of network topologies, node characteristics, and task parameters, facilitating a wide range of experimental setups. An **experimental analysis** demonstrates the capability of the PeersimGym to train effective MARL solutions, exemplified by the performance of a Double Deep Q Network and Advantage Actor Critic algorithm. Our analysis compares the performance of DRL agents to several non-RL algorithms across various network configurations and task offloading scenarios, highlighting the advantages of RL approaches in optimizing Edge Computing networks. The **source code** for PeersimGym, along with usage documentation and testing resources, is made available in the Simulator repository[1], and the Agent repository[2], fostering further research and community engagement in advancing Edge Computing solutions.

---

[1] https://github.com/FredericoMetelo/peersim-environment.

[2] https://github.com/FredericoMetelo/TaskOffloadingAgentLibrary.

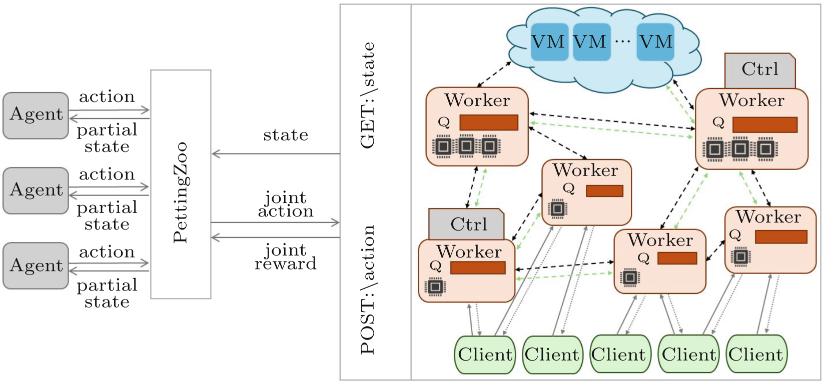


Fig. 1. Simulation Pipeline Overview. Left: PettingZoo API integration, facilitating agent-simulation interaction via Python and RESTful requests for practical task offloading optimization. Right: network topology with worker nodes and connections, enabling task generation and state sharing for RL agent training.

2 Background and Related Work

Task offloading addresses the redistribution of computationally intensive tasks from resource-limited devices to more capable ones to enhance system performance. This process requires strategic decision-making regarding the *what*, *where*, *how*, and *when* aspects of offloading tasks. Literature on task offloading distinguishes vertical offloading to higher-tier systems [19], horizontal offloading among peers [2,31], and hybrid approaches that blend both directions [3]. The choice of offloading destinations varies significantly, considering idle nodes, those with shorter task queues [2], proximity constraints [27,29], or unrestricted selection accounting for potential offloading failure consequences [3]. Failures in offloading can arise from latency constraints [4], node capacity limitations [27], energy shortages, or other [18]. Task modeling further diversifies the field, encompassing homogeneous versus heterogeneous tasks [10], divisible versus indivisible tasks [12], and dependent versus independent tasks [6]. Common objectives for task offloading include minimizing computation and transmission latency [29], reducing energy consumption [7], and avoiding task failures [2,12]. Other considerations such as task utility [8], queue wait times, and CPU utilization [29] are less frequent. Our simulator, PeersimGym, is designed to accommodate this broad spectrum of offloading strategies, task models, and objectives. It offers the flexibility to configure various aspects of the simulation environment, enabling the exploration of a wide range of scenarios and contributing to a more nuanced understanding of task offloading dynamics in distributed computing networks.

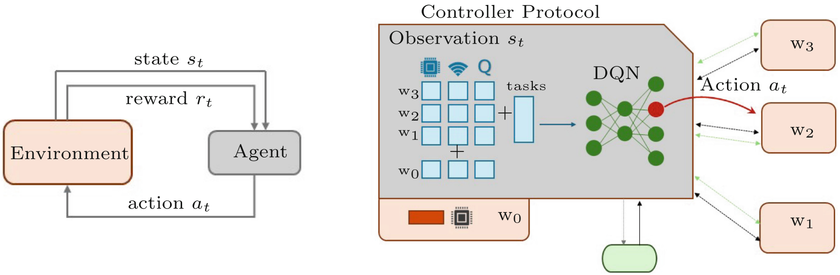
Reinforcement Learning (RL) for Task Offloading. RL is a powerful and dominant approach for solving the task offloading problem, as it can find an optimal solution with excellent efficiency, given a well-defined environment and correct reward shaping. RL has been applied to various Fog and MEC settings, considering single agent methods [27], as well as multi-agent with a set of independent learners [4] or in federated cooperation [25]. Each category assumes different observability and sharing among nodes. Models can be fully or partially observable [3], with local or global optimization objectives in a multi-agent case. Learning agents range from tabular methods [2], and multi-armed bandits [31], to complex deep Q networks [25] or actor-critic agents [3].

Our contribution addresses a gap in the existing literature by introducing a training environment for agents, facilitating uniform comparisons across different solutions. Given the diverse nature of these solutions, our simulation tool offers a high level of customizability to accommodate the diverse requirements. Specifically, we provide a simulation platform tailored for training both centralized and decentralized reinforcement learning algorithms, targeting task offloading in edge systems. The proposed simulator includes a PettingZoo [23] environment to interface the simulation as an integral component that allows a user to design and train an arbitrary RL agent(s) over a selected environment.

Comparative Advantages of PeersimGym Over Existing Simulators and Environments. Available simulators for edge-like networks [11,22] are not prepared out of the box for RL training and do not provide a high level of flexibility for different protocols and topologies. There are also environments for RL that allow for training task offloading RL agents [5,21]; however, the solution of [5] requires implementing the routing mechanism for multi-agent reinforcement learning, and of [21] is built on the engine from [5], and focuses only on a single task vertical offloading scenario. PeersimGym addresses these limitations by enabling the user to configure multiple task offloading scenarios, namely that of horizontal task offloading in the P2P setting for Load Balancing. Furthermore, PeersimGym uses PettingZoo, a version of OpenAI gym [23] focused on MARL, and it provides an API better adapted for the task [23] than OpenAI gym and its successor Gymnasium [26]. To the best of our knowledge, PeersimGym is the only environment for RL developed with MARL task offloading and high configurability and modality as its central focus.

3 Enhancing Task Offloading with MARL

To address the complexity of Task Offloading in highly complex edge environments, researchers leverage RL and its subset, DRL, and by exploring the distributed nature of these systems MARL emerges as promising solution. As a tool to develop MARL algorithms, PeersimGym incorporates a Python-based environment tailored for developing, training, and deploying RL models for task offloading, aligned with the *PettingZoo* framework. With the set of experiments in this paper, we showcase how our simulator can enable further RL and DRL contributions to this field in Sect. 5. Certain Nodes within the network act as RL agents. These agents have the capability to observe the state (albeit partially) and make informed decisions regarding whether to process tasks locally or offload



(a) Reinforcement Learning Loop.

(b) Local state and action for a worker node.

Fig. 2. General and problem-specific RL state action overview.

them. This decision-making process is influenced by the need to balance between local processing, the risk of queue overflow, and the costs associated with task offloading, including potential overload of other nodes.

The Role of Reinforcement Learning Agents. The interaction cycle of an RL agent with its environment is structured around a continuous loop where, at each timestep $t = 1, ..., T$, the agent observes the system state s_t, executes an action a_t based on this observation, and receives feedback in the form of a reward r_t. This feedback reflects the effectiveness of the action, taking into account both its immediate impact and its influence on future states. Through this iterative process, the agent refines its policy—a set of rules determining its actions in various states—to maximize cumulative rewards, thereby aligning with the goal of optimizing task offloading decisions (Fig. 2).

Deep Reinforcement Learning for Task Offloading. Deep Q Networks (DQN) [27] and their variants, such as Double Deep Q Networks (DDQN) [13] and Actor-Critic methods, like the Advantage Actor-Critic (A2C) [3], are prominent DRL approaches applied to task offloading. These methodologies have been shown to stabilize training and enhance learning efficacy through sophisticated neural network architectures that approximate optimal action-selection policies. In PeersimGym, the flexibility of our RL environment supports the integration of various DRL models. For illustrative purposes, we focus on implementations of DDQN and A2C, reflecting their proven effectiveness in recent literature [3,4]. This choice underscores the potential of DRL to address the complexities associated with task offloading in Edge Computing, as shown in Sect. 5.

Framework and Model Dynamics. Task offloading decisions within our simulated environment are modeled as a Markov Game (MG), accommodating the multi-agent aspect of Edge Computing networks. This formulation extends the Markov Decision Process (MDP) framework to scenarios involving multiple decision-makers, thereby capturing the interactive and competitive nature of task offloading. Single-agent RL is modeled as an MDP, with a sequence of states such that the Markov Property holds, i.e., the next state s_{t+1} depends exclusively on the current state s_t and the performed action a_t. When including multiple agents, most MDP convergence properties do not hold, hence, we formulate our problem as an MG [17]. In our setting, in a network of N nodes, an MG is represented as a tuple $\langle n, \mathcal{S}, \mathcal{A}, P, R, \gamma \rangle$, where n is the number of agents (nodes with a controller protocol), $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, P is an (unknown) transition probability function, R is the reward function, and γ is a reward discount factor. Next, we introduce these constructs in more detail.

State Space. The state space in our environment is designed to be highly customizable, enabling the modeling of various Edge Computing scenarios. At time step t, each node will broadcast its local state to each of its neighbors and receive their local states to build a local state representation before deciding on the action. The state space is represented by a tuple $\mathcal{S} = (\mathcal{I}, \mathcal{K}, \mathcal{Q}^t, \mathcal{F}, \mathcal{L}, \{\mathcal{B}_1, ..., \mathcal{B}_n\}, \{\mathcal{P}_1, ..., \mathcal{P}_n\})$, with elements

- $\mathcal{I} = \{1, ..., n\}$, array with the IDs of all the nodes in the network;
- $\mathcal{K} = \{\kappa_1, ..., \kappa_n\}$, layer/tier for each node;
- $\mathcal{Q}^t = \{Q_1^t, ..., Q_n^t\}$, queue size for each node in the network, at time step t;
- $\mathcal{Q}_i^{\max} = \{Q_1^{\max}, ..., Q_n^{\max}\}$, maximum capacity for each node's queue;
- $\mathcal{N}_n$, IDs of the nodes in node n's neighborhood;
- $\mathcal{F} = \{N_\phi^1 \phi_1, ..., N_\phi^n \phi_n\}$, processing power for each node;
- $\mathcal{L} = \{l_1, ..., l_n\}$, position for each node in the network;
- $\mathcal{B}_i = \{B_{i,1}, ..., B_{i,n}\}$, channels' bandwidth for node w_i to all its neighbors;
- P_i, the transmission power of node w_i's antenna.

The size of the state vector will define the input size of a DRL agent (Fig. 2 (b)); hence, in this version of the simulator, the state dimension needs to stay consistent throughout iterations.

Action Space. The action space is similarly designed to reflect the decision-making process regarding task offloading, with actions representing the choice of offloading destinations. This setup facilitates the exploration of strategies that balance local processing advantages against the costs and implications of offloading. The action space, $\mathcal{A}$, corresponds to the output layer of a DRL agent (see Fig. 2). The action $a_t \in \{1, ..., N\}$ represents the index of the destination node, which might be one of the neighboring workers or the observed node itself, in case it decides to process the task locally. We use a_t interchangeably to denote both the index of the destination node and the act of sending a task to that node, as long as clarity is maintained within the context.

Reward Function. The reward function is a critical component guiding the learning process of the RL agent. It is constructed to reward actions that enhance utility—such as task completions—while penalizing undesirable outcomes like excessive delays or system overloads. This balance encourages the development of nuanced offloading policies that consider various operational constraints and performance metrics. By incorporating reward shaping [16], we further refine the learning process, enabling agents to navigate the complex decision space of task offloading more effectively. This approach not only facilitates faster convergence to optimal policies but also allows for a more nuanced understanding of the trade-offs inherent in Edge Computing task management. The reward shaping term can be defined as $F(s_t, a_t, s_{t+1}) = \Xi(s_{t+1}) - \Xi(s_t)$, where $\Xi(s_t)$ is a user-defined potential function over a state, s_t such that the reward becomes $R' = R + F$.

In this paper, we adopt a reward function as defined in [2], i.e., a reward function for agent w_i, R_i, is structured to maximize the utility, $U_i(s_t, a_t)$, and minimize the total delay, $D_i(s_t, a_t)$, and the overloading cost, $O_i(s_t, a_t)$. In particular, the reward for the action a_t in state s_t, received by an agent w_i by offloading (or not) task τ_k, is given by

$$R_i(s_t, a_t) = r_u - (D_i(s_t, a_t) + \chi_O O_i(s_t, a_t)), \tag{1}$$

where r_u is a utility reward and represents the gain over completed tasks, and χ_O is overloading cost weight. Each term of the reward function is explained in detail below. First we introduce an indicator function, $I_i(a_t)$, such that $I_i(a_t) = 1$ *iff* $a_t = i$, meaning the task is meant to be processed in node w_i, otherwise $I_i(a_t) = 0$. The reward function for an agent in the context of task offloading is composed of two primary components: the delay function and the cost of overloading. Each incorporates specific parameters and equations that encapsulate the complexities of decision-making in edge-computing environments. The delay function denoted as $D_i(s_t, a_t)$, is a comprehensive measure that accounts for three distinct temporal aspects associated with task offloading, namely

$$D_i(s_t, a_t) = \chi_D^{\text{wait}} T_{i,a_t}^{\text{wait}}(\tau_k) + \chi_D^{\text{comm}} T_{i,a_t}^{\text{comm}}(\alpha_k^{\text{out}}) + \chi_D^{\text{exc}} T_{i,a_t}^{\text{exc}}(\tau_k), \tag{2}$$

Here, χ_D^{wait}, χ_D^{comm}, and χ_D^{exc} serve as hyperparameters, adjusting the weight of each time-related component within the overall delay function. The waiting time $T_{i,a_t}^{\text{wait}}(\tau_k)$ reflects the duration a task τ_k spends in the queue, either at the originating node w_i or at an offloading target w_j, and is given by

$$T_{i,a_t}^{\text{wait}}(\tau_k) = \frac{Q_i^t}{N_\phi^i \phi_i} + \sum_{j \neq i} \frac{Q_j}{N_\phi^j \phi_j} I_j(a_t), \tag{3}$$

where Q_i^t is the queue size at time t, N_ϕ^i and ϕ_i represent the number of processors and their frequency at node w_i, respectively. To quantify the transmission efficiency, specifically the rate at which bits are communicated per second, we invoke the *Shannon-Hartley* theorem [1]. Accordingly, the communication latency for transmitting α_k^{out} bits between nodes w_i and w_{a_t} is given by

$$T_{i,a_t}^{\text{comm}}(\alpha_k^{\text{out}}) = \frac{\alpha_k^{\text{out}}}{B_{i,a_t} \log(1 + 10^{\frac{P_i + G_{i,a_t} - \omega_0}{10}})}, \tag{4}$$

where B_{i,a_t} is the bandwidth of the communication channel between nodes w_i and w_{a_t}, P_i denotes the transmission power of the source node w_i, G_{i,a_t} is the channel gain, and ω_0 represents the noise power in the communication channel. This formulation underpins our model for evaluating the communication overhead associated with task offloading in Edge networks. P_i, G_{i,a_t} and ω_0 are measured in dB. The execution cost difference $T^{\text{exc}}_{i,a_t}(\tau_k)$ between local and target node processing is quantified as

$$T^{\text{exc}}_{i,a_t}(\tau_k) = \frac{\rho_k \xi_k}{N^{a_t}_{\phi} \phi_{a_t}} - \frac{\rho_k \xi_k}{N^{i}_{\phi} \phi_i}, \tag{5}$$

indicating the variation in processing time due to differences in node capabilities. The cost of overloading, $O_i(s_t, a_t)$, emphasizes the potential system strain caused by task offloading, expressed as $O_i(s_t, a_t) = -\log(p^{a_t}_t)/3$. The probability of overloading $p^{a_t}_t$ and the expected queue state Q'_{a_t} are critical in assessing the impact of offloading decisions on the target node's workload, calculated as $p^{a_t}_t = \max\left(0, \frac{Q^{\max}_{a_t} - Q_{a_t}}{Q^{\max}_{a_t}}\right)$, and $Q'_{a_t} = \min(\max(0, Q_{a_t} - \phi_{a_t}) + 1, Q^{\max}_{a_t})$, where $Q^{\max}_{a_t}$ is the maximum queue capacity of the offloading target, and ϕ_{a_t} is its processing rate. These equations collectively frame the decision-making landscape for RL agents, highlighting the intricate balance between task processing efficiency, communication overheads, and system resilience against overloading.

4 PeersimGym

PeersimGym introduces a novel framework for simulating and training task-offloading MARL algorithms in Edge networks. We leverage the versatile Java-based PeerSim P2P simulator [14] and extend it to model edge systems. Additionally, PeersimGym incorporates a Python API compatible with the PettingZoo framework [26], offering an intuitive structure for developing RL for task offloading. This section delineates the two primary components of our tool: a simulator for custom system network creation and a Python environment for constructing and training RL models, grounded in PettingZoo principles (Fig. 1).

4.1 System Modeling

PeersimGym allows a high level of customization in crafting edge systems, enabling customization of components and their attributes. We introduce models for nodes, tasks, and communication, which collectively define the communication dynamics of the simulation.

Node Model. The simulation framework models a network comprising client devices $\mathcal{C}$, akin to IoT sensors, which generate and dispatch data for processing. This data is handled by worker nodes, which can either process tasks locally or offload them to other nodes with available resources. Worker nodes $\mathcal{W}$ possess distinct characteristics, including:

- **Task queue**, Q_i, a data structure that allows at most $Q_i^{\max}$ received tasks to be stored and await to be processed in a first-in-first-out fashion. Any tasks received above the capacity of the node will be dropped.
- **Number of processors**, N_ϕ^i, of frequency, ϕ_i. The node can process $N_\phi^i \phi_i$ instructions per time step.
- **Transmission power**, P_i, that affects the wireless communication delays.
- **Location of the node**, l_i, which also affects the communication delay and other proximity-based mechanisms.

Worker nodes $\mathcal{W}$ are categorized into tiers, reflecting a hierarchy similar to the fog computing model. This tiered structure, alongside the optional integration of Cloud servers, facilitates the modeling of various network architectures from P2P to hierarchical n-tier systems (Fig. 1). Nodes containing an RL agent can offload tasks to their neighbors, managing offloading decisions and maintaining state information of adjacent nodes.

Task Model. The workload originates from processing tasks τ_i, denoted as tuples representing computational demands. These tasks, generated by clients, include attributes such as instruction count, input/output data sizes, CPU cycle cost per instruction, and processing deadlines. Specifically, a task is represented as $\tau_i = \langle i, \rho_i, \alpha_i^{\text{in}}, \alpha_i^{\text{out}}, \xi_i, \delta_i \rangle$, where i is a unique identifier of the task; ρ_i is the number of instructions to be processed; α_i^{in} is the total data size of the input; α_i^{out} is the data size of the output/results; ξ_i is the cost in CPU cycles per instruction; δ_i is the deadline of the task, i.e., the maximum allowed latency for returning the results. Task arrival follows a Poisson distribution, and if capacity is exceeded, tasks are dropped (Fig. 1).

Communication Model. Task offloading and reception are simulated under the assumption of a generic wireless communication model, as this is the most common in the literature. Utilizing the Shannon-Hartley theorem [1], we calculate the bits transmitted per second, incorporating factors such as channel bandwidth, channel gain, and transmission power (see Eq. 4).

In the context of task generation and reception, nodes equipped with a *controller protocol* play a pivotal role in determining whether to process tasks locally or offload them. This decision-making process is facilitated through the exchange of local state information via one-hop broadcasts. Communication within the network, particularly the offloading interactions among nodes, is predicated on wireless transmission.

4.2 Implementation Details

PeersimEnv. PeersimGym integrates the Peersim simulator and the PettingZoo environment (PZ env), facilitating the configuration and execution of simulations through REST requests. This integration is enabled by encapsulating the Peersim simulation within a Spring Boot REST server, providing endpoints for action posting and state retrieval, adhering to PettingZoo standards (Fig. 1). A full list of the available configurations can be found in the code repository.

Simulator. At its core, PeersimGym employs the Peersim simulation tool, supporting both event-driven and cycle-driven engines. The simulation models a network as a collection of nodes running various protocols, including client, worker, controller, and a customized Simulation Manager protocol. This Simulation Manager periodically pauses the simulation to process actions and resume, maintaining a cyclical operation that aligns with the specified behaviors of worker, client, and controller protocols. Through extensive configurability and a focus on customization, PeersimGym presents a powerful tool for simulating Edge network environments and training RL models for task offloading, allowing for more sophisticated and realistic simulations in Edge Computing research.

Enhanced Simulation Protocols and Event Handling. In the simulation environment of our system, each node adheres to specific protocols designed to mimic the behavior of Edge Computing networks accurately. These protocols, namely the worker, client, and controller protocols, are essential for the dynamic interaction between nodes, ensuring the efficient processing and distribution of tasks. Below, we outline the refined functionalities of these protocols.

- **Worker protocol:** Central to our simulation, the worker protocol governs the processing mechanics of tasks within the network. Upon receiving a task, the worker updates a task-specific instruction counter to track its progress. Completion of a task triggers the protocol to return the result to the originating client. This return path may involve multiple hops, particularly for tasks that have been offloaded across several nodes. Should the queue of the worker deplete, it transitions into an idle state, awaiting new tasks.
- **Client protocol:** The client protocol is responsible for task distribution within the network. It employs a Poisson process to determine the allocation of tasks to neighbors, with eligibility for task receipt specified in the system configuration. This probabilistic approach ensures a realistic simulation of task dissemination behavior observed in Edge Computing scenarios.
- **Controller protocol:** This protocol is coupled with a worker protocol, and monitors the worker's state. Upon detecting a state alteration, the controller initiates a one-hop broadcast to disseminate the updated state information to neighboring nodes. Maintaining an updated network state and facilitating informed decision-making for task offloading. It also acts as the bridge between the simulation and the RL agents, passing the offload instructions to the worker, virtually representing the agents, we shall refer to the offload decisions as if they are made by the Controller.

Event Handling Mechanisms. To enhance the fidelity of our simulation, we implement a realistic event-handling system. This system allows each protocol to respond to specific events that reflect real-world interactions within an Edge Computing environment. These events include the following:

- **Worker Protocol Events:** Engages in routines for handling offloaded tasks from other nodes, new tasks directly from clients, and results of concluded tasks. If space permits, tasks are added to the queue for processing; otherwise,

they are dropped. Completed tasks are either directed back to the origin client or offloaded to closer workers for final delivery.
- **Client Protocol Event:** Manages the completion of tasks, focusing on registering relevant metrics upon the conclusion of tasks sent to workers.
- **Controller Protocol Event:** Handles the Neighbour State update event, ensuring the node updates its information regarding the state of neighboring nodes as necessary.

In the case of the controller protocol, the offloading event is different from the other events. The instructions are received through a REST request from the simulator and are passed by a special class evoking a method directly on each of the controllers which will take the actions immediately

Therefore, by implementing Java classes extending the *AbstractWorker*, *AbstractClient*, and *AbstractController* provided, the user can change the behavior of the elements in the simulation. Furthermore, each of the simulations can be configured based on configuration files. We allow an in-depth configuration of most aspects of the simulation, which include the network topology, the manipulation of what nodes support the controller functions, most properties of the tasks, and the configuration of all protocols. We have also developed a configuration helper tool to simplify the creation of the configuration files. The documentation for all the possible configurations can be found in the environment repository.

The main focus of the introduced simulator is on being highly customizable, which includes implementing new protocols that make use of the highly innate modality of the Peersim Tool and the extensions we put in place to manage the simulation of Edge Networks. To tailor the simulation to the required scenario, it is possible to define different protocols for the clients, controllers, and workers, that extend their abstract implementations. Furthermore, different actions and information available to each node can be defined, and the communication model and neighborhood definition can also be customized. We provide multiple base classes that allow doing out-of-the-box, binary task offloading, where clients generate indivisible tasks and controllers make decisions on where to offload the full tasks; or batch binary task offloading, where clients generate indivisible tasks but controllers decide for each of the tasks arrived in the time after the last offloading decision where they should be processed. More information on what each one does, how to load different modes, and how to create other implementations can be found on the simulator repository. In this paper, we focus on the binary task offloading implementations.

Simulation Data. PeersimGym provides a set of resources for collecting data from the simulation that help with the development process of agents and to provide insights on the behaviors of different agents. We provide log features and a helper class for collecting different metrics from the simulation. Furthermore, we implemented a straightforward visual rendering using Pygame to provide an easy-to-view and understand human-readable execution.

4.3 Reducing the Reality Gap

The optimal training and evaluation of an RL agent for task offloading hinge on utilizing simulation datasets derived from real-world edge systems. Yet, the complexity and diversity inherent to such systems, compounded by the scarcity of standardized reference architectures, benchmarks, and deployment data, present formidable challenges to conducting realistic evaluations of algorithms within actual edge environments [20]. This section delineates our approach to mitigating these challenges by leveraging tools that integrate with PeersimGym to generate plausible topologies [20] and workloads [24], thereby simulating environments that more closely mirror reality.

Trace-Generator tool. We employ the trace-generation tool to synthesize workloads based on real-world cluster traces from Alibaba Cloud [24], producing datasets that mirror actual computational demands. This tool generates workloads comprising multiple jobs, each depicted as a Directed Acyclic Graph (DAG) of tasks, where each task may include several instances requiring specific memory and CPU resources. While the ubiquity of DAGs in real-world applications is undeniable, the current iteration of our simulator does not support them. Consequently, we interpret the jobs within the trace-generator dataset as necessitating the peak CPU and memory resources identified across all tasks within a job. We calculate task instructions by factoring in the CPU frequency, the requested cores, and task duration. The maximum memory usage across all tasks within a job is considered equivalent to the data size in our simulation.

To incorporate the trace-generated data into PeersimGym, we develop a Python script (*Utils/DatasetGen.py*) processing the dataset and outputs a *JSON* file. This file is then utilized by the AlibabaTraceClient, a client implementation within our simulator, which samples tasks based on the synthesized dataset.

Topology Generator. The *Ether* tool [20] enables the generation of realistic infrastructure configurations, drawing from various Edge Computing scenarios. Our focus is on the Urban Sensing Scenario, inspired by the Array of Things project, which emphasizes data collection in smart cities. This scenario features clusters equipped with sensor nodes, each powered by Single Board Computers (SBCs) and connected to base stations comprising servers and GPU-equipped machines. These nodes process tasks locally or offload them to more capable nodes, simulating a realistic Edge Computing environment.

We augmented the Ether project to facilitate the generation of a topology and the assignment of coordinates to nodes. This setup ensures communication compatibility among nodes within the generated network, allowing for seamless integration with PeersimGym through a helper script that imports the topology data, thereby enhancing the realism of the simulator and applicability to real-world Edge Computing scenarios.

5 Experimental Results and Analysis

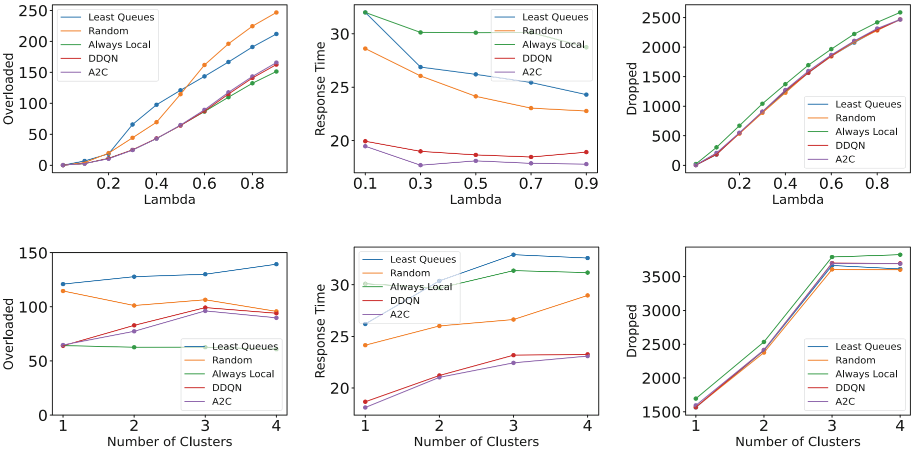
In this section, we evaluate PeersimGym by addressing two pivotal research questions (RQs) that underscore the adaptability and scalability of the MARL solution under varying network conditions:

RQ1 How does the MARL solution adapt its behavior to a fixed network topology and varying task arrival rates?

RQ2 How does the MARL solution adapt its behavior to a fixed task arrival rate with an increasing number of nodes and agents?

5.1 Experimental Setup

The experiments are grounded in four distinct network topologies generated using Ether, each featuring an incremental number of Array of Things (AoT) clusters. These scenarios leverage the realistic topology and workload generation methodologies detailed in Sect. 4.3. Each AoT cluster, predominantly composed of Single Board Computers (SBCs), includes pairs of SBCs and a base station equipped with an Intel NUC and two GPU units, alongside a remote, more potent server. The simulation parameters for all scenarios can be found in the Agent Repository. The topologies vary in cluster numbers, spanning one to four clusters, and correspondingly in node counts, from 12 to 31. The controller protocol is present in varying quantities (8 to 22 nodes across different topologies). Despite uniform CPU frequencies, the computational power varies with the core count, with the remote server having the highest capacity. The communication



(a) Overloaded nodes (b) Response time (c) # of dropped tasks

Fig. 3. Evolution of the different metrics with variable λ (top) and variable cluster number (bottom), averaging 100 episodes.

Table 1. Parameter values in the experimental setup.

Simulation time, T	1000 s	Nodes per tier	(10, 10, 10)
Task arrival rate, λ	0.17	Processor frequencies ϕ_i	$(4, 2, 8)e^7$ MHz
Task input size, α_i^{in}	150 Mbytes	Numbers of cores N_ϕ^i	(1, 1, 2)
Task output size, α_i^{out}	150 Mbytes	Buffer capacities, $Q_i^{\max}$	(20, 10, 100)
Task instructions, ρ_i	8e7	Task utility, r_u	2
CPI, ξ_i	1	Weight waiting, χ_D^{wait}	20
Deadline, δ_i	100 s	Weight excecution, χ_D^{exc}	20
Bandwidth, $B_{i,j}$	2 MHz	Weight comm, χ_D^{comm}	20
Transmission power, P_i	20 dbm	Weight overload, χ_O	150

capabilities and bandwidths are homogeneous across all nodes. Task generation at each SBC node follows a *Poisson*(λ) distribution, over a simulation episode of 1000 time steps. Each agent makes an offloading decision in each time step, through 300 training and 100 inference episodes. See Table 1 for exact values.[3]

To answer the RQs, we test the performance of the DDQN and A2C agents and a set of baseline approaches: 1.) *Local Processing* – never offload tasks. 2.) *Random Offloading* – select the target node randomly. 3.) *Least Queue* – select the neighbor with the shortest task queue at the time as a target.

Metrics. To show that our simulation can be used effectively to train RL agents in a MARL fashion, we select three commonly used metrics in the field to test the agent: 1.) the number of times a node is overloaded, 2.) the average task completion time for the tasks, and 3.) the number of dropped tasks.

5.2 Results Analysis

RQ1: We examine the behaviors of different agents when faced with different workloads by varying the parameter λ that governs the task arrival rate of each SBC node. In Fig. 3 (top), we observe that, with increasing λ, the nodes run out of available resources quickly. The almost linear increase in dropped tasks and overloaded nodes could indicate that the nodes are already working at capacity even for $\lambda \approx 0.1$. The reduction in response time observed in the *Least Queue* and *Random* approaches could be explained by more tasks being processed locally, due to the influx of tasks arriving surpassing the number of tasks offloaded keeping the node always full. In terms of task dropping and node overloading, the *DDQN* constantly achieves as reasonable rates as the best baselines. At the same time, it gives significantly better response time, with a large margin of difference, and apparently not affected by the task arrival rate. Behavior of *A2C* follows closely that of *DDQN*.

---

[3] We provide the in-depth configurations for the environment in the agent repository.

RQ2: Effects of increasing the number of nodes are shown in Fig. 3 (bottom). We test four network topologies of the AoT scenario and observe that the number of dropped tasks increases almost linearly with the increase in the number of nodes – note that the sharp break at the end is due to the 3-cluster topology having nearly the same number of nodes as the 4-cluster topology. This can indicate that the network is exhausting its resources for all of the topologies. The number of dropped tasks increases because the number of nodes, and therefore the total amount of tasks generated, increases. The response times also observe a slight increase due to the more powerful server that is shared across all the clusters filling up, and therefore, the tasks spend more time waiting in its queue, reducing the concurrency that was available when fewer nodes could access the server. The number of overloaded nodes also increases for the *Least Queue*, *A2C*, and *DDQN*, since the shared server is overloading more often. The number of overloads for the *Local Processing* remains stable and is lower than the other approaches because it never incurs the cost of overloading the base station or the server. Still, in turn, the number of dropped tasks is also higher. RL solutions again show clear superiority in terms of response times compared to baselines, while not sacrificing other metrics.

6 Conclusion and Future Work

We introduced PeersimGym, a highly customizable environment for the development and evaluation of MARL-based solutions to the task offloading problem in Edge Computing systems. PeersimGym integrates a MARL environment, compatible with the PettingZoo framework, facilitating agent interaction within a Python-based setting, and a simulation platform constructed atop Peersim, a Java tool for simulating Peer-to-Peer (P2P) networks. Our framework encompasses a suite of protocols for controllers, workers, and networking layers, offering the flexibility to modify system behaviors through configuration changes. The demonstrated efficacy of PeersimGym across various network settings highlights its potential as a training ground for MARL strategies in edge environments. To foster further research and collaboration within the community, we have made the source code for both the simulation environment and the agent development toolkit publicly available. Our documentation, including comprehensive wikis, provides detailed guidance on utilizing PeersimGym, encouraging researchers to explore and benchmark their own RL algorithms using our tool. This initiative aims to accelerate advancements in the domain and contribute to the broader efforts of the research community in optimizing Edge Computing systems.

Future Enhancements. Future enhancements for PeersimGym will focus on dynamic Edge Computing features: (1) Node and System Mobility, adding mobile node and service models with replicas for uninterrupted service; (2) Task Diversity, introducing a wider range of tasks tailored to specific hardware capabilities; (3) Data Locality Sensitivity, implementing prioritization based on data proximity to streamline offloading; and (4) Federated Reinforcement Learning, adopting collaborative training methods across nodes to optimize learning and parameter sharing.

Broader Impact. The versatility of PeersimGym spans beyond Edge Computing into fields like smart grids and satellite communications, where it can optimize energy management and enhance connectivity, respectively. Its adaptability to various P2P scenarios demonstrates the potential for broad technological impacts, though some domain-specific modifications may be required.

Acknowledgments. This work was partially funded by FCT IP, through NOVA LINCS (UIDB/04516/2020), and Project "Artificial Intelligence Fights Space Debris" No C626449889-0046305 co-funded by Recovery and Resilience Plan and NextGeneration EU Funds, www.recuperarportugal.gov.pt. And, by the European Union (TARDIS, 101093006). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.

References

1. Anttalainen, T.: Introduction to Telecommunications Network Engineering, 2nd edn. Artech House Telecommunications Library. Artech House, Boston (2003)
2. Baek, J., et al.: Managing fog networks using reinforcement learning based load balancing algorithm. In: 2019 IEEE WCNC, pp. 1–7 (2019)
3. Baek, J., Kaddoum, G.: FLoadNet: load balancing in fog networks with cooperative multiagent using actor-critic method. IEEE Trans. Netw. Serv. Manag. **20**, 400–414 (2023)
4. Dai, F., et al.: Task offloading for vehicular edge computing with edge-cloud cooperation. World Wide Web **25**(5), 1999–2017 (2022)
5. Gawłowicz, P., Zubow, A.: ns-3 meets OpenAI gym: the playground for machine learning in networking research. In: ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (2019)
6. Geng, L., et al.: Deep reinforcement learning based distributed computation offloading in vehicular edge computing networks. IEEE Internet Things J. **10**, 12416–12433 (2023)
7. Huang, H., Ye, Q., Zhou, Y.: Deadline-aware task offloading with partially-observable deep reinforcement learning for multi-access edge computing. IEEE Trans. Netw. Sci. Eng. **9**(6), 3870–3885 (2021)
8. Jain, V., Kumar, B.: QoS-aware task offloading in fog environment using multiagent deep reinforcement learning. J. Netw. Syst. Manag. **31**(1), 7 (2023)
9. Lin, L., Zhou, W., Yang, Z., Liu, J.: Deep reinforcement learning-based task scheduling and resource allocation for NOMA-MEC in Industrial Internet of Things. Peer-to-Peer Network. Appl. **16**(1), 170–188 (2023)
10. Liu, Y., Yu, H., Xie, S., Zhang, Y.: Deep reinforcement learning for offloading and resource allocation in vehicle edge computing and networks. IEEE Trans. Veh. Technol. **68**(11), 11158–11168 (2019)
11. Mahmud, M.R., Pallewatta, S., Goudarzi, M., Buyya, R.: IFogSim2: an extended iFogSim simulator for mobility, clustering, and microservice management in edge and fog computing environments. CoRR arxiv:2109.05636 (2021)

12. Min, M., et al.: Learning-based computation offloading for IoT devices with energy harvesting. IEEE Trans. Veh. Technol. **68**(2), 1930–1941 (2019)
13. Mnih, V., et al.: Human-level control through deep reinforcement learning. Nature **518**(7540), 529–533 (2015)
14. Montresor, A., Jelasity, M.: PeerSim: a scalable P2P simulator. In: Proceedings of the 9th International Conference on Peer-to-Peer, Seattle, WA, pp. 99–100 (2009)
15. Muniswamaiah, M., Agerwala, T., Tappert, C.C.: A survey on cloudlets, mobile edge, and fog computing. In: 8th IEEE CSCloud/7th IEEE EdgeCom (2021)
16. Ng, A.Y., Harada, D., Russell, S.: Policy invariance under reward transformations: theory and application to reward shaping. In: ICML, pp. 278–287 (1999)
17. Nowé, A., Vrancx, P., De Hauwere, Y.M.: Game Theory and Multi-agent Reinforcement Learning. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-27645-3_14
18. Peng, X., et al.: Deep reinforcement learning for shared offloading strategy in vehicle edge computing. IEEE Syst. J. **17**, 2089–2100 (2022)
19. Qiu, X., et al.: Online deep reinforcement learning for computation offloading in blockchain-empowered mobile edge computing. IEEE Trans. Veh. Technol. **68**(8), 8050–8062 (2019)
20. Rausch, T, et al.: Synthesizing plausible infrastructure configurations for evaluating edge computing systems. In: 3rd USENIX Workshop HotEdge 2020 (2020)
21. Santos, J., Wauters, T., Volckaert, B., De Turck, F.: Reinforcement learning for service function chain allocation in fog computing. In: Book Chapter in revision, Submitted to Communications Network and Service Management in the Era of Artificial Intelligence and Machine Learning, IEEE Press (2020)
22. Sonmez, C., Ozgovde, A., Ersoy, C.: Edgecloudsim: an environment for performance evaluation of edge computing systems. Trans. Emerg. Telecommun. Technol. **29**(11), e3493 (2018)
23. Terry, J.K., et al.: PettingZoo: gym for multi-agent reinforcement learning. CoRR arxiv:2009.14471 (2020)
24. Tian, H., Zheng, Y., Wang, W.: Characterizing and synthesizing task dependencies of data-parallel jobs in alibaba cloud. In: Proceedings of ACM Symposium Cloud Computing (2019)
25. Tong, Z., et al.: Multi-type task offloading for wireless Internet of Things by federated deep reinforcement learning. Futur. Gener. Comput. Syst. **145**, 536–549 (2023)
26. Towers, M., et al.: Gymnasium (2023)
27. Van Le, D., Tham, C.K.: A deep reinforcement learning based offloading scheme in ad-hoc mobile clouds. In: IEEE Infocom Workshops, pp. 760–765 (2018)
28. Varghese, B., Buyya, R.: Next generation cloud computing: new trends and research directions. Futur. Gener. Comput. Syst. **79**, 849–861 (2018)
29. Yu, S., et al.: When deep reinforcement learning meets federated learning: intelligent multitimescale resource management for multiaccess edge computing in 5G ultradense network. IEEE Internet Things J. **8**(4), 2238–2251 (2020)
30. Zhang, F., et al.: Cooperative partial task offloading and resource allocation for IIoT based on decentralized multi-agent deep reinforcement learning. IEEE Internet Things J. (2023)
31. Zhu, Z., Liu, T., Yang, Y., Luo, X.: BLOT: bandit learning-based offloading of tasks in fog-enabled networks. IEEE Trans. Parallel Distrib. Syst. **30**, 2636–2649 (2019)

Robust Interaction-Based Relevance Modeling for Online e-Commerce Search

Ben Chen(✉), Huangyu Dai, Xiang Ma, Wen Jiang, and Wei Ning

Alibaba Group, Hangzhou 310052, China
benchen4395@gmail.com,
{daihuangyu.dhy,qianxiang.mx,wen.jiangw,wei.ningw}@alibaba-inc.com

Abstract. Semantic relevance calculation is vital for e-commerce search engines, as it ensures that the items selected closely align with customer intent. Inadequate attention to this aspect can detrimentally affect user experience and engagement. Traditional text-matching techniques are prevalent but often fail to capture the nuances of search intent accurately, so neural networks now have become a preferred solution to processing such complex text matching. Existing methods predominantly employ representation-based architectures, which strike a balance between high traffic capacity and low latency. However, they exhibit significant shortcomings in generalization and robustness when compared to interaction-based architectures. In this work, we introduce a robust interaction-based modeling paradigm to address these shortcomings. It encompasses 1) a dynamic length representation scheme for expedited inference, 2) a professional terms recognition method to identify subjects and core attributes from complex sentence structures, and 3) a contrastive adversarial training protocol to bolster the model's robustness and matching capabilities. Extensive offline evaluations demonstrate the superior robustness and effectiveness of our approach, and online A/B testing confirms its ability to improve relevance in the same exposure position, resulting in more clicks and conversions. To the best of our knowledge, this method is the first interaction-based approach for large e-commerce search relevance calculation. Notably, we have deployed it for the entire search traffic on alibaba.com, the largest B2B e-commerce platform in the world.

Keywords: E-commerce search · Semantic relevance · Interaction-based

1 Introduction

In online e-commerce platforms, the search engine's effectiveness hinges on two core capabilities: identifying users' explicit demands through queries and mining purchasing preferences from historical click logs. A comprehensive search solution

---

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-70378-2_4.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
A. Bifet et al. (Eds.): ECML PKDD 2024, LNAI 14949, pp. 55–71, 2024.
https://doi.org/10.1007/978-3-031-70378-2_4

requires not only a ranking module that enhances click-through and conversion rates but also a relevance module that validates the appropriateness of displayed items. **S**emantic **R**elevance **C**alculation (SRC), a fundamental component of e-commerce platforms, discerns core keywords within short queries against long item descriptions to accurately score and rank pertinent items [3,4,12]. Prioritizing co-click conversion modeling without considering users' intent and matching relevance can erode user attention and, consequently, engagement and conversion rates over time.

Semantic relevance calculation can be classified as a domain-specific text matching task, markedly distinct from general tasks such as MS MARCO [23] and STS [2], which measure semantic similarity in standard language contexts, or semantic question answering (matching) [1,6,18], which concentrate on the primary themes of queries and documents. SRC for online e-commerce search faces distinct challenges:

1) Query Intent and Keyword Clarity. SRC must distinguish concise user queries that often carry vague meanings and match them with the most matching items. A query like'new apple discount' could ambiguously refer to a promotion on fresh produce, a reduced price on Apple electronics, or a clothing brand's latest offer. Moreover, product descriptions are frequently stuffed with extraneous keywords to gain more exposure, such as a dress described as "elegant evening gown summer crystal luxury sequin red cocktail party", which dilutes the significance of essential keywords and muddles the search accuracy.
2) Latency vs. Precision Trade-off. As e-commerce platforms strive for efficiency, the shift from traditional keyword-based search algorithms to advanced neural-based models marks a significant progression [30,42,43]. These models are divided into two types: representation-based and interaction-based. The former models, leveraging siamese network architectures, encode queries and items into compact embeddings efficiently, making them suitable for high-traffic online searches according to their computational speed. However, their oversimplification often leads to poorer relevance predictions. In contrast, interaction-based models excel in capturing subtle semantic relationships, offering finer distinction and accuracy, but the intensive computations limit their practical application in real-time scenarios with stringent latency requirements.
3) Enhancing Robustness and Generalization. The diverse linguistic expressions arising from cultural differences complicate the accuracy of SRC models, such as various terms for the same discounts of "50% off sale", "half price promotion", and "discounted by half". Meanwhile, to reduce computational overhead, both representation- and interaction-based methods often use techniques like pruning or distillation [10,15,37] to simplify models for efficient real-time processing. However, these condensed models, while effective on familiar data, struggle with unfamiliar pairs, revealing limitations in robustness and generalization. An efficient solution is to expand the training data diversity, but manual annotation is resource-intensive with limited scope, and noisy histor-

ical data dilute their effectiveness. These further weaken models' generalization ability.

In summary, while conventional interaction-based models (BERT) can achieve state-of-the-art performances on real query-item search logs, its computational intensity remains a barrier to its direct implementation in online search engines, despite efforts to mitigate this through distillation and pruning. Consequently, the less computationally demanding representation-based models are often the default choice. To address this challenge, in this paper we introduce a robust interaction-based method for relevance modeling. It encompasses three key innovations. The first is a dynamic-length representation scheme. It can intelligently scale input token size to match the varying lengths of queries and item descriptions, thereby optimizing computational resources. The second is an efficient professional terms recognition strategy. It enhances the model's vocabulary with industry-specific phrases and employs Named Entity Recognition (NER) to highlight subjects and core attributes, so as to reinforce the representation of professional terminology. Finally, to counteract the performance limitations of shallow models, we devised a contrastive adversarial training (CAT) mechanism. It can bolster the model's generalization and robustness by simultaneously optimizing the embedding representations of both inputs and outputs. Impressively, this optimized model, with just 3 layers, outperforms traditional 12-layer BERT base models in efficiency and effectiveness.

We conducted comprehensive offline evaluations using annotated query-item pairs derived from online search logs, and the results showcased significant performance enhancements, affirming the efficacy and robustness of our proposed method. Online A/B tests also demonstrated that it can improve the matching relevance of query-item in the same exposure position, and attract more clicks and conversions. To the best of our knowledge, it is the first interaction-based relevance calculation work for a large-scale e-commerce search engine, accommodating the daily needs of tens of millions of users and serving billions of retrieval page views. Moreover, this method has been successfully deployed for the entire search traffic on alibaba.com, the world's largest B2B e-commerce platform, and has yielded substantial improvements in conversion rates across the board.

2 Related Work

Semantic relevance computation is essentially a text matching task that benefits from various techniques. These range from traditional methods like TF-IDF and BM25 [20,30] to machine learning approaches such as DSSM [13], LSTMs [26,27,35], and CNNs [3,28,31]. These models have limitations in bridging the vocabulary gap and accurately identifying key terms. With the rapid development of BERT-based models enhancing performance across NLP tasks [5,9,16,19,33,34], some advanced implementations like Sentence-BERT, BERT-flow, and SimCSE [7,17,29,32] have emerged as the preferred methods for complex text processing. However, these architectures still struggle with capturing the subtleties of search intent and are generally resource-intensive. Consequently, tailored strategies are necessary to overcome these specific challenges.

A common paradigm utilizes a softmax function on the final [CLS] token or average pooling outputs which are then scaled to a range between 0 and 1 to represent the likelihood of relevance [4,24]. Nogueira et al. implemented a multi-stage BERT-based architecture for ranking, incorporating innovative point-wise and pair-wise classification strategies [25]. Wu et al. introduced a multi-task learning framework aimed at minimizing the query-item vocabulary gap while optimizing multiple objectives [41]. Jiang et al. developed a data-driven relevance prediction framework by distilling knowledge from BERT and its sophisticated teacher models [14]. Garakani et al. utilized a cross-encoder BERT model for query-item relevance prediction, further applying it to re-ranking and the optimization of search quality [8]. While these methodologies surpass the accuracy of representation-based models, the vast traffic volume and the stringent latency requirements of live search environments present significant deployment challenges. To address this, ReprBERT [43] proposes a unique solution that distills the interaction-based capabilities of BERT into a more streamlined representation-based model, employing dual interactive strategies to refine latent semantic interactions. This results in superior performance relative to conventional representation-based models. Moreover, the very recent Interactor [44] can capture fine-grained phrase-level information with a flexible contextualized interaction paradigm, and adopts a novel partial attention scheme to reduce the computational cost while maintaining the effectiveness. Nonetheless, their representational ability does not match that of fully interactive methods, and the SRC performance cannot quite reach the benchmark set by the latter.

3 Methodology

We use **ei-SRC** to indicate the proposed e-commerce interaction-based semantic relevance calculation method for brevity, and decompose its methodology into three components. Firstly, we outline two interaction strategies designed to minimize online computational overhead. Subsequently, we describe techniques developed to enhance the model's proficiency in handling domain-specific terminology. Lastly, we propose a novel training mechanism aimed at fortifying the model's representational capacity and accuracy in relevance matching. The comprehensive architecture of the ei-SRC method is depicted in Fig. 1.

3.1 Dynamic-Length Representation Scheme

The primary distinctions between representation- and interaction-based methodologies in SRC are rooted in their respective processing of queries and the associated computational frameworks. Representation-based techniques typically convert a query into a fixed-dimension vector. By pre-computing item embeddings offline and employing scoring functions like dot-product or cosine similarity online, these methods significantly streamline computational overheads, with the most resource-intensive step being the preliminary query processing. Conversely, interaction-based methods exhibit a higher sensitivity to query length.

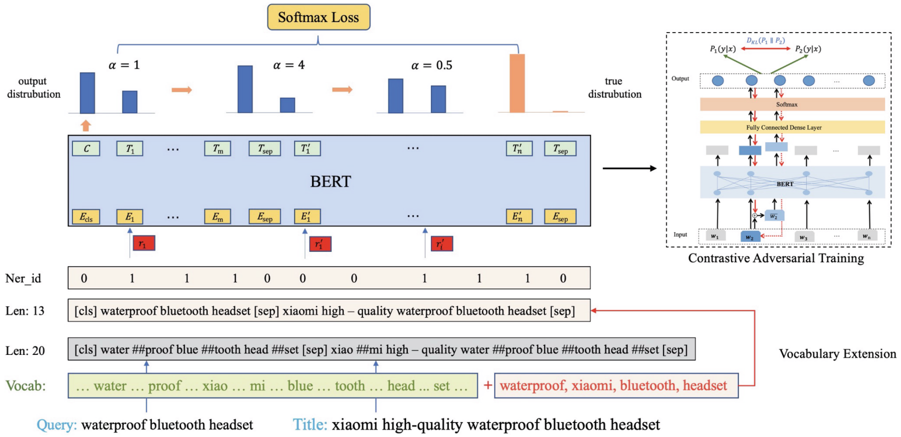

Softmax Loss
output distrubution
α = 1
α = 4
α = 0.5
true distrubution
C
T1
Tm
Tsep
T'1
T'n
Tsep
BERT
Ecls
E1
Em
Esep
E'1
E'n
Esep
r1
r'1
r'i
P1(y|x)
DKL(P1 ∥ P2)
P2(y|x)
Output
Softmax
Fully Connected Dense Layer
BERT
Input
w1
w2
w3
wn
Contrastive Adversarial Training
Ner_id
0 1 1 1 0 0 0 1 1 1 0
Len: 13
[cls] waterproof bluetooth headset [sep] xiaomi high – quality waterproof bluetooth headset [sep]
Len: 20
[cls] water ##proof blue ##tooth head ##set [sep] xiao ##mi high – quality water ##proof blue ##tooth head ##set [sep]
Vocab:
… water … proof … xiao … mi … blue … tooth … head ... set …
+ waterproof, xiaomi, bluetooth, headset
Vocabulary Extension
Query: waterproof bluetooth headset
Title: xiaomi high-quality waterproof bluetooth headset

Fig. 1. The overview of the ei-SRC, the proposed e-commerce interaction-based semantic relevance calculation method. For brevity, the dynamic-length representation scheme is not illustrated, which will be displayed separately.

They operate by processing the query in real-time, which involves grammatical normalization and tokenization to generate a sequence of tokens. These tokens are then merged with pre-processed item description tokens, culminating in an interactive and dynamic computation sequence. Ultimately, a non-linear classifier is utilized to determine the relevance score, providing a detailed assessment of the relationship between the query and the item [24].

To address the varied requirements of industrial search applications and simultaneously evaluate thousands of item candidates per query, interaction-based methods typically pre-define a fixed token length for both queries and item descriptions. This standardization ensures uniform input sizes and consistent computational time across all pairings. However, this token length is often set longer to account for the occasional lengthy query or item description, leading to sub-optimal computational resource use for processing shorter texts. Such inefficiency impedes the real-world deployment of interaction-based methods in an online search platform.

Prior to introducing our optimization strategy, it is essential to explore the factors influencing the computational load of interaction-based models. The BERT framework is known for its multi-layered structure, where each layer principally consists of two core modules: the multi-head attention (MHA) and feed-forward neural network (FFN) sub-layers. Here we indicate n as the batch size during training or the number of item candidates during online inference, l as the total length of the input token sequence, and d as the dimension of the embeddings. The number of attention heads is represented by m, and the attention head is set as a, such that $d = ma$.

The time complexity of multi-head attention sub-layers is:

$$T_{MHA} = O(n * l^2 * m^2 * a) = O(n * l^2 * d * m). \tag{1}$$

The time complexity of FFN sub-layers is:

$$T_{FFN} = O(n * l * k * m). \tag{2}$$

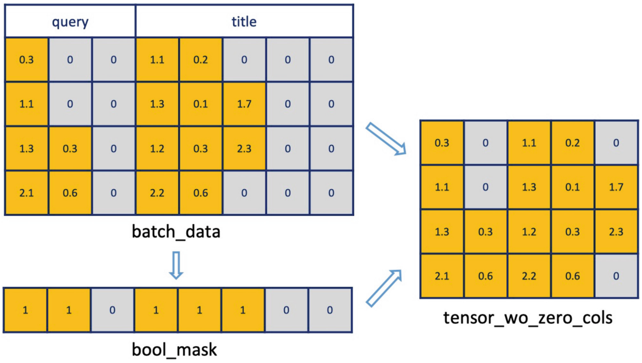


Fig. 2. The overview of dynamic-length representation scheme, where each square is a token. The number shown on it simulates the sum of various types of embeddings.

As model distilling or pruning could minimize k, n, and d, we can see that T_{MHA} is proportional to the square of l, and T_{FFN} have a linear correlation with l. If l can be decreased, the total time consumption will be largely reduced.

Here we design a dynamic-length representation scheme to address this problem. Considering that the descriptions of queries and items are shorter than the settings for most cases, we can shorten the tokens' length by batch-dropping zero padding columns. To describe it precisely, we indicate l_q as the query token length set in advance, and l_i as the length of title tokens, then $l = l_q + l_i$. As shown in Fig. 2, we calculate the max non-zero token length of queries (l'_q) and items titles (l'_t) in batch, and cut the full zero padding columns. The size of new input series is $l' = l'_q + l'_i$, and time complexity T_{MHA} is reduced to the original $(l'/l)^2$ times.

Table 1. Improvements in computational performance for each strategic combination employed within the **ei-SRC**.

Method	GPU utilization	reaction latency	AUC
DRS	−34.61%	−36.26%	+0.4%
\+Cache	−48.08%	−40.66%	–
\+Vocab.	−53.84%	−46.15%	+1.5%
\+CAT.	−51.25%	−44.83%	+6.3%

Furthermore, we also pre-compute the tokens in advance for queries that the user mostly entered, the relevance scores for high-frequency query-item pairs, and then store them in the online cache to reduce the computation cost. The proportion of pre-computed pairs varies with the actual search. For example, we can use the top 20% data in the past six months and implement daily updates.

Table 2. Top 20 words mostly used in alibaba.com, they would be split into many sub-tokens with the original WPE.

rohs	waterproof	100%	oem	pvc
fokison	smd	cnc	acrylic	polyester
osc	bluetooth	capacitors	12v	scooter
mink	diy	wifi	diode	hoodie

Table 3. Average sub-tokens numbers calculation for query-items pairs with original WPE and extended vocabulary.

Types	original word	original WPE	extended vocab.
each word	1.00	1.37	1.22
each title	15.61	21.46	19.06
each pairs	26.41	41.62	36.55

In the real online A/B tests on alibaba.com, this method significantly reduced the computational cost and reaction latency without attenuation of the accuracy. Model architecture and implementation configurations will be detailed in the experimental section. Now we only list the comparison of calculations consumption in Table 1, where "DRS" indicates "Dynamic-length Representation Scheme", and "Cache" means the pre-computed results stored in the online cache. As for the computational performance, we use the GPU utilization, and reaction latency as the metrics.

We can see that DRS can significantly reduce GPU utilization and response latency. Thanks to the reduction of interference caused by all zero padding columns, semantic representation and matching of query-item pairs have been enhanced, with a 0.4% increase in AUC. In addition, offline pre-computation and online caching of query tokens and relevance scores for high-frequency pairs can significantly reduce the overhead of those repeated searches. Finally, these two methods combined contribute to the reduction of GPU utilization by 48%, and response latency by 40%, thereby easing the final resource consumption.

3.2 Professional Terms Recognition Strategy

A notable feature of the tokenization tools WPE [5] and BPE [19] is that they do not cause OOV (Out Of Vocabulary) problems. BERT-based models can effectively combine the split tokens of a single word to express the semantic information well if they have trained on large corpora containing that word multiple times. However, in the e-commerce domain, queries and titles often contain specialized terms that are infrequently found in general corpora. This disparity can result in a misinterpretation of their intended meanings due to standard tokenization methods. For example, the term 'bluetooth' is typically segmented into "blue" and "##tooth", and a lack of adequate training can lead to errors in matching. Consequently, substituting 'bluetooth' with variants like 'blue tooth' or even 'black tooth' in queries and titles might not significantly alter the calculated relevance score in matching, thereby illustrating a potential pitfall in accurately capturing the essence of e-commerce vocabulary.

To address this problem, here we propose an efficient professional terms recognition strategy. It contains two steps: First is the vocabulary extension. We calculated the frequency of each word contained in queries and item descriptions on alibaba.com in the past 12 months, selecting the top 20,000 words which will be split into many sub-tokens with the former tokenization, and added them into the original WPE vocabulary of BERT. Then the model would be continued pre-trained with the new vocabulary.

The top 20 words are listed in Table 2, we can see that extended words are obviously related to the e-commerce field. Moreover, we calculated the average number of sub-tokens for query-item pairs with these two vocabularies separately. As depicted in Table 3, the overall number of each pair is reduced from 41 to 36, and it achieves a computational saving of nearly 22.81% with Equation (1). For real online testings, we also find it can reduce the GPU utilization and response latency significantly, and further improve the metric AUC by nearly 1.5%, as shown in Table 1. That is, it can not only ease the inference burden but also contribute to the enhancement of representations for specific terms.

Additionally, we implement NER [38,39] to recognize the object and core keywords. We use the numbers 1-5 to mark "material", "function", "usage", "specification", "style", and "core" keywords in query-item pairs, and generate the corresponding NER embeddings, which combine the token\segmentation\position embeddings to from the new input embeddings.

3.3 Contrastive Adversarial Training Mechanism

In order to reduce the time-consuming of online relevance inference, pruning or distillation is often adopted to get the shallow-layer models, resulting in the degradation of the model's generalizations to untrained pairs. A useful tip is training these models with a large range of effective data. However, highly reliable annotated pairs are costly and have low coverage, and the data sampling from user behaviors is more noisy because of many unintentional clicks. These all limit the online application of interaction-based methods.

To tackle this problem, here we propose one novel contrastive adversarial training (CAT) mechanism. It aims to improve the robustness of both input and output representations simultaneously and make better discrimination on the hard samples.

The CAT method is detailed as follows. We present query-item pair as $< q_i, p_i >$ (combined as the input x_i) and the relevance label as $y_i \in \{0, 1\}$. θ denotes the parameters of the corresponding model, and the basic objective for the SRC model is to minimize the negative log-likelihood loss function as:

$$L_{BCE} = -\frac{1}{N}\sum_{i=1}^{n} \log p(y_i|x_i, \theta)). \tag{3}$$

For the first step, we adopt adversarial training [22,47] to reduce the model's sensitivity to perturbed input embeddings, and then improve robustness to various original examples. Specifically, we first calculate a small adversarial perturbation r_{adv} based on the back-propagated gradient value:

$$r^i_{adv} = -\epsilon \mathbf{g} / \|\mathbf{g}\|_2 \text{ where } \mathbf{g} = \nabla x_i \log p(y_i|x_i, \theta), \tag{4}$$

where ϵ is the coefficient weight to control the size of perturbation, then add it to the original input x to form one new perturbed embedding $x + r_{adv}$, and finally design a new function to fit the relevance label:

$$L_{ATE} = -\frac{1}{N}\sum_{i=1}^{n} \log p(y_i|x_i + r^i_{adv}, \theta). \tag{5}$$

.

On the other hand, aiming to enhance the output representation, we try to minimize the difference between two output distributions from the same model with different dropouts. This strategy is adopted for two reasons: one is to eliminate the nonnegligible inconsistency between training and inference caused by the randomness introduced by dropout [11,40], and the other is to enhance the output representation. In detail, we randomly discard some non-substantial words filtered out by NER, to construct a new query-item pair with substantially the same semantics for each inference. Then the bidirectional KL divergence of their output passing through the same model should be kept minimal.

However, the simple combination of these two methods would cause four inferences, leading to a significant increase in computational cost. Considering the first inference of adversarial training only to be conducted for the r_{adv} generation without any parameter updating, here we design one optimized scheme. That is, we retain the back-propagation of L_{BCE}, and then minimize the bidirectional KL divergence of output distributions between x and x_{adv} as:

$$\begin{aligned} L_{ADV} = \frac{1}{2N}\sum_{i=1}^{n} &(D_{KL}(p(y_i|x_i, \theta)||p(y_i|x_i + r^i_{adv}, \theta)) \\ &+ D_{KL}(p(y_i|x_i + r^i_{adv}, \theta)||p(y_i|x_i, \theta))). \end{aligned} \tag{6}$$

.

So all the computations only need two inferences, and the total loss is:

$$L_{total} = \alpha_1 \cdot L_{BCE} + \alpha_2 \cdot L_{ATE} + \alpha_3 \cdot L_{ADV} \tag{7}$$

where α_1, α_2 and α_3 are the weighting parameters.

Moreover, in order to remind the model to emphasize the hard mining examples, we replace the original softmax function with heated-up softmax [46]:

$$p_{(m_i|x_i,\theta)} = \frac{exp(\alpha * z_i)}{exp(\alpha * z_0) + exp(\alpha * z_1)}, \tag{8}$$

where z_i is the output logit of x_i, m is the binary category with $m_i \in 0, 1$, and α denotes the temperature parameter.

We start the training process with a large α, to concentrate on hard samples. Then we gradually reduce α to shift the model's attention to boundary samples, and eventually, we set a minimal value for fine-tuning the easy pairs. As depicted in Fig. 1, α adjusts the distribution of the last layer output and enables the model to focus on more hard mining samples as the training goes on.

As a side note, prior work [21] proposed a similar contrastive representation adversarial learning for text classification. However, it only cares about the input representation by minimizing the difference of outputs with increasing ϵ in Equation (4), but our CAT mechanism is designed to regularize the model by optimizing both input and output representations. The offline experiments below show that it can improve not only the robustness to examples with keyword repetition and stacking but also enhance the generalization for query-items matching of multiple industries.

4 Experiments

In this section, we conduct comprehensive evaluations on manually annotated query-item pairs offline and rigorous A/B online testings online to verify the feasibility of the proposed methods.

Dataset. We extracted the highly reliable user-clicked pairs from the online search logs of alibaba.com over the past year to facilitate the continue pre-training. The selection of pairs was subject to two constraints: 1) Each query had to comprise a minimum of two words, including at least one core keyword, to filter out queries with ambiguous semantics such as 'dress' or 'sports shoes,' which often reflect unclear user search intents and result in a scattered range of clicked items. 2) Pairs exhibiting deeper click behaviors were re-sampled to a certain number, predicated on the basic cognition that query reflects the user's intent and their clicks signify varying degrees of interest in different products. For alibaba.com, the cross-border B2B platform, user click behaviors are categorized into five escalating levels: page-click $\leq$ add-to-cart $\leq$ contact-supplier $\leq$ order $\leq$ pay. Based on multiple experimental tests, here we set the re-sampled number for each level as (1,1,2,3,5). In total, 80 million query-item pairs were collected, with 70 million designated for continuing pre-training, and the remaining 10 million high-frequency ones were reserved for subsequent fine-tuning.

Additionally, we further gathered a dataset of 250,000 query-item pairs and marked them as relevant or irrelevant via manual annotation. The relevance judgments were based on three criteria: the subject, the presence of core keywords, and units of measurement such as minimum order quantity, delivery time, and size specifications. A pair was marked as relevant only if it satisfied all three conditions. Each pair was assessed by three individuals to mitigate subjective bias. To further ensure accuracy, an expert specializing in search business reviewed the collective judgments. Subsequently, 150,000 annotated pairs were utilized for additional fine-tuning, while the remaining 100,000 served as the evaluation set.

Baselines. For comprehensive evaluations, we utilize the widely-employed models BERT, RoBERTa, StructBERT [36], and the very recent domain-specific models ReprBERT [43], as well as Interactor [44]. All were pre-trained on the same dataset as base models. Subsequently, we distilled BERT into smaller variants (L3-H128-A4), denoted as BERT$_{mini}$. Additionally, we trained a Sentence-BERT (SBERT) [29] and its distilled counterpart (SBERT$_{mini}$) to compare the efficacy of representation-based models. All models were initially fine-tuned on 10 million online query-item pairs and subsequently on annotated datasets.

Implementation Details. During pre-training phase, the learning rate and batch size were configured to 5e-4 and 32 respectively. For fine-tuning, we adjusted the learning rate to 2e-5, increased the batch size to 1024, and set the weighting parameters $\alpha_{1,2,3}$ as (0.5, 0.5, 0.01). The maximum token lengths were established at 16 for queries and 36 for item descriptions, while the actual input token lengths varied according to the dynamic-length representation scheme. All models were fine-tuned within 3 epochs with early-stopping. We employed the batch negative sampling strategy for the initial stage on 10 million user-clicked pairs. All experiments were conducted using Tensorflow 1.12 for both online and offline evaluations, with the implementations available for reproducibility.

4.1 Offline Evaluation

For evaluating SRC as a binary classification task, we utilized AUC, Micro/Macro F1 scores, and Spearman's and Pearson's correlation coefficients as evaluated metrics. As depicted in Table 4, comparative results for base models (X_{base} with 12 layers) and the scaled-down mini models (X_{mini} with 3 layers) are presented for both representation-based and interaction-based approaches. It was observed that, at comparable parameter scales, representation-based models lagged in performance, exhibiting an approximate 3.5% decrease in AUC when compared to interaction-based models. Furthermore, the X_{base} models outperformed their X_{mini} counterparts, reflecting a more robust interaction and representational capacity in larger models. Nonetheless, the computational intensity of X_{base} models imposes limitations for their practical deployment.

Notably, despite the parity in parameter magnitude, ei-SRC surpasses the performance of two mini models SBERT$mini$ and BERT$mini$, with improvements of 10.37% and 6.27% respectively. Moreover, ei-SRC demonstrates superior performance across all metrics, even when compared to larger representation-based model SBERT$_{base}$ and interaction-based models BERT, RoBERTa, StructBERT, and recent domain-specific model ReprBERT, as well as Interactor. Remarkably, the 3-layer configuration of our model attains better results than the conventional 12-layer models on real-world search query-item pair evaluations. These findings not only underscore the efficacy of the methods proposed but also highlight the model's robustness and its ability to generalize across a diverse range of industrial applications.

To validate the efficacy of proposed strategies, we carried out supplementary ablation studies. These included assessments of computational performance, as summarized in Table 1, and comparisons of evaluated metrics, as detailed in Table 5. We observed a marked decrease in computational cost with the incremental application of dynamic-length representation (DRS), which led to a reduction of 34.61%. The introduction of the online cache strategy contributed to a further reduction of 48.08%, while the expansion of the vocabulary (Vocab.) yielded a decrease of 53.84% in computational overhead. On the other hand, the incorporation of contrastive adversarial training (CAT) resulted in increased processing time due to its requirement for dual inferences. Nevertheless, CAT significantly enhanced the performance across all metrics: AUC increased by 5.85%, F1 scores by 22.72%/7.79%, Spearmanr by 9.21%, and Pearsonr by 35.74%.

Table 4. Comparison results of representation and interaction-based methods on manual annotated data

Strategy	AUC	F1	Spearmanr	Pearsonr
$SBERT_m$	0.8184	0.61/0.71	0.5515	0.3934
$SBERT_b$	0.8554	0.60/0.76	0.6061	0.4751
$BERT_m$	0.8500	0.65/0.77	0.6147	0.5079
$BERT_b$	0.8907	0.72/0.78	0.6742	0.6360
$ReprBERT_b$	0.8923	0.73/0.80	0.6804	0.6389
$Interactor_b$	0.8926	0.73/0.81	0.6810	0.6387
$RoBERTa_b$	0.8964	0.78/0.82	0.6865	0.6926
$StructBERT_b$	0.9011	0.78/0.81	0.6947	0.7096
ei-SRC (3L)	**0.9033**	**0.81/0.83**	**0.6984**	**0.7181**

Table 5. Ablation experiment results of increment strategies

Method	AUC	F1	Spearmanr	Pearsonr
$BERT_m$	0.8500	0.65/0.77	0.6147	0.5079
\+DRS	0.8533	0.66/0.77	0.6395	0.5290
\+AT	0.8852	0.75/0.80	0.6720	0.6404
\+CAT	0.9033	0.81/0.83	0.6984	0.7181

As illustrated in Table 5, the three strategies collectively improved the model's matching results to different degrees. In conclusion, the combined application of these methods can conserve nearly half of the computational resources while substantially enhancing the model's relevance-matching capabilities.

4.2 Online Experiments

We implemented the ei-SRC model within the online search engine of alibaba.com, the largest B2B e-commerce platform in the world, which serves tens of millions of users generating billions of page-views (PVs) daily. The SRC module is integrated into the final stage of the search ranking process. Initially, it evaluates all candidate items-typically between 3,000 to 5,000-and filters out those with low correlation scores to ensure that no more than 2,000 items are selected for ranking. Subsequently, for each query-item pair, the SRC score is combined with the ranking model's scores and other factors to determine the final rank, so that one pair with higher SRC score would be listed in more forward exposure position.

Table 6. Online A/B testing results, where P-value means the statistical significance

Strategy	CTR	CVR	PAY	P-value
\+DRS	+0.93%	+1.09%	+0.76%	0.018
\+AT	+0.68%	+1.37%	+1.66%	0.023
\+CAT	+1.54%	+1.02%	+1.87%	0.004

Table 7. Manual evaluation results for online relevance, all testing pairs are extracted in the same exposure position

Strategy	Good	Fair	Bad
\+DRS	+0.91%	−0.93%	−3.61%
\+AT	+4.01%	−21.67%	−3.13%
\+CAT	+4.77%	−3.87%	−8.93%

In detail, the initial online model was a variant of sBERT_{mini}, and we sequentially conducted three A/B testings: BERT_{mini} with Dynamic-length Representation Scheme (\+DRS), followed by the inclusion of Adversarial Training (\+AT), and finally, the incorporation of Contrastive Adversarial Training (\+CAT). We assessed the impact of each strategy on click through rate (CTR), average conversion rate (CVR), and average earnings rate (PAY) by comparing performance metrics before and after each implementation. Notably, the improvements reported are incremental relative to the preceding iteration model rather than the original BERT_{mini}, and the statistical significance (P-value) of all results was below 0.05. As indicated in Table 6, there were substantial increases in CTR, CVR, and PAY after the application of each strategy, with the CAT strategy demonstrating particularly notable enhancements: CTR improved by 1.54%, CVR by 1.02%, and PAY by 1.87%.

Furthermore, to ascertain the actual impact on online search relevance, we conducted additional manual evaluations. Following each update, we randomly selected queries and extracted 2000 query-item pairs from identical exposure positions, ensuring all other variables remained constant. We engaged experts to rate each pair as 'Good' (both subject and core keywords match), 'Fair' (only subject matches), or 'Bad' (subjects differ). The outcomes of these assessments are presented in Table 7. The results indicate that each implemented strategy incrementally improved semantic relevance, with the incorporation of CAT yielding the most significant enhancement. In conclusion, our proposed method was demonstrated to substantially improve the user search experience, leading to increased clicks and conversions, and ultimately boosting the industry revenue.

Additional evaluation results, annotated datasets, and codes for dynamic-length representation scheme and contrastive adversarial training mechanism will be publicly accessible to facilitate further research. We intend to expand our investigation into the integration of the ei-SRC framework within ranking models and its application to recommendation systems and P4P advertising.

5 Conclusion

In this study, we introduced a robust interaction-based method for modeling semantic relevance in online e-commerce search engines. This method incorporates a dynamic length representation scheme, a professional terms recognition strategy, and a contrastive adversarial training scheme to improve relevance matching. Extensive experiments on offline annotated query-item pairs and rigorous online A/B tests have verified its effectiveness for enhancing the search experience and boosting the industry revenue. Significantly, this approach has been stably operational on www.alibaba.com, handling its full search traffic for over 12 months, undergoing several iterative improvements.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Antoniou, C., Bassiliades, N.: A survey on semantic question answering systems. Knowl. Eng. Rev. **37**, e2 (2022)
2. Cer, D., Diab, M., Agirre, E., Lopez-Gazpio, I., Specia, L.: Semantic textual similarity-multilingual and cross-lingual focused evaluation. In: Proceedings of the 2017 SEMVAL International Workshop on Semantic Evaluation. (2017)

3. Chen, Y.: Convolutional neural network for sentence classification. Master's thesis, University of Waterloo (2015)
4. Chen, Z., Chen, W., Xu, J., Liu, Z., Zhang, W.: Beyond semantics: learning a behavior augmented relevance model with self-supervised learning. In: Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (2023)
5. Devlin, J.: BERT: pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
6. Dhakal, A., Poudel, A., Pandey, S., Gaire, S., Baral, H.P.: Exploring deep learning in semantic question matching. In: 2018 IEEE 3rd International Conference on Computing, Communication and Security (ICCCS), pp. 86–91. IEEE (2018)
7. Gao, T., Yao, X., Chen, D.: SimCSE: simple contrastive learning of sentence embeddings. arXiv preprint arXiv:2104.08821 (2021)
8. Garakani, A.B., et al: Improving relevance quality in product search using high-precision query-product semantic similarity. In: Proceedings of The Fifth Workshop on e-Commerce and NLP (ECNLP 5), pp. 44–48 (2022)
9. He, P., Liu, X., Gao, J., Chen, W.: DeBERTa: decoding-enhanced BERT with disentangled attention. arXiv preprint arXiv:2006.03654 (2020)
10. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 (2015)
11. Hinton, G.E., Srivastava, N., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.R.: Improving neural networks by preventing co-adaptation of feature detectors. arXiv preprint arXiv:1207.0580 (2012)
12. Hu, W., Dang, A., Tan, Y.: A survey of state-of-the-art short text matching algorithms. In: Tan, Y., Shi, Y. (eds.) DMBD 2019. CCIS, vol. 1071, pp. 211–219. Springer, Singapore (2019). https://doi.org/10.1007/978-981-32-9563-6_22
13. Huang, P.S., He, X., Gao, J., Deng, L., Acero, A., Heck, L.: Learning deep structured semantic models for web search using clickthrough data. In: Proceedings of the 22nd ACM international conference on Information & Knowledge Management, pp. 2333–2338 (2013)
14. Jiang, Y., et al.: BERT2DNN: BERT distillation with massive unlabeled data for online e-commerce search. In: 2020 IEEE International Conference on Data Mining (ICDM), pp. 212–221. IEEE (2020)
15. Jiao, X., et al: TinyBERT: distilling BERT for natural language understanding. arXiv preprint arXiv:1909.10351 (2019)
16. Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., Soricut, R.: ALBERT: a lite BERT for self-supervised learning of language representations. arXiv preprint arXiv:1909.11942 (2019)
17. Li, B., Zhou, H., He, J., Wang, M., Yang, Y., Li, L.: On the sentence embeddings from pre-trained language models. arXiv preprint arXiv:2011.05864 (2020)
18. Liu, X., et al.: LCQMC: a large-scale Chinese question matching corpus. In: Proceedings of the 27th International Conference on Computational Linguistics, pp. 1952–1962 (2018)
19. Liu, Y., et al.: RoBERTa: a robustly optimized BERT pretraining approach. arXiv preprint arXiv:1907.11692 (2019)
20. Manning, C.D.: An Introduction to Information Retrieval. Cambridge University Press (2009)
21. Miao, D., et al.: Simple contrastive representation adversarial learning for NLP tasks. arXiv preprint arXiv:2111.13301 (2021)
22. Miyato, T., Dai, A.M., Goodfellow, I.: Adversarial training methods for semi-supervised text classification. arXiv preprint arXiv:1605.07725 (2016)

23. Nguyen, T., et al.: Ms marco: a human generated machine reading comprehension dataset. Choice **2640**, 660 (2016)
24. Nogueira, R., Cho, K.: Passage re-ranking with BERT. arXiv preprint arXiv:1901.04085 (2019)
25. Nogueira, R., Yang, W., Cho, K., Lin, J.: Multi-stage document ranking with BERT. arXiv preprint arXiv:1910.14424 (2019)
26. Palangi, H., et al.: Semantic modelling with long-short-term memory for information retrieval. arXiv preprint arXiv:1412.6629 (2014)
27. Palangi, H., et al.: Deep sentence embedding using long short-term memory networks: analysis and application to information retrieval. IEEE/ACM Trans. Audio Speech Lang. Process. **24**(4), 694–707 (2016)
28. Pang, L., Lan, Y., Guo, J., Xu, J., Wan, S., Cheng, X.: Text matching as image recognition. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 30 (2016)
29. Reimers, N., Gurevych, I.: Sentence-BERT: sentence embeddings using siamese BERT-networks. arXiv preprint arXiv:1908.10084 (2019)
30. Robertson, S.E., Walker, S., Jones, S., Hancock-Beaulieu, M.M., Gatford, M.: Okapi at trec-3. Nist Special Publication Sp **109**, 109 (1995)
31. Shen, Y., He, X., Gao, J., Deng, L., Mesnil, G.: A latent semantic model with convolutional-pooling structure for information retrieval. In: Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, pp. 101–110 (2014)
32. Su, J., Cao, J., Liu, W., Ou, Y.: Whitening sentence representations for better semantics and faster retrieval. arXiv preprint arXiv:2103.15316 (2021)
33. Sun, Y., et al.: Ernie 3.0: large-scale knowledge enhanced pre-training for language understanding and generation. arXiv preprint arXiv:2107.02137 (2021)
34. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, vol. 30 (2017)
35. Wan, S., Lan, Y., Xu, J., Guo, J., Pang, L., Cheng, X.: Match-SRNN: modeling the recursive matching structure with spatial RNN. arXiv preprint arXiv:1604.04378 (2016)
36. Wang, W., et al.: StructBERT: incorporating language structures into pre-training for deep language understanding. arXiv preprint arXiv:1908.04577 (2019)
37. Wang, W., Wei, F., Dong, L., Bao, H., Yang, N., Zhou, M.: MiniLM: deep self-attention distillation for task-agnostic compression of pre-trained transformers. Adv. Neural. Inf. Process. Syst. **33**, 5776–5788 (2020)
38. Wang, X., Jiang, Y., Bach, N., Wang, T., Huang, F., Tu, K.: Structure-level knowledge distillation for multilingual sequence labeling. arXiv preprint arXiv:2004.03846 (2020)
39. Wang, X., et al.: Automated concatenation of embeddings for structured prediction. arXiv preprint arXiv:2010.05006 (2020)
40. Wu, L., et al.: R-drop: regularized dropout for neural networks. Adv. Neural. Inf. Process. Syst. **34**, 10890–10905 (2021)
41. Wu, X., Magnani, A., Chaidaroon, S., Puthenputhussery, A., Liao, C., Fang, Y.: A multi-task learning framework for product ranking with BERT. In: Proceedings of the ACM Web Conference 2022, pp. 493–501 (2022)
42. Xiao, R., et al.: Weakly supervised co-training of query rewriting andsemantic matching for e-commerce. In: Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, pp. 402–410 (2019)

43. Yao, S., Tan, J., Chen, X., Zhang, J., Zeng, X., Yang, K.: ReprBERT: distilling BERT to an efficient representation-based relevance model for e-commerce. In: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp. 4363–4371 (2022)
44. Ye, W., et al.: Fast semantic matching via flexible contextualized interaction. In: Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining, pp. 1275–1283 (2022)
45. Zeng, A., et al.: GLM-130B: an open bilingual pre-trained model. arXiv preprint arXiv:2210.02414 (2022)
46. Zhang, X., Yu, F.X., Karaman, S., Zhang, W., Chang, S.F.: Heated-up softmax embedding. arXiv preprint arXiv:1809.04157 (2018)
47. Zhu, C., Cheng, Y., Gan, Z., Sun, S., Goldstein, T., Liu, J.: FreeLB: enhanced adversarial training for natural language understanding. arXiv preprint arXiv:1909.11764 (2019)

Learning Optimal Linear Precoding for Cell-Free Massive MIMO with GNN

Benjamin Parlier[1], Lou Salaün[2](✉), and Hong Yang[3]

[1] Université Paris-Saclay, Centrale-Supélec, 91190 Gif-sur-Yvette, France
[2] Nokia Bell Labs, 91300 Massy, France
lou.salaun@nokia-bell-labs.com
[3] Nokia Bell Labs, Murray Hill, NJ 07974, USA

Abstract. We develop a graph neural network (GNN) to compute, within a time budget of 1 to 2 milliseconds required by practical systems, the optimal linear precoder (OLP) maximizing the minimal downlink user data rate for a Cell-Free Massive MIMO system – a key 6G wireless technology. The state-of-the-art method is a bisection search on second order cone programming feasibility test (B-SOCP) which is a magnitude too slow for practical systems. Our approach relies on representing OLP as a node-level prediction task on a graph. We construct a graph that accurately captures the interdependence relation between access points (APs) and user equipments (UEs), and the permutation equivariance of the Max-Min problem. Our neural network, named OLP-GNN, is trained on data obtained by B-SOCP. We tailor the OLP-GNN size, together with several artful data preprocessing and postprocessing methods to meet the runtime requirement. We show by extensive simulations that it achieves near optimal spectral efficiency in a range of scenarios with different number of APs and UEs, and for both line-of-sight and non-line-of-sight radio propagation environments.

Keywords: Graph neural network · optimal linear precoding · cell-free massive MIMO · max-min SINR

1 Introduction

We employ a graph neural network (GNN) to solve an important problem relating to a key 6G wireless technology – Cell-Free Massive MIMO (CFmMIMO). The concept of CFmMIMO was first introduced in [23] and further analyzed in [11,12]. "MIMO" refers to "Multiple Input Multiple Output" that takes advantage of spatial multiplexing to serve multiple users simultaneously, thereby greatly increases the spectral efficiency in terms of bits per second per Hertz. "Massive" refers to the hundreds of service antennas in the systems. "Cell-Free", in contrast to cellular, refers to a wireless network where a large number of access points (APs) are distributed in a geographic area to jointly serve a collection of users simultaneously. CFmMIMO relies on a well-designed precoder to

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024

A. Bifet et al. (Eds.): ECML PKDD 2024, LNAI 14949, pp. 72–88, 2024.
https://doi.org/10.1007/978-3-031-70378-2_5

beamform ultrahigh data rate to users. We construct a GNN to compute, within a time budget of 1 to 2 milliseconds, the optimal linear precoder (OLP) that maximizes the minimal (Max-Min) downlink user data rate for a CFmMIMO.

Motivations for Max-Min, OLP and time budget are summarized in the following. Max-Min is highly desirable because all wireless systems aim to achieve the highest data rate possible for all users. A key advantage of CFmMIMO is all the users in the system have statistically identical large scale fading profiles. This contrasts with the traditional cellular system in which large scale fading profiles are uneven for users near the base stations and users at the cell edge. Conceptually, achieving equal throughputs for all users is natural for CFmMIMO. OLP is the optimal precoder among all linear precoders. Furthermore, it is effectively the optimal precoder for massive MIMO. By virtue of the law of large numbers, many service antennas effectively orthogonalize the communication channels, thereby making linear precoding substantially optimal [9]. Millisecond scale time budget is critical for real world applications. For typical mobility applications in urban and suburban scenarios [24], a new precoder must be calculated every 1 to 2 milliseconds to adapt to fast changing wireless communication channels.

To the best of our knowledge, this work is the first attempt in searching for a practically feasible means of computing OLP for CFmMIMO. In this paper, we model OLP as a node-level prediction task on a graph. We represent the wireless communication channel between each AP and user equipment (UE) as a graph node and encode its channel coefficient as a node feature. We define two types of edges: an edge of type-UE connects two nodes (i.e., channels) that are interfering, while a type-AP edge indicates that they share the same transmitter.

Our model, named OLP-GNN, takes as input the aforementioned graph and outputs a precoding matrix which is trained to approximate the optimal linear precoder. It is based on the graph transformer architecture [18]. To satisfy the stringent runtime requirement, our GNN model has to be small: 6 hidden layers and about 22.4k trainable parameters. Given this, we design problem-specific data preprocessing and postprocessing methods to improve OLP-GNN's accuracy. The preprocessing step consists in converting the complex-valued channel coefficients into 4 real-valued components which are then used as input of OLP-GNN. The GNN then predicts 6 features for each node that are combined to obtain a complex-valued precoding matrix in the postprocessing. These features have physical and mathematical interpretations, e.g., signal strength, interference power and power budget constraint. The postprocessing step also ensures that each AP's power budget constraint is satisfied.

We show via simulations that our solution can compute substantially optimal precoders within the time budget of 1 to 2 milliseconds for up to 96 APs and 18 UEs. We compare the spectral efficiency of OLP-GNN to two practical precoders, Maximum Ratio Transmission (MR) and Zero Forcing (ZF), highlighting the performance gain achieved by making OLP computable in real-time. We also show that a single trained model generalizes well to various system sizes and scenarios, including line-of-sight (LoS), non-line-of-sight (NLoS), urban and rural radio propagation environments.

2 Related Work

CFmMIMO Precoder Designs: There are many papers on CFmMIMO precoder designs. Unlike OLP, all variants of MR (also known as conjugate beamforming) [8,15,25] and ZF [5,7,11] are sub-optimal. In this paper, we will include MR and ZF as baselines. The scalable CFmMIMO framework is introduced in [2] which requires all processing tasks to have finite complexity as the number of UEs increases. In such a framework, the computations have to be distributed. This is not the case of our approach. Indeed, OLP-GNN is computed in a centralized manner with full knowledge of all channels. [3] proposed a combination of MR and ZF, with most APs doing decentralized MR to minimize the fronthaul burden. [5] developed JointCFNet, a convolutional neural network for joint user association and power control with local partial protective ZF. [10] considered the uplink counterpart of OLP, i.e., the calculation of uplink joint optimal beamforming and power control for CFmMIMO.

Graphs Neural Networks for CFmMIMO: GNNs have been applied to the following optimization problems in CFmMIMO. Reference [15] studied the downlink power control assuming MR precoding, while [17] solved the uplink pilot power control. The authors of [7] tackled the joint downlink and uplink power control in a full-duplex system assuming ZF precoding. SINRnet is proposed in [13] to maximize the downlink energy efficiency with MR precoding in an unsupervised manner. [14] optimizes the AP selection. The graph structures in [4,13,15] are similar to ours where nodes encode channels. However, their node features represent average channel amplitudes (real-valued), while we use instantaneous channel coefficients (complex-valued). The interference graph in [4] can be seen as a special case of our graph where each AP serves exactly one UE, and only UE-type edges are considered. Other papers follow a different but common graph construction for wireless systems where each node is either an AP or a UE [7,14,17].

3 System Model and Notation

3.1 Notation

We write vectors with bold font lowercase letters, e.g., $\mathbf{v}$, and matrices with bold font capital letters, e.g., $\mathbf{A}$. Superscripts ^{T} and * denote respectively the transpose and complex conjugate transpose of a matrix. Thus, ^{T*} and ^{*T} denote the un-transposed conjugate. All vectors are assumed to be column vectors. For $\mathbf{v} \in \mathbb{C}^K$, $\text{diag}(\mathbf{v}) \in \mathbb{C}^{K\times K}$ denotes the diagonal matrix with $\mathbf{v}$ as diagonal values. For $\mathbf{A} \in \mathbb{C}^{K\times K}$, $\text{diag}(\mathbf{A}) \in \mathbb{C}^K$ denotes the diagonal matrix whose diagonal is the diagonal of matrix $\mathbf{A}$. $\|\cdot\|_2$ and $\|\cdot\|_\infty$ denote the 2-norm and infinity norm respectively. For $\mathbf{A} \in \mathbb{C}^{M\times K}$, $\bar{\mathbf{a}}_m$ denotes the m-th row of $\mathbf{A}$. For two matrices $\mathbf{A}$ and $\mathbf{B}$ of compatible sizes, $[\mathbf{A}, \mathbf{B}]$ denotes their concatenation along the second dimension. $\mathbf{I}_K$ is the K-dimensional identity matrix. $\mathbb{R}^+$ denotes the set of all real positive numbers. Let $x \in \mathbb{C}$ be a complex number, we denote its magnitude (absolute value) by $|x|$ and its phase by $\text{phase}(x)$. We define $[K] = \{1, \cdots, K\}$.

3.2 Cell-Free Massive MIMO

We consider a CFmMIMO system where M APs transmit simultaneously to K UEs in the downlink. A fundamental assumption of massive MIMO is that M is greater than K [9]. The channel between any AP $m \in [M]$ and UE $k \in [K]$ is characterized by a complex channel coefficient $g_{m,k} \in \mathbb{C}$. The matrix of all channel coefficients is called *channel matrix* and is denoted by $\mathbf{G} \in \mathbb{C}^{M \times K}$. We have

$$\mathbf{G} = \begin{pmatrix} g_{1,1} & \cdots & g_{1,K} \\ \vdots & \vdots & \vdots \\ g_{M,1} & \cdots & g_{M,K} \end{pmatrix} = \begin{pmatrix} \mathbf{g}_1 \cdots \mathbf{g}_K \end{pmatrix} = \begin{pmatrix} \bar{\mathbf{g}}_1^T \\ \vdots \\ \bar{\mathbf{g}}_M^T \end{pmatrix},$$

where $\mathbf{g}_k \in \mathbb{C}^M$ is the channel vector between the k-th UE and all M APs, and $\bar{\mathbf{g}}_m \in \mathbb{C}^K$ the channel vector between the m-th AP and all K UEs. The APs are connected to a central controller that collects the channel state information (CSI) which gives us $\mathbf{G}$. The computations and neural network inferences presented in this paper are performed on this central controller, then the results are sent to each AP.

Let $\mathbf{x} \in \mathbb{C}^K$ be the signal received by the K users. It can be modelled as:

$$\mathbf{x} = \mathbf{G}^T(\sqrt{\rho_d}\mathbf{s}) + \mathbf{w}, \tag{1}$$

where ρ_d is the downlink signal to noise ratio (SNR) for each AP, $\mathbf{s} \in \mathbb{C}^M$ is the power normalized precoded signal to be transmitted by the M APs and $\mathbf{w} \in \mathbb{C}^K$ is a circularly-symmetric Gaussian noise vector. The APs are subject to a power constraint set to

$$\|\mathbb{E}(\mathbf{s}^{*T} \odot \mathbf{s})\|_\infty \leq 1, \tag{2}$$

with $\mathbb{E}$ the expectation and $\odot$ the element wise multiplication.

3.3 Precoding Matrix and Downlink SINR Calculation

We denote by $\mathbf{q} \in \mathbb{C}^K$ the users' message-bearing symbols to be transmitted. We assume, as in [22], that $\mathbf{q}$ has zero mean, unit variance and that the symbols are uncorrelated between users such that the following holds

$$\mathbb{E}(\mathbf{q}\mathbf{q}^*) = \mathbf{I}_K. \tag{3}$$

As highlighted in Eq. (1), the signal to be transmitted by the APs is $\mathbf{s} \in \mathbb{C}^M$. Therefore $\mathbf{q}$ must be converted from the user data symbols space $\mathbb{C}^K$ to the precoded signals space $\mathbb{C}^M$. This is done with a linear precoding matrix $\varDelta$ as follows

$$\mathbf{s} = \varDelta\mathbf{q}, \quad \text{where} \quad \varDelta = \begin{pmatrix} \delta_{1,1} & \cdots & \delta_{1,K} \\ \vdots & \vdots & \vdots \\ \delta_{M,1} & \cdots & \delta_{M,K} \end{pmatrix} \in \mathbb{C}^{M \times K}. \tag{4}$$

The assumption on $\mathbf{q}$ in Eq. (3) combined with Eqs. (2) and (4) imposes the following power constraints on $\varDelta$

$$\forall\, m \in [M], \quad \|\bar{\boldsymbol{\delta}}_m\|_2 \leq 1, \tag{5}$$

where $\bar{\boldsymbol{\delta}}_m = (\delta_{m,1}, \cdots, \delta_{m,K})^T \in \mathbb{C}^K$.

From Eqs. (1) and (4) we get $\mathbf{x} = \sqrt{\rho_d}\mathbf{G}^T \Delta \mathbf{q} + \mathbf{w}$, which allows us to express the signal received at the k-th user as

$$x_k = \sqrt{\rho_d}\mathbf{g}_k^T \Delta \mathbf{q} + w_k = \sqrt{\rho_d}\mathbf{g}_k^T \boldsymbol{\delta}_k q_k + \sqrt{\rho_d}\sum_{l \neq k} \mathbf{g}_k^T \boldsymbol{\delta}_l q_l + w_k,$$

where $\boldsymbol{\delta}_k = (\delta_{1,k}, \cdots, \delta_{M,k})^T \in \mathbb{C}^M$. We know from Eq. (3) that a signal emitted for a specific user k is uncorrelated with interfering signals intended for other users. Similarly, white additive noise is uncorrelated with both intended and interfering signals. Since the intended signal, interfering signals and noise are mutually uncorrelated, we can calculate their contribution to power separately. Hence, the power of the signal x_k received by the k-th user can be written as $\mathbb{E}(x_k^* x_k) = \rho_d |\mathbf{g}_k^T \boldsymbol{\delta}_k|^2 + \rho_d \sum_{l \neq k} |\mathbf{g}_k^T \boldsymbol{\delta}_l|^2 + 1$, with the following terms:

- Signal power (SP): $\rho_d |\mathbf{g}_k^T \boldsymbol{\delta}_k|^2$ is the power of the signal intended for user k.
- Interference power (IP): $\rho_d \sum_{l \neq k} |\mathbf{g}_k^T \boldsymbol{\delta}_l|^2$.
- Noise power (NP) is equal to 1.

The *signal to interference plus noise ratio* (SINR) of a user k is defined as the ratio between its intended signal power and the interference power plus noise power. Thus, the SINR of user k can be calculated as

$$\text{SINR}_k = \frac{\text{SP}}{\text{IP} + \text{NP}} = \frac{\rho_d |\mathbf{g}_k^T \boldsymbol{\delta}_k|^2}{1 + \rho_d \sum_{l \neq k} |\mathbf{g}_k^T \boldsymbol{\delta}_l|^2}. \tag{6}$$

Equation (6) can be expressed otherwise by introducing the following matrix $\mathbf{A} \in \mathbb{C}^{K \times K}$ which combines both effects of precoding and channel propagation.

$$\mathbf{A} = \begin{pmatrix} a_{1,1} & \cdots & a_{1,K} \\ \vdots & \vdots & \vdots \\ a_{K,1} & \cdots & a_{K,K} \end{pmatrix} = \mathbf{G}^T \Delta. \tag{7}$$

Equation (6) then becomes

$$\text{SINR}_k = \frac{\rho_d |a_{k,k}|^2}{1 + \rho_d \sum_{l \neq k} |a_{k,l}|^2}. \tag{8}$$

3.4 Optimal Linear Precoding

We define the optimal linear precoding, denoted by Δ_{OLP}, as the solution to the following max-min SINR problem.

$$\begin{aligned} &\max_{\Delta} \quad \min_{k} \quad \text{SINR}_k, \\ &\text{subject to} \quad \|\bar{\boldsymbol{\delta}}_m\|_2 \leq 1, \quad \forall m. \end{aligned} \tag{$\mathcal{P}$}$$

The objective of this problem is to maximize the minimum SINR among all UEs while satisfying the power constraint (5). Note that the objective is a function of

$\varDelta$ as shown in (4) and (6). We will explain in the following paragraphs how the optimal $\varDelta_{OLP}$ can be obtained by a combination of bisection search and second order cone programming (SOCP) feasibility search.

We first focus on the following feasibility subproblem. Given a threshold value t_{SINR}, it consists in checking whether there exists a feasible solution such that $\min_k \mathrm{SINR}_k \geq t_{SINR}$. This inequality can be expanded with Eq. (8) as

$$\forall k \in [K], \quad \frac{\rho_d |a_{k,k}|^2}{1 + \rho_d \sum_{l \neq k} |a_{k,l}|^2} \geq t_{SINR},$$

which is equivalent to

$$|a_{k,k}|^2 \geq t_{SINR} \left(\frac{1}{\rho_d} + \sum_{l \neq k} |a_{k,l}|^2 \right). \tag{9}$$

By introducing the following matrix $\tilde{\mathbf{A}} = \left[\mathbf{A} - \mathrm{diag}(a_{1,1}, \cdots, a_{K,K}); \frac{1}{\sqrt{\rho_d}} \mathbf{1}_{K \times 1} \right]$ in $\mathbb{C}^{K \times (K+1)}$, inequality (9) can be simplified as

$$\forall k \in [K], \quad |a_{k,k}| \geq \sqrt{t_{SINR}} \|\tilde{\mathbf{a}}_k\|_2, \tag{10}$$

where $\tilde{\mathbf{a}}_k = (\tilde{a}_{k,1}, \cdots, \tilde{a}_{k,K+1}) \in \mathbb{C}^{K+1}$. We note that the left term in Eq. (10) is convex due to the absolute value $|\cdot|$. It has to be concave to match a standard form of SOCP constraint. To this end, we will restrict the set of possible values for $a_{k,k}$.

We derive from Eq. (7) that $\varDelta$ can be written as a function of $\mathbf{A} \in \mathbb{C}^{K \times K}$ and an arbitrary matrix $\mathbf{U} \in \mathbb{C}^{M \times K}$ such that

$$\varDelta = \mathbf{G}^{\dagger} \mathbf{A} + P_{\mathbf{G}^T} \mathbf{U}, \tag{11}$$

where $\mathbf{G}^{\dagger} = \mathbf{G}^{T*} \left(\mathbf{G}^T \mathbf{G}^{T*} \right)^{-1}$ is the Moore-Penrose pseudo-inverse of $\mathbf{G}^T$, and $P_{\mathbf{G}^T} = \mathbf{I}_M - \mathbf{G}^{\dagger} \mathbf{G}^T$ is the orthogonal projection onto the null space of $\mathbf{G}^T$.

We can see that the right multiplication of $\varDelta$ by a matrix $\mathrm{diag}(e^{i\theta_1}, \cdots, e^{i\theta_K})$ does not change its 2-norm, thus leaving the right hand side of (10) unchanged. Besides, if $(\mathbf{A}, \mathbf{U})$ satisfies inequality (10) and the power constraint (5), then so does $\left(\mathbf{A}\mathrm{diag}(e^{i\theta_1}, \cdots, e^{i\theta_K}), \mathbf{U}\mathrm{diag}(e^{i\theta_1}, \cdots, e^{i\theta_K}) \right)$. Therefore, if we multiply $\mathbf{A}$ and $\mathbf{U}$ by $\mathrm{diag}(e^{i\theta_1}, \cdots, e^{i\theta_K}) = \mathrm{diag}(-\mathrm{phase}(\mathbf{A}))$, we can restrict our search to positive real values of $a_{k,k}$, for all k, instead of complex values. With this assumption, the max-min problem ($\mathcal{P}$) can be reformulated as:

$$\begin{array}{lll} \max\limits_{\varDelta} & t_{SINR}, & \\ \text{subject to} & a_{k,k} \geq \sqrt{t_{SINR}} \|\tilde{\mathbf{a}}_k\|_2, & \forall k, \\ & a_{k,k} \in \mathbb{R}^+, & \forall k, \\ & \|\bar{\boldsymbol{\delta}}_m\|_2 \leq 1, & \forall m. \end{array} \tag{$\mathcal{P}'$}$$

The constraints of $\mathcal{P}'$ are written in a standard form suitable for SOCP. Hence, for any value of t_{SINR}, SOCP can be applied to check the feasibility of the constraints. A bisection search can be used on top of SOCP to find the maximum value of t_{SINR}. We will refer to this method as B-SOCP throughout the paper.

3.5 Zero Forcing and Maximum Ratio Precoding

Other linear precoding schemes are often considered, which are by definition sub-optimal compared to OLP but less costly to solve. In this paper, we will compare our solution to two precoding schemes, namely Zero Forcing (ZF) and Maximum Ratio Transmission (MR). ZF minimizes the interference while MR maximizes the signal. The optimal is a trade-off between these two extremes. The ZF precoder has a closed-form expression which can be computed by matrix multiplications and inversions [11]. For MR, the max-min objective can be achieved by solving a B-SOCP problem similar to $\mathcal{P}'$ [12].

It is known that for some regimes (e.g., high signal-to-noise, large number of APs), ZF outperforms MR whereas for other regimes it is the opposite. This can be seen in Sect. 5 for example, where MR achieves higher spectral efficiency than ZF in Fig. 3d and ZF beats MR in Fig. 3c. In all scenarios, OLP significantly outperforms both MR and ZF. However, the computational complexity of B-SOCP makes it unsuitable for real world systems with millisecond-scale runtime requirements. This shows the importance of developing an approximation of OLP with several order of magnitudes faster runtimes.

4 Graph Neural Network

In this section, we describe our solution, named OLP-GNN, to tackle the max-min SINR problem ($\mathcal{P}$). OLP-GNN is trained with OLP data obtained by running B-SOCP in a simulated environment. The objective is to approximate the performance of OLP with a low and practical computational complexity.

4.1 Graph Representation

The input and output of our max-min problem ($\mathcal{P}$) are respectively the channel matrix $\mathbf{G}$ and the precoding matrix Δ_{OLP}. One can see that for any permutation applied to the rows and/or columns of $\mathbf{G}$, the same permutation is applied to the optimal solution Δ_{OLP}. Thus the problem is independent from the row and column indexing. This property is called *permutation equivariance* and GNNs are known to satisfy this property, which make them suitable for our problem.

To train a GNN the input channel matrix $\mathbf{G}$ and output precoding matrix Δ_{OLP} must be represented as graphs. We define a directed graph as (V, E) where V is the set of nodes and E the set of directed edges. We define a node as a pair $(m, k) \in [M] \times [K]$, thus V has $M \times K$ nodes. We also define π, a bijection from the set of pairs (m, k) to the set of nodes $V = [MK]$, that associates to each (m, k) pair a node index $i \in V$, such that $\pi(m, k) = i$.

In our problem when two UEs share the same AP, i.e., they are in the same row of Δ_{OLP}, they mutually influence each other through the power constraint of Eq. (5). Similarly, when two APs serve the same UE k, i.e., they are on the same column of Δ_{OLP}, they both have an impact on the calculation of SINR_k in Eq. (6). To encode these properties in our graph, we consider two types of edges. We set an edge between $i, j \in V$ if and only if they share a common AP or a common UE. We denote this edge by $e = (i, j)$ and its type by $\text{type}(e) \in \{\text{AP}, \text{UE}\}$ depending on whether i and j share a common AP or UE. The graph does not have self loop, i.e., $\forall i \in V, (i, i) \notin E$. This formally translates to for all $m, m' \in [M]$, $m \neq m'$, and for all $k, k' \in [K]$, $k \neq k'$, we have:

i. $(\pi(m, k), \pi(m, k')) \in E$ is of type AP,
ii. $(\pi(m, k), \pi(m', k)) \in E$ is of type UE.

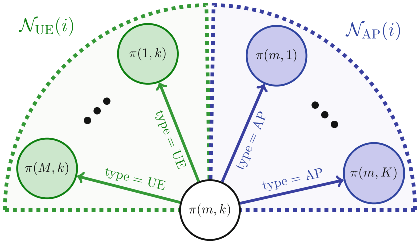


Fig. 1. Neighbors and outgoing edges of a typical node $\pi(m, k) = i \in V$

The heterogeneous nature of E allows a GNN to process differently the information on a node and its neighbors based on their edge type. Thus, for each node $i \in V$, we define two disjoint sets of neighbors depending on their edge type: $\mathcal{N}_{AP}(i) = \{j \in V \,|\, (i, j) \in E \text{ and } \text{type}((i, j)) = \text{AP}\}$ and $\mathcal{N}_{UE}(i) = \{j \in V \,|\, (i, j) \in E \text{ and } \text{type}((i, j)) = \text{UE}\}$. Figure 1 illustrates the neighbors of a typical node. We note that for each node $i \in V$, $\mathcal{N}_{AP}(i)$ has $K - 1$ elements and $\mathcal{N}_{UE}(i)$ has $M - 1$ elements, for a total of $M + K - 2$ neighbors. In other words, each node has $M + K - 2$ outgoing edges, and the same number of incoming edges. Since the graph contains MK nodes, the total number of edges is $MK(M + K - 2)$.

4.2 Data Preprocessing and Postprocessing

In this subsection, we consider a typical node $i \in V$ corresponding to the channel between AP m and UE k, i.e., $\pi(m, k) = i$. At each iteration t of the GNN, node i is associated with a tensor $h_i(t)$, called *node feature*. $h_i(0)$ is its input feature and $h_i(T)$ is its output feature, where T is the number of iterations/layers. We assume that all tensors in OLP-GNN are real-valued. Since the input and target data are complex matrices, we decompose them into their magnitude and phase components. Another possibility is to decompose them into their real and imaginary parts. However, we found magnitude-phase representation to be more fitting, probably due to its much smaller ranges.

We see from Eq. (11) that the precoding matrix depends directly on $\mathbf{G}^\dagger$. During the development of OLP-GNN, we observed that learning the pseudo-inverse $\mathbf{G}^\dagger$ gave an unsatisfactory accuracy and a poor generalization to different number of APs and UEs. Indeed, the performance drops by $5 - 20\%$ on the validation datasets without $\mathbf{G}^\dagger$ in the input of our $T = 6$ layers model. Thus, we decided to compute $\mathbf{G}^\dagger$ beforehand using fast numerical methods and include it in OLP-GNN's input. Therefore, $h_i(0)$ contains 4 elements which are the magnitudes and phases of $\mathbf{g}_{m,k}$ and $\mathbf{g}^\dagger_{m,k} = (\mathbf{G}^\dagger)_{m,k}$, i.e., $h_i(0) = \Big(|\mathbf{g}_{m,k}|,\ \text{phase}(\mathbf{g}_{m,k}),\ |\mathbf{g}^\dagger_{m,k}|,\ \text{phase}(\mathbf{g}^\dagger_{m,k})\Big)$.

We split the target precoding matrix Δ_{OLP} in three components $\mathbf{G}^\dagger \text{diag}(\mathbf{A})$, $\mathbf{G}^\dagger(\mathbf{A}-\text{diag}(\mathbf{A}))$, $P_{\mathbf{G}^T}\mathbf{U}$. Therefore the output and target feature tensor contains the following 6 terms: magnitude and phase of $(\mathbf{G}^\dagger \text{diag}(\mathbf{A}))_{m,k}$, magnitude and phase of $(\mathbf{G}^\dagger(\mathbf{A} - \text{diag}(\mathbf{A}))_{m,k}$, magnitude and phase of $(P_{\mathbf{G}^T}\mathbf{U})_{m,k}$. This split is motivated by the distinct physical meanings of these terms. The diagonal elements $(a_{k,k})_{k\in[K]}$ of $\mathbf{A}$ represent user k's useful signal. The non-diagonal elements $(a_{l,k})_{k\neq l}$ correspond to the interfering signals intended for user k but received by another user l. In this sense, $\mathbf{A}$ fully characterizes the SINR. On the other hand, $P_{\mathbf{G}^T}\mathbf{U}$ only influences the power constraint without changing the SINRs.

For OLP-GNN to extract useful information from the features, they must be of the same order of magnitude. The absolute values of the input, output and target features range over several orders of magnitude. Therefore, we apply a $\log_2$ transformation to all absolute values (magnitude terms). As an example, we typically have $10^{-15} \leq |g_{m,k}| \leq 10^{-5}$, hence $\log_2(|g_{m,k}|)$ belongs to $[-50, -16]$. We do not apply a $\log_2$ transformation to the phase terms as they are already in a small range between 0 and 2π. All the features are then normalized to have zero mean and unitary variance.

Let $\mathbf{y}_1$, $\mathbf{y}_2$, $\mathbf{y}_3$ be the OLP-GNN predictions of the aforementioned three terms: $\mathbf{G}^\dagger \text{diag}(\mathbf{A})$, $\mathbf{G}^\dagger(\mathbf{A}-\text{diag}(\mathbf{A}))$, $P_{\mathbf{G}^T}\mathbf{U}$. They are obtained by de-processing the output tensors $h_i(T)$ for all nodes $i \in V$. We apply the following postprocessing to impose some desired properties on the output

$$\begin{cases} \mathbf{y}_1' = \mathbf{G}^\dagger \text{real}(\text{diag}(\mathbf{G}^T\mathbf{y}_1)), \\ \mathbf{y}_2' = \mathbf{G}^\dagger(\mathbf{G}^T\mathbf{y}_2 - \text{diag}(\mathbf{G}^T\mathbf{y}_2)), \\ \Delta = \mathbf{y}_1' + \mathbf{y}_2' + \mathbf{y}_3. \end{cases}$$

As we have $\mathbf{I}_K = \mathbf{G}^T\mathbf{G}^\dagger$, the postprocessing on $\mathbf{y}_1'$ ensures that $\mathbf{G}^T\mathbf{y}_1'$ is a real diagonal matrix. The postprocessing on $\mathbf{y}_2'$ enforces that the diagonal elements of $\mathbf{G}^T\mathbf{y}_2'$ are equal to zero. Once this is applied on the output features, we further impose the power constraint (5) by applying the following projection

$$\forall m \in [M], \text{ if } \|\bar{\boldsymbol{\delta}}_m\|_2 \geq 1 \text{ then } \bar{\boldsymbol{\delta}}_m \leftarrow \frac{\bar{\boldsymbol{\delta}}_m}{\|\bar{\boldsymbol{\delta}}_m\|_2}.$$

Δ obtained by the sum of components $\mathbf{y}_1'$, $\mathbf{y}_2'$, $\mathbf{y}_3$, and after the above projection is applied is the predicted precoding matrix of OLP-GNN.

4.3 Structure of the Neural Network

Let $\mathcal{L}$ be the linear operator. For the sake of clarity, the linear layers in this section will be written with different subscripts and superscripts, e.g., $\mathcal{L}^1_{AP,t}$, $\mathcal{L}^4_{UE,t}$. This is done to indicate that they are applied on different edge types (AP or UE), at different iterations $t \in \{0, \ldots, T\}$ and they do not share any trainable parameter.

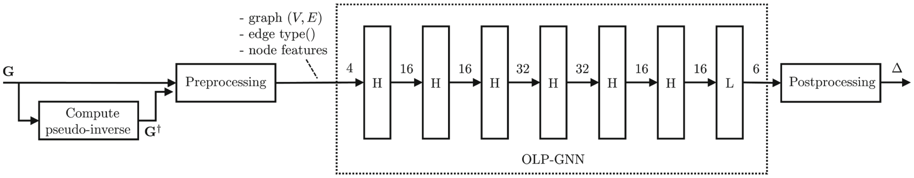


Fig. 2. Structure of OLP-GNN. 'H' refers to the hidden attention layer, and 'L' is the final linear layer. The number between each layer represents the node feature size.

For each node $i \in V$, its feature h_i is updated based on itself and its direct neighbors at the previous step. Thus, $h_i(t+1)$ is a function of $h_i(t)$ and $h_j(t)$, $\forall j \in \mathcal{N}_{AP}(i) \cup \mathcal{N}_{UE}(i)$, updated according to the following rule

$$h_i(t+1) = \text{Norm}\left(\text{ReLU}(f_{AP,t}(i) + f_{UE,t}(i))\right), \tag{12}$$

where Norm denotes the layer normalization and ReLU the rectified linear unit activation function. For $i, j \in V$, $\bullet \in \{\text{AP}, \text{UE}\}$ and $t \in \{0, \dots, T-1\}$, function $f_{\bullet,t}$ implements the graph transformer of [18] with a single attention head. It is defined as

$$f_{\bullet,t}(i) = \mathcal{L}^1_{\bullet,t}(h_i(t)) + \sum_{j \in \mathcal{N}_\bullet(i)} \alpha_{\bullet,t}(i,j) \times \mathcal{L}^2_{\bullet,t}(h_j(t)). \tag{13}$$

The attention coefficient $\alpha_{\bullet,t}(i,j)$ is equal to $\langle \mathcal{L}^3_{\bullet,t}(h_i(t)), \mathcal{L}^4_{\bullet,t}(h_j(t)) \rangle$ divided by $\sum_{u \in \mathcal{N}_\bullet(i)} \langle \mathcal{L}^3_{\bullet,t}(h_i(t)), \mathcal{L}^4_{\bullet,t}(h_u(t)) \rangle$, where $\langle x, y \rangle = \exp\left(\frac{x^T y}{\sqrt{d}}\right)$ is the exponential scalar product [20] and d the size of tensors x and y.

In this context, the attention is an efficient mechanism to select which neighbors have the most impact on improving the node's OLP prediction task according to their channels. It is important to note that a permutation of the nodes indices does not change the output value due to the summation used as an aggregator in Eq. (13). As a consequence, the update rule (12) satisfies the permutation equivariance property of our problem.

Figure 2 shows the structure of our solution. OLP-GNN has $T = 6$ hidden attention layers implementing the update rule (12). We choose this value since increasing the number of layers to 7 does not improve the average performance by more than 1% while increasing the runtime by $5 - 15\%$. The final iteration is a simple linear layer. OLP-GNN has a total of 22.4 k trainable parameters.

4.4 Training and Loss Function

For training OLP-GNN, we generate data from two environments: free space 60GHz LoS and urban 2 GHz NLoS. Each environment is simulated with the following number of APs and UEs placed randomly in a circular area of 500 m radius: $(M, K) = (32, 6)$, $(32, 9)$, $(64, 9)$, $(64, 18)$. For each channel matrix $\mathbf{G}$, we compute the corresponding target Δ_{OLP} using B-SOCP. Each of the above 8 datasets has 10k samples, for which 9k are used for training, 500 for validation and 500 for testing. In summary, there is a total of 72 k training samples. Details about the simulation settings can be found in the next section. Moreover, additional test datasets will be introduced there to evaluate the generalizability of OLP-GNN.

We consider the mean square error loss of the per-user SINR to train our model, i.e., $\sum_{k=1}^{K}(\mathrm{SINR}_k^* - \mathrm{SINR}_k)^2/K$, where SINR_k^* is the target SINR value and SINR_k is the SINR value predicted by OLP-GNN for user k. The SINRs used in the loss are expressed in dB. We use the Adam optimizer [6] for the training with a learning rate of 7×10^{-4}, a batch size of 16, and 1000 epochs.

5 Numerical Results

In this section, we describe in detail the simulation used to generate training, validation and test data for OLP-GNN. We then analyze our solution in terms of spectral efficiency, computational complexity and runtime. We also compare OLP-GNN to the classical MR and ZF precodings, as well as the target OLP.

5.1 Simulation Parameters and Performance Metrics

We simulate three CFmMIMO environments, namely 60 GHz LoS, urban 2 GHz NLoS and rural 450 MHz NLoS. The LoS model is identical to the one in [22] for 60 GHz carrier frequency. The NLoS environments are modeled following the ITU-R [16] recommendations. Specifically, we consider the urban macro and rural macro NLoS radio propagation models with respectively 2 GHz and 450 MHz carrier frequencies. We deliberately choose different carrier frequencies to show the generalizability of our solution. We consider a bandwidth of 20 MHz for all three environments.

For each environment we define 24 scenarios with different number of APs ($M = 24 \ldots 96$) and UEs ($K = 4 \ldots 36$) as summarized in Table 1. The APs and UEs are randomly positioned inside a circular area of radius 500 m (4 km for the rural environment). As explained in Sect. 4.4, four LoS datasets and four urban NLoS datasets are used for training with 9k training samples each. The rest of the LoS and urban NLoS scenarios each has 500 validation samples for hyperparameter tuning and 500 test samples. Finally, the rural NLoS scenarios are dedicated exclusively for testing with 500 samples each.

In contrast to [11,15,25] which only consider large-scale fading to optimize the power control, precoding is done at a much shorter time scale which requires to account for fast fading. Indeed, the channel coefficient $g_{m,k}$ is equal to the large-scale fading between the m-th AP and k-th UE multiplied by a fast fading term $\zeta_{m,k}$. Here, we assume the fast fading to be i.i.d. Rayleigh distributed, i.e., $\zeta_{m,k} = (x_1 + x_2 \mathrm{i}) / \sqrt{2}$, where x_1 and x_2 are independent standard normal random variables and i denotes the imaginary unit. The magnitude of the complex random variable $\zeta_{m,k}$ follows Rayleigh distribution.

Table 1. OLP-GNN performance and runtimes.

Graph size		Spectral efficiency loss compared to the optimal (%)						Runtime (ms)	
M	K	LoS		Urban NLoS		Rural NLoS		average	std
		median	95%-likely	median	95%-likely	median	95%-likely		($\times 10^{-3}$)
24	4	0.92	1.26	0.83	1.87	0.79	2.99	0.94	7.25
24	5	0.74	1.07	0.72	1.46	1.22	3.40	0.94	7.10
24	6	0.71	1.34	0.74	3.49	1.31	2.77	0.95	7.75
24	9	0.86	1.96	1.20	5.59	2.67	7.76	0.95	8.18
32	4	1.33	1.65	0.32	0.16	0.29	−0.05	0.95	7.82
32	6	0.50	0.69	0.27	0.72	0.76	0.75	0.96	7.85
32	8	0.50	0.80	0.44	0.62	0.74	1.10	0.96	7.86
32	9	0.64	0.65	0.55	1.21	0.06	2.69	0.96	7.84
32	12	0.55	0.88	0.65	1.18	1.18	4.99	0.96	7.55
32	16	1.17	3.37	0.35	4.06	2.21	11.57	0.97	8.29
48	8	0.62	0.76	0.24	0.64	0.09	0.33	0.99	7.59
48	12	0.56	0.69	0.56	0.72	0.47	1.07	1.02	8.70
48	16	0.61	0.69	0.48	0.09	0.12	1.43	1.06	7.65
48	24	1.31	2.94	0.07	2.25	0.40	2.56	1.24	8.08
64	6	0.94	1.28	0.59	0.64	0.46	0.46	0.99	7.87
64	9	0.47	0.83	0.29	0.90	0.44	0.04	1.04	6.99
64	12	0.56	0.78	0.51	0.71	0.29	0.15	1.09	7.73
64	18	0.58	0.69	0.54	0.50	0.24	0.32	1.28	8.09
64	24	0.44	0.59	0.37	0.63	−0.18	−0.92	1.41	7.96
64	32	1.70	3.50	−0.06	1.73	−0.49	0.82	1.55	8.54
96	9	0.79	1.19	0.46	0.42	0.37	0.47	1.19	7.45
96	18	0.69	0.68	0.48	0.55	0.41	0.66	1.59	8.33
96	27	0.72	0.92	0.57	1.04	0.26	−0.04	2.02	8.64
96	36	1.30	2.52	0.41	0.95	0.05	−0.13	2.53	9.67

We define the *spectral efficiency* (SE) in bit/s/Hz of user k as $\mathrm{SE} = \log_2(1 + \mathrm{SINR}_k)$. We introduce two metrics based on SE to study the performance of OLP-GNN, the *performance loss at median* and the *95%-likely SE loss*. The performance loss at median refers to the relative difference in spectral efficiency between our solution and OLP taken at the median of their cumulative distribution functions (CDF). The 95%-likely performance metric is the relative loss at the 5-th percentile, thus indicating the coverage quality for 95% of users.

The performances of our solution in terms of spectral efficiency, runtime and complexity will be compared to the optimal B-SOCP solution which defines the upper bound for spectral efficiency but achieved in a time consuming manner.

The simulations and algorithms are implemented in Python 3 and can be found at https://github.com/Nokia-Bell-Labs/olp-gnn. The optimal linear precoding matrix is obtained by solving problem $\mathcal{P}'$ using the MOSEK solver [1] for SOCP combined with a bisection search as explained in Sect. 3.4. The bisection search terminates when precision $\epsilon = 0.01$ is reached for all SINRs. The OLP-GNN is implemented in PyTorch 2 and is compiled with PyTorch's default backend TorchInductor for runtime measurements.

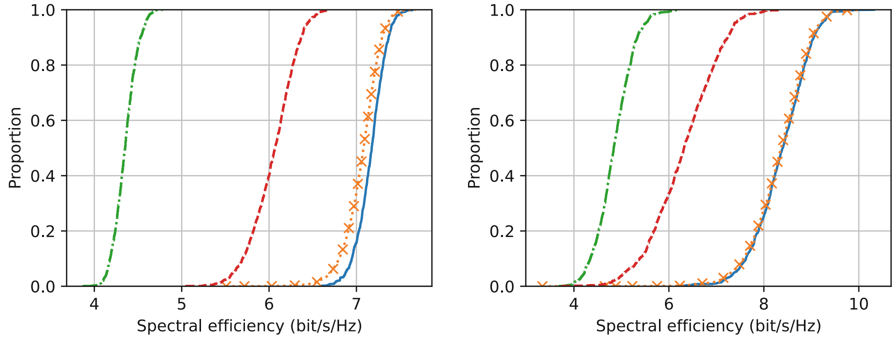
5.2 Spectral Efficiency

To highlight the generalizability of OLP-GNN, we test it on a rural NLoS environment. This is in contrast to the LoS and urban NLoS environments seen during training. The rural channel distribution differs from the training datasets due to the different radio propagation models, cell sizes and carrier frequencies employed. The results are summarized in Table 1. Note that, the median or 95%-likely SE loss written in the table can be negative since the precoder produced by OLP-GNN may give higher SE to some users at the expense of other users. In this case, the max-min objective is not reached and OLP-GNN necessarily under-performs at other parts of the CDF. For example, in the rural NLoS scenario with 64 APs and 24 UEs, OLP-GNN achieves negative median and 5-th percentile losses. Nevertheless, it is 0.20% away from optimal at the 1st percentile of the CDF (not shown on the table).

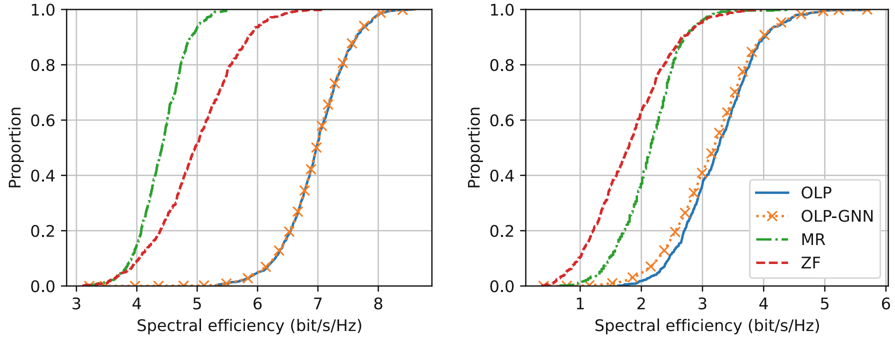
Figure 3 shows the spectral efficiency of MR, ZF, OLP and OLP-GNN on four different scenarios. We first present the performance on scenarios with 96 APs and 36 UEs. These scenarios are relevant to evaluate the generalization of our solution since their graphs are bigger than the ones used for training, which have at most 64 APs and 18 UEs. For the LoS environment in Fig. 3a, OLP-GNN approximates the optimal with 1.30% loss at median and a 95%-likely SE loss of 2.52%. Moreover, it outperforms MR and ZF precodings by respectively 63% and 17% at median. We obtain similar results for the urban NLoS and rural NLoS scenarios in Fig. 3b and 3c. In both cases, OLP-GNN has less than 0.5% loss at median and 1% loss at the 5-th percentile. It significantly outperforms the baseline ZF by 32% in the urban scenario and by 40% in the rural scenario.

In Table 1, we observe 95%-likely SE losses of at most 4% for LoS datasets, 6% for urban NLoS and 12% for rural NLoS scenarios. Due to generalization error, the bottom 5-th percentile is degraded in rural NLoS scenarios compared to their urban NLoS and LoS counterparts. Nonetheless, the median performance loss remains lower than 3% in all environments. Furthermore, even in the worst case scenario shown in Fig. 3d, our solution improves both median SE and 95%-likely SE over MR by around 50%. It also outperforms ZF by respectively 78% and 162%, at median and 5-th percentile respectively.

In NLoS scenarios with 24 or 32 APs, we see the performance degrading when the number of users K increases. This is partly due to the difficulty of approximating matrix $\mathbf{A}$ when the problem has lesser degrees of freedom and the UEs suffer greater interference. In this case, the diagonal elements of $\mathbf{A}$ vary considerably, with some values being a magnitude of order higher than the others. Thus, the model trained in this study is best suited for "massive MIMO" systems where there are sufficiently more transmitting antennas M than receivers K.



(a) LoS: 96 APs, 36 UEs (b) Urban NLoS: 96 APs, 36 UEs



(c) Rural NLoS: 96 APs, 36 UEs (d) Rural NLoS: 32 APs, 16 UEs

Fig. 3. Cumulative distribution functions of the downlink SE for MR, ZF, OLP and OLP-GNN for different environments and graph sizes.

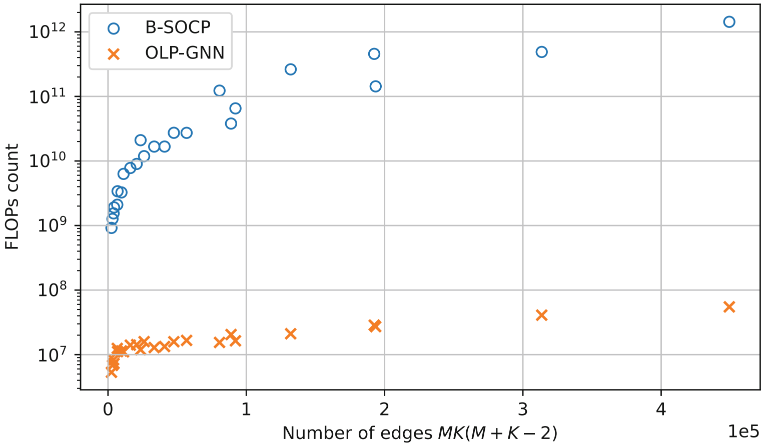


Fig. 4. Number of FLOPs versus the number of edges for B-SOCP and OLP-GNN. Each point represents one rural NLoS scenario.

5.3 Computational Complexity and Runtime

The asymptotic time complexity of OLP-GNN is $O(MK(M+K))$. This can be derived by noting that the update rule (13) aggregates neighboring features with a complexity proportional to the number of edges. The SOCP solver in MOSEK is based on primal-dual interior point method which has asymptotic complexity $O\big(n^{3.5}\log(\xi^{-1})\big)$ [21], where $n = M + K$ in our problem and ξ is the duality gap at termination. By applying bisection search on top of SOCP with a precision ϵ, we deduce that B-SOCP runs in $O\big((M+K)^{3.5}\log(\xi^{-1})\log(\epsilon^{-1})\big)$. In comparison, OLP-GNN has lower asymptotic complexity and does not have an iteration-complexity depending on a precision hyperparameter, ξ or ϵ.

In practice, we evaluate the complexity of our algorithms with the number of floating point operations (FLOPs). The FLOPs of our solution and the B-SOCP method are given in Fig. 4. These FLOPs are obtained with PyPAPI (a tool to access low-level hardware performance counters) on an Intel Core i9-10980XE CPU. We see that B-SOCP requires respectively 5.7×10^2, 4.0×10^3 and 1.2×10^4 times more FLOPs than OLP-GNN for $(M, K) = (32, 9)$, $(64, 18)$ and $(96, 27)$. As a consequence, OLP-GNN is several order of magnitude faster than B-SOCP.

We also measure the runtimes of OLP-GNN on a NVIDIA RTX A4000 GPU. Each dataset is repeated 10 times to obtain the runtime statistics in Table 1. These runtimes take into account preprocessing ($0.31 - 0.43$ ms), OLP-GNN inference and postprocessing ($0.17 - 0.18$ ms). For 24 and 32 APs, and up to 16 UEs, the average runtimes are under 1 ms. In the larger scenarios with up to 96 APs and 18 UEs, the runtimes are under 2 ms. In all cases, the standard deviations (std) are lower than 0.01 ms, which indicates that these runtimes are consistently within the 1 to 2 millisecond requirement stated in the introduction. This shows that OLP-GNN is implementable in practice with some limitations on the system size. Moreover, dedicated hardware and code optimization could further reduce the runtimes.

6 Conclusions

In this paper, we apply a graph neural network to the downlink max-min precoding problem in CFmMIMO. Our solution, named OLP-GNN, approximates the optimal linear precoder with several orders of magnitude faster runtimes than the state-of-the-art, making it feasible for real deployment for the first time. Indeed, the runtimes remain under 1ms for up to 32 APs and 16 UEs, and under 2 ms for up to 96 APs and 18 UEs.

The characteristics of communication channels between transmitters and receivers can vary greatly. We evaluate our trained model on both LoS/NLoS and urban/rural use-cases, demonstrating its generalizability to different environments and system sizes. Simulations show that the median spectral efficiencies achieved by OLP-GNN are less than 3% away from optimal on all scenarios.

Reducing further the time complexity of our solution would enable its execution on less powerful and costly hardware. In the current work, OLP-GNN takes as input $\mathbf{G}^{\dagger}$ which must be computed beforehand. Computing such a pseudo-inverse using classical numerical methods causes some overhead on the preprocessing time. It is therefore desirable to develop an end-to-end GNN without the above overheads. We note that one way to speed up the inference time may be to apply GNN-specific quantization methods [19].

References

1. Andersen, E.D., Roos, C., Terlaky, T.: On implementing a primal-dual interior-point method for conic quadratic optimization. Math. Program. **95**(2), 249–277 (2003)
2. Björnson, E., Sanguinetti, L.: Scalable cell-free massive MIMO systems. IEEE Trans. Commun. **68**(7), 4247–4261 (2020)
3. Du, L., Li, L., Ngo, H.Q., Mai, T.C., Matthaiou, M.: Cell-free massive MIMO: joint maximum-ratio and zero-forcing precoder with power control. IEEE Trans. Commun. **69**(6), 3741–3756 (2021)
4. Eisen, M., Ribeiro, A.: Optimal wireless resource allocation with random edge graph neural networks. IEEE Trans. Signal Process. **68**, 2977–2991 (2020)
5. Hao, C., Vu, T.T., Ngo, H.Q., Dao, M.N., Dang, X., Wang, C., Matthaiou, M.: Joint user association and power control for cell-free massive MIMO. IEEE Internet Things J. **11**(9), 15823–15841 (2024)
6. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
7. Li, B., Yang, L.L., Maunder, R.G., Sun, S., Xiao, P.: Heterogeneous graph neural network for power allocation in multicarrier-division duplex cell-free massive MIMO systems. IEEE Trans. Wireless Commun. **22**(5), 3554–3569 (2023)
8. Luo, L., Zhang, J., Chen, S., Zhang, X., Ai, B., Ng, D.W.K.: Downlink power control for cell-free massive MIMO with deep reinforcement learning. IEEE Trans. Veh. Technol. **71**(6), 6772–6777 (2022)
9. Marzetta, T.L., Larsson, E.G., Yang, H., Ngo, H.Q.: Fundamentals of Massive MIMO. Cambridge University Press (2016)
10. Miretti, L., Cavalcante, R.L.G., Stańczak, S.: Joint optimal beamforming and power control in cell-free massive MIMO. In: IEEE Globecom (2022)
11. Nayebi, E., Ashikhmin, A., Marzetta, T.L., Yang, H., Rao, B.D.: Precoding and power optimization in cell-free massive MIMO systems. IEEE Trans. Wireless Commun. **16**(7), 4445–4459 (2017)
12. Ngo, H.Q., Ashikhmin, A., Yang, H., Larsson, E.G., Marzetta, T.L.: Cell-free massive MIMO versus small cells. IEEE Trans. Wireless Commun. **16**(3), 1834–1850 (2017)
13. Raghunath, R., Peng, B., Jorswieck, E.A.: Energy-efficient power allocation in cell-free massive MIMO via graph neural networks. arXiv preprint arXiv:2401.14281 (2024)
14. Ranasinghe, V., Rajatheva, N., Latva-aho, M.: Graph neural network based access point selection for cell-free massive MIMO Systems. In: IEEE Globecom (2021)
15. Salaün, L., Yang, H., Mishra, S., Chen, C.S.: A GNN approach for cell-free massive MIMO. In: IEEE Globecom (2022)
16. Series, M.: Guidelines for evaluation of radio interface technologies for IMT-Advanced. Report ITU M.2135-1 (2009)

17. Shen, Y., Zhang, J., Song, S., Letaief, K.B.: Graph neural networks for wireless communications: from theory to practice. IEEE Trans. Wireless Commun. **22**(5), 3554–3569 (2022)
18. Shi, Y., Huang, Z., Feng, S., Zhong, H., Wang, W., Sun, Y.: Masked label prediction: unified message passing model for semi-supervised classification. arXiv preprint arXiv:2009.03509 (2020)
19. Tailor, S.A., Fernandez-Marques, J., Lane, N.D.: Degree-quant: quantization-aware training for graph neural networks. arXiv preprint arXiv:2008.05000 (2020)
20. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, vol. 30 (2017)
21. Wright, S.J.: Primal-dual Interior-point Methods. SIAM (1997)
22. Yang, H.: Cell-free massive MIMO in LoS. In: IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), pp. 672–677 (2021)
23. Yang, H., Marzetta, T.L.: Capacity performance of multicell large-scale antenna systems. In: 2013 51st Annual Allerton Conference on Communication, Control, and Computing (Allerton), pp. 668–675 (2013)
24. Yang, H., Marzetta, T.L.: Total energy efficiency of cellular large scale antenna system multiple access mobile networks. In: IEEE OnlineGreenComm (2013)
25. Zhao, Y., Niemegeers, I.G., Heemstra de Groot, S.: Power allocation in Cell-free massive MIMO: a deep learning method. IEEE Access **8**(5), 87185–87200 (2020)

Multiple Hypergraph Learning for Ephemeral Group Recommendation

Rui Zhao[1,2], Beihong Jin[1,2(✉)], Yimin Lv[1,2], Yiyuan Zheng[1,2],
and Weijiang Lai[1,2]

[1] Key Laboratory of System Software (Chinese Academy of Sciences) and State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing, China
[2] University of Chinese Academy of Sciences, Beijing, China
`Beihong@iscas.ac.cn`

Abstract. Ephemeral Group Recommendation (EGR) refers to recommending items for a temporarily existing group, where the ephemeral group has little or no historical interactions with items while each group member has his/her own interaction history. We note that EGR not only faces the challenge of extremely sparse or nonexistent group-item interactions and also has its own special needs. EGR needs to seek the common preferences of the members instead of maximizing the personalized needs of individuals. In particular, group preferences may not necessarily be related to the timeliness and intensity of the member's individual behavior and preferences. Following this line of thought, we propose an EGR model named HL4EGR. Specifically, we adopt hypergraphs to model complex relationships among users, items, and groups, during which we weaken the timeliness and intensity of user behavior and preferences and augment training data by discovering implicit and explicit group-group similarities. Moreover, we design a cross-hypergraph contrastive learning strategy to align embeddings for the same group in different hypergraphs, which enables group preferences to reflect the common preferences of group members comprehensively. We conduct extensive experiments on three real-world datasets, and the experimental results show that our model HL4EGR outperforms state-of-the-art models.

Keywords: Recommender Systems · Group Recommendation · Hypergraph Neural Networks · Contrastive Learning

1 Introduction

In recent years, recommender systems have been offering personalized item recommendations on online services. With the increasing social activities of users, providing group recommendations on online services is poised to become a new and viable pathway to attract users and boost user engagement continuously.

The group recommendation is to recommend items of common interests (such as dining restaurants, travel destinations, and gathering venues) for a group of

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
A. Bifet et al. (Eds.): ECML PKDD 2024, LNAI 14949, pp. 89–105, 2024.
https://doi.org/10.1007/978-3-031-70378-2_6

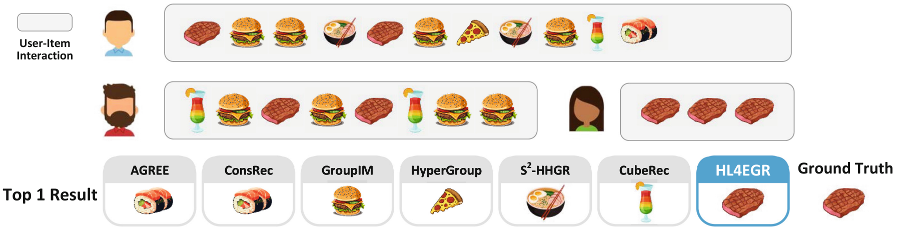


Fig. 1. Case study. The icon denotes the type of restaurant. The sequence of icons in a rectangle denotes the interaction history of the person on the left side.

members, often taking group preferences as a guideline [9,10,13,19]. Depending on the way a group is established, group recommendation can be divided into two categories: persistent group recommendation (PGR) and ephemeral group recommendation (EGR). The former is for a group with fixed members having long-term and extensive interactions between the group and items. The latter is for a temporarily formed group without fixed members, where the group has little or no historical interactions with items, making it impossible to directly learn from those interactions. We focus on EGR in this paper.

We note that the group recommendation, either PGR or EGR, should adhere to the norm of seeking common ground, which facilitates the smooth progress of collective activities. Specifically, the group recommendation models are supposed to treat the common preferences of all the members as the group preferences, and seek factors behind the consensus among group members. In particular, the model should treat the historical and current preferences of a group member equally, without weakening the role of the group member's historical preferences over time. This stems from real-life experience: if a member's historical preferences are the same as the current preferences of the other members in a group, even if the member's current preferences have deviated from his/her historical preferences, the member may still reach a compromise with the other members, accepting the item that is consistent with his/her historical preferences. Moreover, the model are expected to treat the strong preferences of one member and the weak preferences of another member equally, and should not be misled by the strong preferences exhibited by one or a few members.

Taking a real ephemeral group from the Yelp dataset as an example. As depicted in Fig. 1, the group consists of three persons. The person on the top has wide-ranging interests, where steak is his early preference. The person on the middle left has interest in steak and hamburger, and the person on the middle right has interest in steak only. Finally, this group actually visits a steakhouse, a restaurant that is acceptable to all three of them. Unfortunately, existing EGR models fail to obtain the correct result. For the case in the Fig. 1, GroupIM [14] and CubeRec [6] recommend a burger shop and a dessert shop, respectively. It seems that these two models are influenced by the strong personal preferences of two group members. S^2-HHGR [20] and HyperGroup [8] recommend a noodle shop and a pizza shop, respectively. These two models seem to ignore historical preferences of members and be influenced by the user with wide-ranging preferences. In addition, two PGR models, i.e., AGREE [4] and ConsRec [16], also give incorrect results.

To realize "seeking common ground" in group recommendations and figure out a feasible solution to the inherent problem in EGR, that is, group-item interactions are extremely sparse or nonexistent, we propose a multi-hypergraph model named HL4EGR (Hypergraph Learning for Ephemeral Group Recommendation). The model employs hypergraphs to capture the relationship among users, items, and groups, and adopts a two-stage framework consisting of pre-training and fine-tuning. During the pre-training, we choose the hypergraph to model user-item interactions, where a hyperedge connects all the items that a user interacts with, thus equally treating historical interactions and current interactions. The item embeddings obtained by the pre-training are subsequently clustered to identify user preferences. At the stage of fine-tuning, we construct three hypergraphs to model user-group affiliations and two types of group-group similarities, respectively. Here, two types of similarities are given from two different perspectives, one explicit from the perspective of items interacted with by the members and another implicit from the perspective of common preferences of members. Both of them emphasize the commonality of member behavior or member preferences, and weaken the intensity of member behavior or preferences. Further, we maximize the agreement between contrastive views of groups by cross-hypergraph contrastive learning. Finally, we aggregate these group embeddings to generate group preferences and then perform the prediction for groups. For the case in Fig. 1, our model recommends a steakhouse that is in line with the ground truth. Our contributions are summarized as follows.

- We construct four hypergraphs and learn the complex relationships among users, items, and groups through hypergraph convolutions. Particularly, by means of hypergraphs, we weaken the timeliness and intensity of user behavior and preferences and captures their common preferences effectively, thus satisfying the intrinsic requirement of group recommendation.
- We highlight that identifying and leveraging similarities between groups provides a practicable way to cope with the absence of group-item interactions. Moreover, we take group self-discrimination as the self-supervised task, which offers auxiliary supervision signals via two views of a group w.r.t. explicit and implicit group-group similarities for reinforcing group representation learning.
- We conduct extensive experiments on three public datasets. The experimental results show that HL4EGR consistently outperforms the state-of-the-art models, showing relative gains of 8.92%-15.93% on Recall@50 and 13.37%-18.88% on NDCG@50, respectively.

2 Related Work

Early group recommendation adopts collaborative filtering to obtain the member's scores on items and then aggregates their scores to get group preferences by some hand-crafted heuristic rules. Customary aggregation methods include the least misery [1], the average [3], and the maximum satisfaction strategy [2]. However, these predefined aggregation strategies lack the flexibility to achieve optimal performance in group recommendation. Subsequent work on group recommendation shifts towards how to efficiently aggregate the preference representations of all group members to the group preference. For example, multiple group recommendation models such as AGREE [4], SoAGREE [5] and MoSAN [15] propose different attention-based aggregation methods.

With the development of graph neural networks, the tripartite graph [12] has been employed to model users, items, and groups relationships and then learn group representations. Furthermore, hypergraphs are found to be more suitable for modeling groups because hyperedges in the hypergraph can connect two or more nodes and represent a more general topological relationship. Some models [8,10,16,20] apply the hypergraph to model groups and then employ Hypergraph Neural Networks (HNNs) [17] to generate group representations. For example, ConsRec [16] models users and items as nodes, groups as hyperedges, and learns group representations through HNNs. In addition, CubeRec [6] adaptively generates a hypercube representation for each group. However, these models do not discover the essence of user preferences playing a role in group recommendation scenarios. They do not treat strong and weak preferences equally, nor do they give equal weight to historical and current preferences.

Recently, the research on group recommendation [6,14,20] has attempted to incorporate self-supervised learning to alleviate the data sparsity problem. For example, for enhancing the user and group representations, GroupIM [14] proposes maximizing mutual information between members within a group and the group. S^2-HHGR [20] designs a double-scale node dropout strategy and performs node self-discrimination on different user representations. However, existing methods mainly focus on finding self-supervision signals in user-group relationships without considering group-group relationships. Besides, some studies rely on introducing additional information to improve the performance of group recommendations. For example, KGAG [7] introduces knowledge graphs into group recommendation. SIGR [18] and HyperGroup [8] introduce social relationships among users to learn group preferences influenced by social relationships.

Compared to existing work, our model employs multiple hypergraphs to model different relationships among users, items, and groups from multiple perspectives, using the prior about the role of user preferences for group recommendations as an inductive bias of the model. Moreover, our model captures self-supervision signals from the similarities between groups, and then learn more comprehensive group representations.

3 Methodology

3.1 Model Overview

Let $\mathcal{U}$, $\mathcal{V}$ and $\mathcal{G}$ denote the user set, item set, and ephemeral group set, respectively. An ephemeral group $g_k \in \mathcal{G}$ consists of $|g_k|$ users, i.e., $g_k = \{u_i^{g_k}\}_{i=1}^{|g_k|}$,

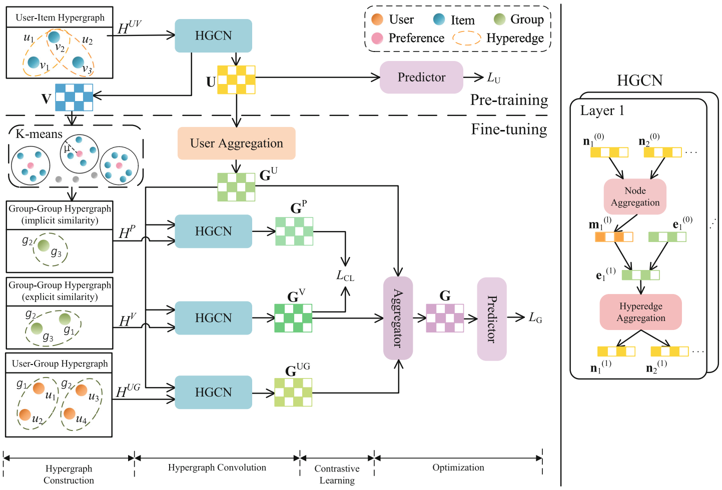


Fig. 2. Architecture of our HL4EGR.

where $u_i^{g_k} \in \mathcal{U}$. There are two types of observed interactions among users, items, and ephemeral groups, i.e., user-item interactions denoted as $\mathbf{X} \in \mathbb{R}^{|\mathcal{U}| \times |\mathcal{V}|}$, and group-item interactions represented as $\mathbf{Y} \in \mathbb{R}^{|\mathcal{G}| \times |\mathcal{V}|}$, where the element y_{kj} of the matrix $\mathbf{Y}$ is equal to 1 if the group g_k has historical interactions with the item v_j, otherwise $y_{kj} = 0$.

Given an ephemeral group g_k, our task is to predict the item that the group g_k is most likely to be satisfied with.

For this task, we propose a multi-hypergraph model HL4EGR, whose architecture is shown in Fig. 2. We build four hypergraphs to model the user-item interactions, the user-group affiliations, and the explicit and implicit group-group similarities.

As shown in Fig. 2, the training of HL4EGR is divided into two stages, i.e., pre-training and fine-tuning.

In the first stage, we construct the user-item hypergraph H^{UV} and perform the convolution operation on H^{UV}, thus obtaining the user embeddings $\mathbf{U}$ and item embeddings $\mathbf{V}$. $\mathbf{U}$ is utilized for initializing the group embeddings used in the second stage and $\mathbf{V}$ is applied to characterize the user preferences by clustering.

In the second stage of training, i.e., fine-tuning, except for constructing the user-group hypergraph H^{UG}, we also construct two group-group hypergraphs H^V and H^P, portraying explicit and implicit similarities between groups, respectively. Then we perform the hypergraph convolution operations to obtain group embeddings. Furthermore, we adopt a cross-hypergraph contrastive learning strategy to align embeddings of the same group from both explicit and implicit perspectives, thus obtaining more comprehensive group preferences that are devoted to group recommendation.

3.2 Hypergraph Construction

User-Item Hypergraph . We define the user-item hypergraph as $H^{UV} = (\mathcal{V}, \mathcal{E}^{UV})$, where a node of H^{UV} is an item in $\mathcal{V}$, a hyperedge $e_i^{UV} \in \mathcal{E}^{UV}$, $i \in [1, |\mathcal{U}|]$ connects all the items that user u_i interacts with, and $|\mathcal{E}^{UV}| = |\mathcal{U}|$. As shown in Fig. 2, user u_1 has historical interactions with item v_1 and item v_2, thus we connect $\{v_1, v_2\}$ with a hyperedge. Such hyperedges eliminate temporal differentiations of interactions, treating historical interactions and current interactions equally.

User-Group Hypergraph . We define the user-group hypergraph as $H^{UG} = (\mathcal{U}, \mathcal{E}^{UG})$, where a node of H^{UG} is a user in $\mathcal{U}$, a hyperedge $e_k^{UG} \in \mathcal{E}^{UG}$, $k \in [1, |\mathcal{G}|]$ connects all the users in group g_k, and $|\mathcal{E}^{UG}| = |\mathcal{G}|$. As shown in Fig. 2, we connect the group member $\{u_1, u_2\}$ of the group g_1 with a hyperedge, which reflects the user-group affiliation.

Group-Group Hypergraphs . For alleviating the data sparsity issue, we construct two group-group hypergraphs, i.e., $H^V = (\mathcal{G}, \mathcal{E}^V)$ and $H^P = (\mathcal{G}, \mathcal{E}^P)$.

In hypergraph H^V, $\mathcal{G}$ is taken as the node set, and a hyperedge $e_k^V \in \mathcal{E}^V$, $k \in [1, |\mathcal{G}|]$ connects all such groups, provided that a member of that group and a member of group g_k interact with the same item. In other words, hypergraph H^V contains the explicit similarities between groups.

Complementary to H^V, hypergraph H^P implies the implicit similarities between groups, i.e., the preference similarities between groups. The group preference is essentially a collection of member preferences. Specifically, with the consideration of the interference of noisy behavior, we regard the items that all users have interacted with as the starting point to model the user's preferences, instead of capturing the user's preferences from a user's behavior. We perform K-means clustering on the item embeddings $\mathbf{V}$ obtained by the pre-training on the hypergraph H^{UV} and generate c clustering centers. Next, for each user, given an item that this user has interacted with, if the distance between the item embedding and the center of the category the item belongs to is less than μ, this center is considered to be a preference of this user. Subsequently, the preferences of group members are merged to form a set of group preferences. Then, we build a hyperedge $e_k^P \in \mathcal{E}^P$, $k \in [1, |\mathcal{G}|]$ to connect the groups that has common preferences with group g_k.

When building the group-group hypergraphs, we treat all the hyperedges equally (i.e., hyperedges with same weights), thus flattening the intensity of a user's individual behavior and preferences. This enables HL4EGR to more fairly learn the common preferences of users within the group, reducing the impact of the intensity of a user's personal behavior and preferences on the group-group similarity.

3.3 Hypergraph Convolution

In HL4EGR, we design a HyperGraph Convolutional Network (HGCN) to learn representations of nodes and hyperedges in a hypergraph. Without loss of generality, we formalize four hypergraphs uniformly as $H = (\mathcal{N}, \mathcal{E})$, where $\mathcal{N}$ denotes the node set and $\mathcal{E}$ denotes the hyperedge set. The learning process of the l-th layer of HGCN is as follows.

Firstly, we aggregate representations of all nodes connected by hyperedge e_k as follows.

$$\mathbf{m}_k^{(l)} = \mathrm{AGG}(\mathbf{n}_i^{(l-1)} | n_i \in e_k) \tag{1}$$

where $e_k \in \mathcal{E}$ denotes the k-th hyperedge, $\mathbf{n}_i^{(0)}$ is the initial embedding of node $n_i \in \mathcal{N}$, $\mathbf{n}_i^{(l-1)}$ is the embedding of the node n_i in the $(l-1)$-th layer, $\mathrm{AGG}(\cdot)$ denotes an aggregation function, realized as an average pooling function.

Then, we concatenate node aggregation representation $\mathbf{m}_k^{(l)}$ and hyperedge representation $\mathbf{e}_k^{(l-1)}$ to update the hyperedge representation as follows.

$$\mathbf{e}_k^{(l)} = \mathrm{CONCAT}(\mathbf{m}_k^{(l)}, \mathbf{e}_k^{(l-1)})\mathbf{W}^H \tag{2}$$

where $\mathbf{e}_k^{(0)}$ is the initial embedding of hyperedge e_k, $\mathbf{e}_k^{(l)}$ denotes the embedding of the hyperedge e_k in the l-th layer. $\mathbf{W}^H \in \mathbb{R}^{2d \times d}$ is a learnable matrix.

Moreover, node representations can be updated as follows.

$$\mathbf{n}_i^{(l)} = \mathrm{AGG}(\mathbf{e}_k^{(l)} | e_k \in \mathcal{E}_i) \tag{3}$$

where $\mathcal{E}_i$ represents the set of hyperedges connected to the node n_i.

Finally, we can obtain the embedding $\mathbf{n}_i$ of the node n_i, and the embedding $\mathbf{e}_k$ of the hyperedge e_k as follows.

$$\mathbf{n}_i = \sum_{l=1}^{L} \mathbf{n}_i^{(l)}, \quad \mathbf{e}_k = \sum_{l=1}^{L} \mathbf{e}_k^{(l)} \tag{4}$$

where L is the number of convolutional layers.

During pre-training, we first randomly initialize the representations of nodes and hyperedges in H^{UV} and feed them into an HGCN. Then, we iterate and optimize the HGCN by the cross entropy loss (L_U in Fig. 2). After pre-training, we obtain user embeddings $\mathbf{U} \in \mathbb{R}^{|\mathcal{U}| \times d}$ and item embeddings $\mathbf{V} \in \mathbb{R}^{|\mathcal{V}| \times d}$.

During fine-tuning, we first aggregate user embeddings $\mathbf{U}$ to generate group embeddings $\mathbf{G}^U \in \mathbb{R}^{|\mathcal{G}| \times d}$. In detail, taking the group g_k as an example, we leverage an attention mechanism to aggregate embeddings of users in the group g_k, thereby obtaining the initial group representation $\mathbf{g}_k^U = \mathbf{G}^U(k, :)$. This process can be formalized as follows.

$$\mathbf{g}_k^U = \sum_{u_i \in g_k} \alpha_i \mathbf{u}_i \tag{5}$$

$$\alpha_i = \frac{\exp(tanh(\mathbf{u}_i \mathbf{W}^{AGG} + b))}{\sum_{u_{i'} \in g_k} \exp(tanh(\mathbf{u}_{i'} \mathbf{W}^{AGG} + b))} \tag{6}$$

where $\mathbf{u}_i = \mathbf{U}(i,:)$ denotes the embedding of the user u_i obtained by pre-training, α_i is the attention weight w.r.t. the user u_i. $\mathbf{W}^{AGG} \in \mathbb{R}^d$ is a learnable vector and b is a bias.

Next, we use $\mathbf{G}^U$ to initialize hyperedges of H^{UG}, H^V and H^P, and nodes of H^V and H^P, and use user embeddings $\mathbf{U}$ obtained by the pre-training to initialize node representations on the hypergraph H^{UG}, and then feed them into corresponding HGCNs.

Finally, by performing the calculations over these three HGCNs, we obtain representations of hyperedges in three hypergraphs, denoted as $\mathbf{G}^{UG}$, $\mathbf{G}^V$, and $\mathbf{G}^P$, respectively. Given group g_k, its embeddings from three hypergraphs are $\mathbf{g}_k^{UG} = \mathbf{G}^{UG}(k,:)$, $\mathbf{g}_k^V = \mathbf{G}^V(k,:)$, and $\mathbf{g}_k^P = \mathbf{G}^P(k,:)$, respectively.

3.4 Cross-Hypergraph Contrastive Learning

To learn more comprehensive group preferences, we design a contrastive learning strategy on two group-group hypergraphs, i.e., the hypergraph H^V reflecting explicit similarities and the hypergraph H^P implying implicit similarities, aligning two embeddings of the same group in H^V and H^P. Concretely, we regard the representations w.r.t. the same group in two hypergraphs H^V and H^P as positive sample pairs. The representations w.r.t. different groups in the same batch in two hypergraphs H^V and H^P are considered as negative sample pairs. We take InfoNCE loss as the contrastive learning loss as follows.

$$L_{CL} = -\sum_{g_k \in \mathcal{G}} \log \frac{\exp(sim(\mathbf{g}_k^V, \mathbf{g}_k^P)/\tau)}{\exp(sim(\mathbf{g}_k^V, \mathbf{g}_k^P)/\tau) + N_V + N_P} \tag{7}$$

$$N_V = \sum_{g_{k'} \in \mathcal{G}_k^-} \exp(sim(\mathbf{g}_{k'}^V, \mathbf{g}_k^P)/\tau), \quad N_P = \sum_{g_{k'} \in \mathcal{G}_k^-} \exp(sim(\mathbf{g}_k^V, \mathbf{g}_{k'}^P)/\tau) \tag{8}$$

where $\mathbf{g}_k^V$ and $\mathbf{g}_k^P$ form a pair of positive samples, corresponding to the representations of the group g_k in the hypergraph H^V and H^P, respectively. $\mathcal{G}_k^-$ is the set of negative samples w.r.t. the group g_k, which is composed of other groups (i.e., $k' \neq k$) within the same batch. $sim(\cdot)$ function is adopted for calculating the similarity of a pair of vectors, which refers to the cosine similarity in this paper. τ is the temperature parameter.

3.5 Model Optimization

During pre-training, we predict the interaction probabilities $\hat{\mathbf{x}}_\mathbf{i} \in R^{|\mathcal{V}|}$ of user u_i on the item set $\mathcal{V}$ as follows.

$$\hat{\mathbf{x}}_\mathbf{i} = softmax(\mathbf{u}_i \mathbf{W}^{UV}) \tag{9}$$

where $\mathbf{u}_i = \mathbf{U}(i,:)$obtained from hypergraph H^{UV}, $\mathbf{W}^{UV} \in \mathbb{R}^{d \times |\mathcal{V}|}$ is a learnable matrix.

Then we calculated the cross entropy loss L_U as follows.

$$L_U = -\frac{1}{|\mathcal{U}|}\sum_{i=1}^{|\mathcal{U}|}\sum_{j=1}^{|\mathcal{V}|} x_{ij}\log \hat{x}_{ij} \tag{10}$$

where $\hat{x}_{ij}$ refers to the interaction probability of the user u_i w.r.t. the item v_j. x_{ij} is the ground truth of user-item interaction.

During fine-tuning, given group representations from different hypergraphs, we adopt an adaptive aggregation strategy to fuse different group embeddings, i.e., $\mathbf{g}_k^U$ obtained from Eq. 5, $\mathbf{g}_k^{UG}$ in the hypergraph H^{UG}, and $\mathbf{g}_k^V$ in the hypergraph H^V, to generate the group preference $\mathbf{g}_k$ for the group g_k as follows.

$$\mathbf{g}_k = \alpha\mathbf{g}_k^U + \beta\mathbf{g}_k^{UG} + \gamma\mathbf{g}_k^V \tag{11}$$

where $\alpha = \sigma(\mathbf{g}_k^U\mathbf{W}^U)$, $\beta = \sigma(\mathbf{g}_k^{UG}\mathbf{W}^{UG})$, and $\gamma = \sigma(\mathbf{g}_k^V\mathbf{W}^V)$. $\mathbf{W}^U$, $\mathbf{W}^{UG}$, and $\mathbf{W}^V \in \mathbb{R}^d$ are learnable matrices. σ is the sigmoid activation function.

We predict the interaction probabilities $\hat{\mathbf{y}}_k \in \mathbb{R}^{|\mathcal{V}|}$ of the group g_k on the item set $\mathcal{V}$ as follows.

$$\hat{\mathbf{y}}_k = softmax(\mathbf{g}_k\mathbf{W}^{GV}) \tag{12}$$

where $\mathbf{W}^{GV} \in \mathbb{R}^{d\times|\mathcal{V}|}$ is a learnable matrix.

Then, we adopt the cross entropy loss as the main loss, calculated as follows.

$$L_G = -\frac{1}{|\mathcal{G}|}\sum_{k=1}^{|\mathcal{G}|}\sum_{j=1}^{|\mathcal{V}|} y_{kj}\log \hat{y}_{kj} \tag{13}$$

where $\hat{y}_{kj}$ refers to the interaction probability of the group g_k w.r.t. the item v_j. y_{kj} is the ground truth.

We adopt a multi-task strategy to jointly optimize the main group recommendation task and the auxiliary contrastive learning task as follows.

$$L = L_G + \lambda L_{CL} \tag{14}$$

where λ is a hyperparameter.

3.6 Complexity Analysis

Space Complexity. In HL4EGR, the learnable parameters are mainly from embeddings of users, items, and groups. In addition, as for hypergraph convolutions, since we have four hypergraphs in two stages, each with L layers, the number of parameters is $4L \times 2d^2$. The number of parameters for two prediction layers in two stages is $2|\mathcal{V}|d$. Thus, the space complexity of HL4EGR is $O(Ld^2 + |\mathcal{U}|d + |\mathcal{V}|d + |\mathcal{G}|d)$.

Time Complexity. The computation amount of HL4EGR is mainly concentrated on the hypergraph convolutions. Let $|H|$ be the number of nonzero elements in the adjacency matrix of hypergraph H. The time complexity of each hypergraph convolution computation is $O(L \times (2|H|d + 2|\mathcal{E}|d^2))$, where $|\mathcal{E}|$ is the number of hyperedges. For hypergraphs H^{UV}, H^{UG}, H^{V}, and H^{P}, the numbers of hyperedges are $|\mathcal{U}|$, $|\mathcal{G}|$, $|\mathcal{G}|$, and $|\mathcal{G}|$, respectively. The total time complexity of HL4EGR is $O(Ld^2|\mathcal{G}| + Ld^2|\mathcal{U}| + Ld(|H^{UV}| + |H^{UG}| + |H^{V}| + |H^{P}|))$.

Table 1. Statistics of datasets.

Dataset	# Users	# Items	# Groups	# User-Item interactions	# Group-Item interactions	Avg. of Group-Item interactions	Avg. group size
Weeplaces	8,643	25,081	22,733	508,486	67,133	2.95	2.89
Yelp	34,504	22,611	24,103	482,273	26,883	1.18	4.45
Douban	25,377	21,696	72,962	2,817,094	118,022	1.62	5.23

4 Experiments

4.1 Experimental Settings

Datasets. We conduct experiments on three public datasets.

- **Weeplaces**. It records users' check-ins in location-based social networks. We extract check-ins from points of interest (POIs) in all major cities in the U.S. We follow the same operations as in GroupIM [14] for constructing user-POI interactions and group-POI interactions.
- **Yelp**. It records users' check-ins in local businesses (e.g., restaurants). We use the dataset published in [18], which includes users' check-ins on businesses located in Los Angeles, as well as groups' check-in information.
- **Douban**. It is also published in [18], recording the information of users organizing and participating in social activities. We filter out users and items with fewer than 10 interactions.

Table 1 lists the statistics of the three datasets. As shown in Table 1, the average of group-item interactions is less than 3, which manifests that we conduct experiments on ephemeral groups. We randomly split all the groups of each dataset into training, validation, and test sets with a ratio of 7:1:2. We ensure that each group can only appear in one of the three sets.

Baselines. We compare HL4EGR to the following baselines:

- Two PGR models: **AGREE**[1], which is a classical PGR model using an attention mechanism for member aggregation [4]. **ConsRec**[2], the state-of-the-art

---

[1] https://github.com/LianHaiMiao/Attentive-Group-Recommendation.

[2] https://github.com/FDUDSDE/WWW2023ConsRec.

model for PGR, which proposes an HNN to learn member-level aggregation and captures the group consensus on three views [16].

- Four EGR models: **GroupIM**[3], which maximizes user-group mutual information for group recommendation [14]. **HyperGroup**[4], which models groups as hyperedges to learn group representations [8]. **S^2-HHGR**[5], which uses a hierarchical hypergraph and a node dropout strategy on the hypergraph to learn group preferences [20]. **CubeRec**[6], the state-of-the-art model for EGR, which utilizes the geometric expressiveness of hypercubes and hypercube intersection-based self-supervision to obtain the group representations [6].

Table 2. Overall performance. The values in bold and underlined are the best and second best results in each row.

Weeplaces

Metric	AGREE	ConsRec	GroupIM	HyperGroup	S^2-HHGR	CubeRec	HL4EGR	Improv.(%)
R@20	0.1612	0.1949	0.4132	0.3926	0.3930	0.4119	**0.4776**	15.58
R@50	0.2927	0.3686	0.5688	0.5438	0.5337	0.5719	**0.6367**	11.33
N@20	0.0637	0.0731	0.2218	0.1720	0.1794	0.2162	**0.2549**	14.92
N@50	0.0913	0.1062	0.2528	0.1964	0.1958	0.2485	**0.2866**	13.37

Yelp

Metric	AGREE	ConsRec	GroupIM	HyperGroup	S^2-HHGR	CubeRec	HL4EGR	Improv.(%)
R@20	0.1940	0.2236	0.5548	0.5247	0.5276	0.5802	**0.6647**	14.56
R@50	0.3388	0.3753	0.6603	0.6273	0.6158	0.6995	**0.7619**	8.92
N@20	0.0898	0.1025	0.3743	0.3408	0.3580	0.3790	**0.4503**	18.81
N@50	0.1168	0.1319	0.3953	0.3612	0.3719	0.4027	**0.4696**	16.61

Douban

Metric	AGREE	ConsRec	GroupIM	HyperGroup	S^2-HHGR	CubeRec	HL4EGR	Improv.(%)
R@20	0.1236	0.1471	0.2184	0.2103	0.2014	0.2244	**0.2771**	23.48
R@50	0.2557	0.2913	0.3634	0.3574	0.3371	0.3715	**0.4307**	15.93
N@20	0.0455	0.0550	0.0886	0.0841	0.0838	0.0926	**0.1142**	23.32
N@50	0.0728	0.0841	0.1173	0.1131	0.1106	0.1218	**0.1448**	18.88

Implementation Details. We implement our model in PyTorch. In our model, the number of hypergraph convolutional layers L is set to 2 and temperature τ is set to 1. We tune the weight of contrastive learning loss λ, the number of clustering centers c, the threshold of distance to any clustering center μ for every dataset, finally setting λ to 0.3, μ to 0.2 for all datasets, c to 64 for Weeplaces, 128 for Yelp and Douban. We optimize the model via the Adam optimizer with

---

[3] https://github.com/CrowdDynamicsLab/GroupIM.

[4] https://github.com/FDUDSDE/WWW2023GroupRecBaselines.

[5] https://github.com/0411tony/HHGR.

[6] https://github.com/jinglong0407/CubeRec.

the learning rate 0.001. The implementation code has been released[7]. For the sake of fairness, we set the size of all embeddings d to 64, the batch size to 256 in all the experiments. For all baselines, the hyperparameters are set to values corresponding to best performance reported in their respective papers. Experiments are conducted on NVIDIA RTX3090 GPU with 24G memory.

Metrics. To evaluate the performance of recommending items to groups, we adopt two metrics, i.e., Recall@K and NDCG@K (R@K and N@K for short), where Recall focuses on whether the group actually chooses the recommended item, NDCG focuses on the ranking of the recommended items and K is set to either 20 or 50.

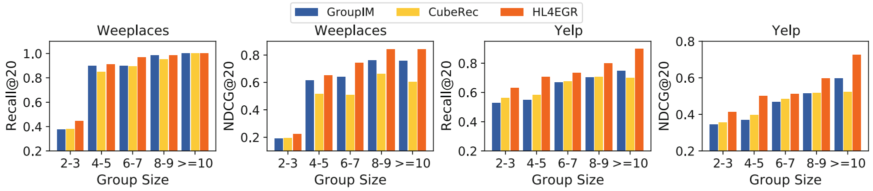


Fig. 3. Group recommendation performance on groups of different sizes.

4.2 Performance Comparison

Overall Performance. Table 2 lists the experimental results of our proposed model and compared models on the three datasets. From Table 2, we have the following observations.

- The PGR models are far inferior to the EGR models in all metrics. This is because PGR models depend on group-item interactions to learn group preferences; however, these interactions become extremely sparse or nonexistent in the context of ephemeral groups, ultimately leading to a decline in performance.
- Hypergraph-based models, i.e., ConsRec, HyperGroup, S^2-HHGR, and HL4EGR outperform the traditional attention-based model, i.e., AGREE, which demonstrates that the hypergraph structure excels in modeling user-group affiliations.
- Three EGR models equipped with self-supervised learning, i.e., GroupIM, CubeRec, and HL4EGR, outperform other models. This might be attributed to the fact that these EGR models can discover and utilize additional supervision signals, thus improving the quality of group embeddings. This shows the advantages of self-supervised learning in EGR.

---

[7] https://github.com/ZhaoRui-7/HL4EGR.

- Our HL4EGR outperforms all baselines on three datasets. Taking Recall@20 as an example, compared to the best baseline on each of the three datasets, HL4EGR shows improvements of 14.56% - 23.48%, averaging at 17.87%.

Performance on Groups of Different Sizes. We split the test set into five subsets by the range of the number of group members, i.e., 2-3, 4-5, 6-7, 8-9, and >=10 members. We choose GroupIM and CubeRec for comparison because they are the top-2 best baselines, and we conduct experiments on Weeplaces and Yelp datasets.

As shown in Fig. 3, HL4EGR outperforms GroupIM and CubeRec in almost all cases, except on groups of 10 or more members in Weeplaces, where all three models have the same Recall values (reaching the maximum value of 1). In particular, HL4EGR outperforms the other two models for the case of groups of 2-3 members, indicating that HL4EGR is more suitable for real-life group recommendations, where the size of groups shows the long-tail distribution. Meanwhile, compared to other two models, HL4EGR has more significant performance gains for groups of 10 or more members in Yelp. The reason behind might be that HL4EGR can correct group representations by treating all behavior and preferences of all members indiscriminately in terms of timeliness and intensity, thus capturing common preferences of groups more accurately, while the number of group members increases.

Table 3. Ablation study.

(a) Effect of multiple hypergraphs

Model	Weeplaces		Yelp	
	R@20	N@20	R@20	N@20
HL4EGR	**0.4776**	**0.2549**	**0.6647**	**0.4503**
(A) w/o H^{UV}	0.3674	0.1916	0.4929	0.3233
(B) w/o H^{P}	0.4674	0.2442	0.6542	0.4335
(C) w/o H^{V}	0.4585	0.2421	0.6377	0.4226
(D) w/o H^{UG}	0.4716	0.2498	0.6546	0.4384

(b) Effect of pre-training or weights

Model	Weeplaces		Yelp	
	R@20	N@20	R@20	N@20
(E)	0.3079	0.1651	0.4596	0.3093
(F)	0.4009	0.2201	0.5634	0.4052
(G)	0.4644	0.2419	0.6546	0.4401
(H)	0.4715	0.2501	0.6563	0.4372
(I)	0.4770	0.2500	0.6630	0.4466

4.3 Ablation Study

Effect of Multiple Hypergraphs. We design four variants to observe the effect of four hypergraphs in HL4EGR on the performance. Variant A deletes the HGCN on H^{UV} but also performs the pre-training, taking randomly initialized user and item embeddings as input and cross entropy loss, i.e., L_U as the optimization goal. Variant B removes the HGCN on H^P, which triggers a cascading removal of the contrastive learning module, since H^P is treated as a source of self-supervision signals. Variant C removes the HGCN on H^V, which leads to the removal of contrastive learning module as well as the reduction of one source of the group preference. Variant D eliminates the HGCN on H^{UG}.

The experimental results on Weeplaces and Yelp are listed in Table 3(a). Compared to HL4EGR, all variants show different degrees of performance degradation, illustrating that each hypergraph is effective. Variant A shows the notable performance degradation, indicating that hypergraph H^{UV} is the underpinning of the whole model. The direct reason behind this is that the user and item embeddings derived from H^{UV} are subsequently used for the construction and learning of other hypergraphs, which imposes a great positive impact on performance. The performance decrease of variant B shows the effect of alleviating data sparsity and adjusting group embeddings via contrastive learning. Variant C has a significant decline in performance, while compared to variant D, which shows that group-group relationships play a more important role than inherent memberships of groups in group recommendation.

Effect of Pre-Training. To observe the impact of pre-training, we construct two extra variants of HL4EGR, namely variants E and F. Variant E removes the pre-training, thus collapsing into the scaled-down version of only containing H^{UG} and H^V and taking randomly initialized user embeddings as the input of fine-tuning. Variant F substitutes SASRec [11] for the HGCN on H^{UV}.

The performance of variants E and F is shown in the top half of Table 3(b). Variant E without pre-training shows the worst performance, indicating that the pre-training is indispensable.

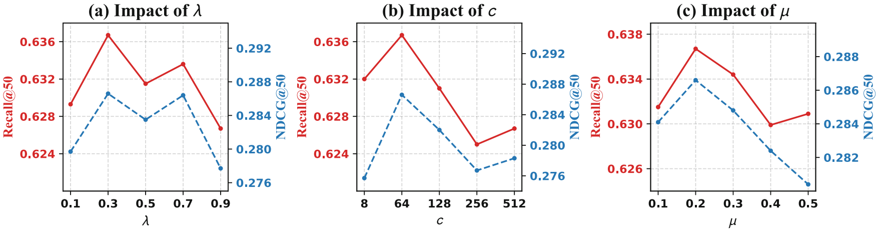


Fig. 4. Sensitivity analysis of hyperparameters λ, c and μ on Weeplaces dataset.

Variant F was originally anticipated to exhibit high performance because SASRec adopts a self-attentive mechanism that learns both long-term and short-term dependencies and produces high-quality user and item embeddings. However, experimental results show that variant F does not surpass the original HL4EGR. This observation reveals that the self-attention mechanism, which is good at capturing temporal dependencies embedded in sequences, does little to help eliciting the group preferences. Presumably the reason for this would be that group preferences are time-insensitive.

Effect of Hyperedge Weights. As mentioned in Sect. 3.2, when constructing group-group similarity hypergraphs in HL4EGR, the weights on the hyperedges are assigned to the same value, aiming to weaken the effect of the intensity of individual member behavior and preferences. To observe how weight values affect performance, we modify H^V and H^P by setting weights to the number of items interacted with by both members of two groups and the number of common preferences of two groups, respectively, and construct three variants of HL4EGR. Variant G introduces weights on H^V and H^P. Variant H introduces weights only on H^V while variant I does so only on H^P.

The experimental results are shown in the bottom half of Table 3(b). It can be seen that variants G, H and I have lower performance than HL4EGR. In particular, variant G has a significant performance degradation, which shows that simultaneously emphasizing the intensities of member behavior and preferences has a significant negative effect on group recommendation.

4.4 Hyperparameter Sensitivity Analysis

Impact of Contrastive Loss Weight λ. Figure 4(a) shows the results on Weeplaces dataset. This shows that appropriate contrastive learning loss can effectively normalize the group representations.

Impact of Number of Clustering Centers c. Figure 4(b) shows the results. HL4EGR achieves best performance on Weeplaces when $c = 64$. From Fig. 4(b), we believe that when c is very small, the model is unable to distinguish a user's different preferences, leading to false similarity when modeling group-group similarity. When c is too large, the model tends to identify the same preference of a user as multiple preferences, which fails to weaken the intensity of user's individual preference.

Impact of Distance Threshold μ. Figure 4(c) shows the results. HL4EGR achieves best performance when $\mu = 0.2$. We think that when μ is very small, items that indicate user preferences are filtered out; when μ becomes large, more items including noisy items are retained. Both cause performance reduction.

5 Conclusion

Ephemeral group recommendation is a challenging recommendation task, not only because group-item interactions are not enough to learn group preferences directly, but also because there are essential differences between group recommendation and personalized recommendation. This paper proposes a model HL4EGR that models the user-item interactions, user-group affiliations, and group-group similarities into multiple hypergraphs, reflecting the essence of the group recommendation. Meanwhile, HL4EGR also designs a contrastive learning strategy on the hypergraphs, which enables HL4EGR to learn more comprehensive group preferences. The results of experiments on public datasets show that HL4EGR substantially improves the accuracy of ephemeral group recommendation results.

Acknowledgment. This work was supported by the National Natural Science Foundation of China under Grant No. 62072450.

References

1. Amer-Yahia, S., Roy, S.B., Chawlat, A., Das, G., Yu, C.: Group recommendation: semantics and efficiency. Proc. VLDB Endowment **2**(1), 754–765 (2009)
2. Baltrunas, L., Makcinskas, T., Ricci, F.: Group recommendations with rank aggregation and collaborative filtering. In: Proceedings of the Fourth ACM Conference on Recommender Systems, pp. 119–126 (2010)
3. Boratto, L., Carta, S.: State-of-the-art in group recommendation and new approaches for automatic identification of groups. In: Soro, A., Vargiu, E., Armano, G., Paddeu, G. (eds.) Information retrieval and mining in distributed environments, vol. 324, pp. 1–20. Springer, Berlin, Heidelberg (2011). https://doi.org/10.1007/978-3-642-16089-9_1
4. Cao, D., He, X., Miao, L., An, Y., Yang, C., Hong, R.: Attentive group recommendation. In: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, pp. 645–654 (2018)
5. Cao, D., He, X., Miao, L., Xiao, G., Chen, H., Xu, J.: Social-enhanced attentive group recommendation. IEEE Trans. Knowl. Data Eng. **33**(3), 1195–1209 (2019)
6. Chen, T., Yin, H., Long, J., Nguyen, Q.V.H., Wang, Y., Wang, M.: Thinking inside the box: learning hypercube representations for group recommendation. In: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 1664–1673 (2022)
7. Deng, Z., Li, C., Liu, S., Ali, W., Shao, J.: Knowledge-aware group representation learning for group recommendation. In: 2021 IEEE 37th International Conference on Data Engineering (ICDE), pp. 1571–1582. IEEE (2021)
8. Guo, L., Yin, H., Chen, T., Zhang, X., Zheng, K.: Hierarchical hyperedge embedding-based representation learning for group recommendation. ACM Trans. Inf. Syst. (TOIS) **40**(1), 1–27 (2021)
9. He, Z., Chow, C.Y., Zhang, J.D.: Game: learning graphical and attentive multi-view embeddings for occasional group recommendation. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 649–658 (2020)
10. Jia, R., Zhou, X., Dong, L., Pan, S.: Hypergraph convolutional network for group recommendation. In: 2021 IEEE International Conference on Data Mining (ICDM), pp. 260–269. IEEE (2021)
11. Kang, W.C., McAuley, J.: Self-attentive sequential recommendation. In: 2018 IEEE International Conference on Data Mining (ICDM), pp. 197–206. IEEE (2018)
12. Li, K., Wang, C.D., Lai, J.H., Yuan, H.: Self-supervised group graph collaborative filtering for group recommendation. In: Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining, pp. 69–77 (2023)
13. Sajjadi Ghaemmaghami, S., Salehi-Abari, A.: Deepgroup: group recommendation with implicit feedback. In: Proceedings of the 30th ACM International Conference on Information & Knowledge Management, pp. 3408–3412 (2021)

14. Sankar, A., Wu, Y., Wu, Y., Zhang, W., Yang, H., Sundaram, H.: GroupIM: a mutual information maximization framework for neural group recommendation. In: Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, pp. 1279–1288 (2020)
15. Vinh Tran, L., Nguyen Pham, T.A., Tay, Y., Liu, Y., Cong, G., Li, X.: Interact and decide: medley of sub-attention networks for effective group recommendation. In: Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 255–264 (2019)
16. Wu, X., Xiong, Y., Zhang, Y., Jiao, Y., Zhang, J., Zhu, Y., Yu, P.S.: ConsRec: learning consensus behind interactions for group recommendation. In: Proceedings of the ACM Web Conference 2023, pp. 240–250 (2023)
17. Yadati, N., Nimishakavi, M., Yadav, P., Nitin, V., Louis, A., Talukdar, P.: HyperGCN: a new method for training graph convolutional networks on hypergraphs. In: Advances in Neural Information Processing Systems, vol. 32 (2019)
18. Yin, H., Wang, Q., Zheng, K., Li, Z., Yang, J., Zhou, X.: Social influence-based group representation learning for group recommendation. In: 2019 IEEE 35th International Conference on Data Engineering (ICDE), pp. 566–577. IEEE (2019)
19. Yuan, Q., Cong, G., Lin, C.Y.: COM: a generative model for group recommendation. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 163–172 (2014)
20. Zhang, J., Gao, M., Yu, J., Guo, L., Li, J., Yin, H.: Double-scale self-supervised hypergraph learning for group recommendation. In: Proceedings of the 30th ACM International Conference on Information & Knowledge Management, pp. 2557–2567 (2021)

Spoofing Transaction Detection with Group Perceptual Enhanced Graph Neural Network

Le Kang[1,2](✉), Tai-Jiang Mu[1], and XiaoDong Ning[2]

[1] Department of Computer Science and Technology, Tsinghua University, Beijing, China
[2] Zhengzhou Commodity Exchange, Zhengzhou, China
`kangl18@mails.tsinghua.edu.cn`

Abstract. Spoofing involves gaining illegal profits through the strategic deployment of a series of deceptive orders to mislead the actions of other traders. This dishonest practice poses a significant threat to the robustness and sustainability of real-world financial markets. Recently, more and more fraudsters have organized like an enterprise with a higher level of concealment, making it very challenging to detect in practice. Existing efforts primarily focus on identifying individual spoofing behavior or specific spoofing orders, ignoring organized spoofing transactions in the form of groups. Therefore, in this paper, we propose GPEGNN, a novel Group Perceptual-Enhanced Graph Neural Network approach, aimed at detecting organized spoofing transactions. In particular, we first construct the user's trading behavior into transaction graphs. Then, we devise *local* context learning layers to represent the node-level features by a multi-layer graph attention network, and the edge-level features by the self-attention and multilayer perceptron network. Concurrently, we incorporate a *global* context learning module to learn group-level features by community-centric encoder. These *local* and *global* features are jointly optimized within the detection network and trained in an end-to-end framework. Our proposed approach is evaluated by real-world application scenarios in a commodity futures exchange. The experimental results demonstrate the superiority of our proposed approach, showcasing its ability to identify more instances of spoofing trades with relatively high precision compared with state-of-the-art baselines. Our approach thereby bridges the gap between spoofing detection industrial practice and graph learning theory, which we believe could conduct a positive impact on the health of the financial trading ecosystem.

Keywords: spoofing detection · graph attention network · group-level features · community-centric encoder

1 Introduction

Spoofing transaction [18], in the context of financial markets, refers to a deceptive trading behavior where traders engage in manipulative activities to create a false

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
A. Bifet et al. (Eds.): ECML PKDD 2024, LNAI 14949, pp. 106–122, 2024.
https://doi.org/10.1007/978-3-031-70378-2_7

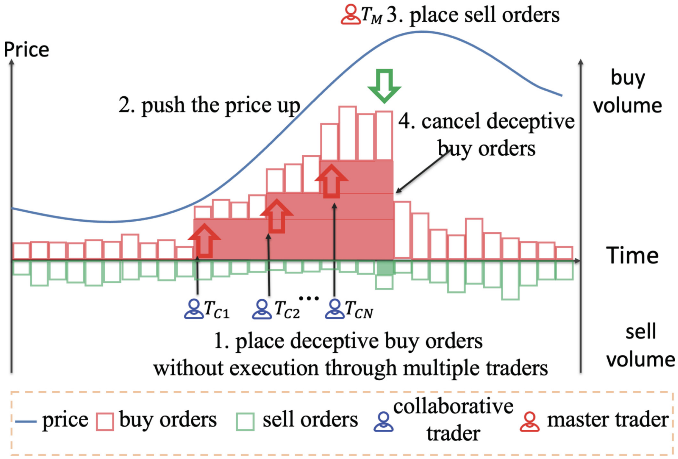


Fig. 1. A typical example of the challenging group-based spoofing problem.

or misleading impression in the market. The purpose of spoofing transaction is to influence the prices of financial instruments, such as stocks, bonds or futures, thereby enhance profits [2]. As a significant form of market manipulation, spoofing not only inflicts extensive economic losses to the financial system, triggering instability in the financial market, but also seriously disrupts the normal order of the financial market. Thus, spoofing is strictly forbidden by all major financial markets around the world [17]. For example, in 2020, the Commodity Futures Trading Commission (CFTC) ordered JPMorgan to pay record $920 million for spoofing and manipulation[1].

Various jurisdictions [16,26] have implemented regulations to detect spoofing behaviors from statistical rules [15] to machine learning methods [10]. With the rapid evolution of high-frequency trading techniques [20], spoofing have become so deceptive that they cannot be detected by classical machine learning models, which treat each trading behavior as an isolated event. Recently, graph neural networks (GNNs) have been leveraged for financial fraud detection and achieved remarkable success [5,23], as GNNs model historical inter-connected trading behavior in a graph-based perspective, which could better mine the grouped conspiracy patterns. For example, RTG-Trans [13] constructs the trading graph by treating transactions as nodes and their relation as edges, and then utilizes graph convolution to learn latent representations for spoofing detection. GTAN [25] leverages the graph attention mechanism on the temporal transaction graphs and achieves superior performance in fraud detection.

However, existing graph neural network approaches mostly learn the node and edge features by aggregation-based operation [27], which face significant challenges in the *real-world spoofing detection problem* because they could only represent the *local* context of a suspicious transaction. While an increasing num-

---

[1] https://www.cftc.gov/PressRoom/PressReleases/8260-20.

ber of traders are organized like enterprises to conduct conspiracy spoofing to make illegal profits from well-behaved participants, especially innocent small- and medium-sized investors. Figure 1 illustrates a typical example of group-based spoofing transaction, showing how to push up the stock price through organized fake orders and then sell real orders to make profits. First, multiple collaborative traders submit a large number of buy orders at a price higher than the current market price, gradually pushing up the price of the target asset. Then the master trader submits a real sell order at the desired price. After the real order is completed, the collaborative trader immediately cancels the previous buy orders. The market price will gradually return to the previous price level. Through the spoofing transaction, the master trader sells the asset at a higher price and makes an excess gain. The characteristic of group-based spoofing transaction is that multiple traders collude with each other and present an imperceptible and disguised trend. Each trader's behavior appears to be normal when monitored individually, so detecting group-based transaction is very challenging. In contrast, a wider *global* context in the group perception of the transaction graph could significantly benefit the model to better capture organized patterns, which is also proved by pioneer researchers [6]. But we still face significant challenges on how to adaptively incorporate both the *local* and *global* context in organized spoofing transaction detection given the cunning nature of conspiracy traders who can swiftly update spoofing patterns.

Therefore, in this paper, we propose a novel Group Perceptual-Enhanced GNN approach, named GPEGNN, for spoofing transaction detection. In particular, we first construct the raw trading data into transaction graphs. Then, we devise a multi-layer graph attention, self-attention, and multilayer perceptron network that represents the topological and attribute information to model the *local* context of the nodes and edges in the graph. Next, we design group enhancement layers to accommodate nodes with similar transaction characteristics and aggregate them into groups to learn the *global* context of organized behavior patterns. These *local* and *global* features are jointly optimized within the detection network. The experimental results strongly demonstrates the superior performance of our methods compared with state-of-the-art baselines. In a nutshell, the main contributions of this paper include:

- To the best of our knowledge, this is the first graph neural network approach that addresses the spoofing detection problem by jointly learning from the *local* nodes/edges features and the *global* grouped graph features.
- We devise group enhancement layers to locate nodes with similar patterns and aggregate them into groups to learn the *global* context of organized behavior patterns, and a neural network to model the *local* entity features. They are jointly optimized in an end-to-end framework to better detect the constantly updated spoofing patterns.
- We evaluated the proposed approach on a real-world spoofing detection scenarios. The state-of-the-art performance of the evaluation highlights the potential of our method to provide valuable and constructive evidence to

financial regulators by effectively bridging the gap between spoofing detection industrial practice and graph learning theory.

2 Related Works

2.1 Spoofing Detection

Spoofing poses a critical and real-world threat to the stability and integrity of the financial market and has attracted widespread attention. Early research mainly used statistics-based methods [3], trying to identify potential spoofing by analyzing the statistical characteristics of transaction behavior. Subsequently, machine learning and deep learning techniques [4] have been utilized for detecting spoofing behavior, such as recurrent neural networks [1], attention-based neural network [23]. Recently, graph neural networks have demonstrated exceptional performance in modeling inter-connected behavior and achieved notable success across various domains [6], which is also applied in financial fraud detection. For example, RTG-Trans [13] employs a deep graph learning-based transformer to capture relationship features and temporal features for spoofing detection jointly. Although existing graph neural network-based methods for spoofing detection improve performance, they still face significant challenges in spoofing transaction detection because most GNN-based methods learn features with adjacent nodes' *local* context information which will inevitably lead to suboptimal performance in extracting the conspiracy spoofing patterns without a *global* group feature of the transaction graphs. In this paper, we propose a group perceptual-enhanced GNN-based approach that could jointly learn the *local* and *global* context of transaction graph in an end-to-end framework and could subsequently benefit to address the spoofing detection problem.

2.2 Graph Learning in Finance

With the rapid advancement of artificial intelligence technology, graph-based machine learning has garnered significant attention [23]. This technology has been widely applied in fields such as image processing, natural language processing and knowledge graphs, proving highly effective and reliable in finance as well [8,21]. For example, SemiGNN [23] introduces an innovative semi-supervised attention map neural network, leveraging both labeled and unlabeled data for fraud detection. This method enhances fraud detection models by incorporating a semi-supervised learning framework to utilize the rich information in unlabeled data. Wang et al. [22] proposes an adaptive heterogeneous multi-view graph learning method to address the small sample problem in credit risk prediction for small and medium-sized enterprises. To enhance GNN-based fraud detectors against fraudsters' feature and relationship camouflage, Dou et al. [6] introduces a label-aware similarity measure and a similarity-aware neighbor selector using reinforcement learning. For the sequence-based fraud detection problem, Gadbench [19] presents a graph abnormal detection framework to model user behavior sequences, enhancing performance and providing interpretable prediction results. Although existing spoofing detection methods perform well, they do not fully explore the relationships between transactions. Therefore, we introduce a novel approach which enables simultaneous learning of both *local* and *global* contexts within transaction graphs in an end-to-end framework.

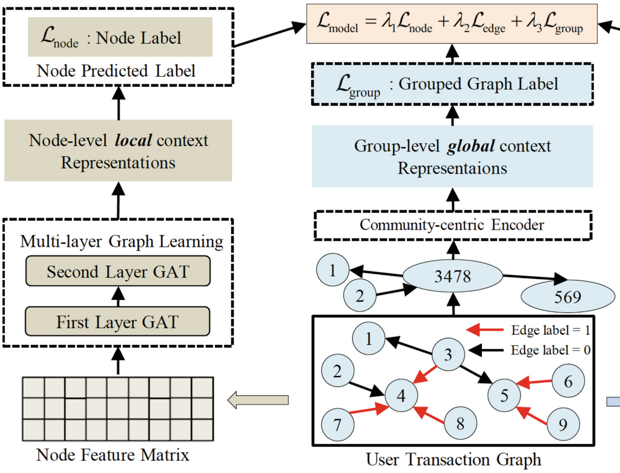

$\mathcal{L}_{node}$: Node Label
Node Predicted Label
$\mathcal{L}_{model} = \lambda_1\mathcal{L}_{node} + \lambda_2\mathcal{L}_{edge} + \lambda_3\mathcal{L}_{group}$
$\mathcal{L}_{edge}$: Edge label
Edge Predicted Label
$\mathcal{L}_{group}$: Grouped Graph Label
Node-level local context Representations
Group-level global context Representaions
Edge-level local context Representations
Multi-layer Graph Learning
Second Layer GAT
First Layer GAT
Community-centric Encoder
Attentional Feature Learning
Self-Attention Layer
MLP Layer
1
2
3478
569
Edge label = 1
Edge label = 0
3
4
5
6
7
8
9
Node Feature Matrix
User Transaction Graph
Edge Feature Matrix

Fig. 2. The model architecture of the proposed Group Perceptual-Enhanced Graph Neural Network (GPEGNN). We first construct the transaction graph from the raw trading data. Then, we devise a multi-layer attention module to learn the node-level *local* context representation, and an attentional feature learning module to learn the edge-level *local* context representation. Next, we propose a community-centric encoder to capture the group-enhanced *global* context features. These *local* and *global* features are jointly optimized and trained in an end-to-end framework.

3 The Proposed Method

3.1 Model Architecture Overview

In this section, we introduce the architecture of our proposed model in detail. As illustrated in Fig. 2, our GPEGNN primarily comprises four components: transaction graph construction, node and edge representation learning, group-enhanced representation layer, and prediction network layer. We devise multi-layer graph learning and attention feature learning module for node-level and edge-level *local* representation learning. In particular, we convert raw trading records into transaction graphs, generate node features from edge features through graph representation learning, then learn node representations using graph attention layers, and finally put them into a fully connected layer to generate node classification results. In the group-enhanced representation layer, we generate new graph aggregation process and train neural networks to achieve group-level *global* features. Given that spoofing usually occurs in gangs, we use a group aggregation strategy to merge the nodes that are inferred as spoofing by the encoder into groups, then obtain aggregated group user transaction graph, and finally fed it into the community-centric encoder to learn the group representation. The prediction network adopts a comprehensive optimization strategy to optimize the *local* and *global* representations by integrating node classification loss, transaction classification loss, and group detection loss.

3.2 Transaction Graph Construction

Given the transaction data of all users, we first construct the transaction graph $G = (V, E)$, where $V = \{v_1, v_2, \cdots, v_n\}$ represents transaction nodes, $E = \{e_1, e_2, \cdots, e_m\}$ represents the relations between transaction nodes, n is the number of transactions, and m is the number of sequences between two transactions. If two transactions share the same trader with time period t_1 or two transactions are placed within time period t_2 ($t_1 > t_2$), we create an edge between two nodes. Because there must be lots of transaction placing and transaction cancellation operations in the process of spoofing, and these operations all occur in a short period, we connect these transactions in a short period in the order of transaction placement. We use the order attributes, transaction attributes, and order book features of the transaction to represent node v_i, and D is the dimension of node features. For any transaction v_i, if it is associated with spoofing, the node is considered a negative sample; otherwise, it is a positive sample. We denote the adjacency matrix of the graph as $A = (a_{ij})_{n*n}$, where a_{ij} equals 1 if node i and node j are connected, and 0 otherwise.

3.3 Node and Edge Representation Learning

After the transaction graph is constructed, we need to characterize the nodes and edges in the graph. To fully utilize the relationship information within the graph network, we utilize deep graph representation learning techniques to construct the attribute matrix X for nodes and the attribute vector $Embed_{v_i}$ for individual nodes. Specifically, X captures comprehensive node features, while $Embed_{v_i}$ encodes specific attributes for each node. These are elaborated as follows:

$$\begin{aligned} Embed_{v_i} &= \frac{1}{|M_i|} \sum_{j \in M_i} Embed_{e_j} \\ \hat{y_i} &= \frac{1}{|M_i|} \sum_{j \in M_i} y_j \end{aligned} \tag{1}$$

where M_i denotes the index of the edge connected to node v_i, $Embed_{e_j}$ represents the feature vector of the edge, y_j is the true label of the j-th transaction, and $\hat{y_j}$ is the predicted label of a potential spoofing transaction in the model training. During this process, the node v_i's attribute vector is aggregated based on its connected edges, consolidating relevant information from its neighboring nodes. The label assigned to the node v_i signifies whether it is involved in spoofing activities.

To learn the representations of the nodes in the transaction graph, we use the graph attention layer (GAT) to encode the graph network in the Multi-layer Graph Learning. We construct a two-layer GAT with the following formula:

$$\begin{aligned} e_{ij}^{(l)} &= a(W^{(l)}h_i^{(l-1)}, W^{(l)}h_j^{(l-1)}) \\ \alpha_{ij}^{(l)} &= \frac{\exp(e_{ij}^{(l)}))}{\sum_{k \in N_i} \exp(e_{ik}^{(l)})} \\ h_i^{(l)} &= \sigma(\textstyle\sum_{j \in N_i} \alpha_{ij}^{(l)} W^{(l)} h_j^{(l-1)}) \\ X^{(l)} &= \alpha^{(l-1)}(\alpha^{(l-2)} X^{(l-1)} H^{(l-2)}) H^{(l-1)} \end{aligned} \tag{2}$$

where $W^{(l)}$ is the weight matrix of the l-th layer, $h_i^{(l)}$ represents the node i's representation in the l-th layer, N_i is the set of neighboring nodes of i, a and σ are the attention function and the LeakyReLU activation function. $\alpha^{(l)}$ is the l-th GAT layer's attention matrix, $X^{(l)}$ is the l-th GAT layer's input node attribute matrix, and $H^{(l)}$ are the parameters to be learned. After the l layers of the graph neural network, we obtain $X^{(l)}$, which represents the node classification results and the probability of node spoofing. We then use a multi-layer perceptron (MLP) network as a classifier, which can be represented as:

$$X^{(l+1)} = sigmoid(MLP(X^{(l)})) \tag{3}$$

where MLP is a one-layer fully connected network followed by the sigmoid activation function. $X^{(l+1)}$ represents the binary classification outcomes for all nodes, indicating the likelihood of a node engaging in spoofing transactions. Based on $X^{(l+1)}$, nodes are classified into two categories, and cross-entropy is used to compute the node classification loss:

$$\mathcal{L}_{\text{node}} = \frac{1}{n} \sum_{i=1}^{n} \bar{y_i} \log(\bar{p_i}) + (1 - \bar{y_i}) \log(1 - \bar{p_i}) \tag{4}$$

where $\bar{p_i}$ represents the node v_i's predicted label, $\bar{y_i}$ represents the node v_i's actual label.

Once the transaction graphs are constructed, we procee to learn the representation of edges. When there exists an edge e_k connecting nodes v_i and v_j, we merge the representations $Embed_{v_i}$, $Embed_{v_j}$, and $Embed_{e_k}$ to refresh e_k's representation with the following procedure:

$$Embed_{e_k'} = [Embed_{v_i} || Embed_{v_j} || Embed_{e_k}] \tag{5}$$

We then feed the updated representation into a one-layer MLP and self-attention layer to infer the edge-level features in the Attentional Feature Learning. Finally, the edge loss is defined as:

$$\mathcal{L}_{\text{edge}} = \frac{1}{m} \sum_{k=1}^{m} y_k \log(p_k) + (1 - y_k) \log(1 - p_k) \tag{6}$$

where p_k and y_k represent the prediction and ground-truth label of the k-th edge (if one of the connected nodes of edge e_k is labeled as a spoofing transaction, we market the k-th edge as a spoofing relation), respectively.

3.4 Group-Enhanced Representation Layer

In the real-world financial markets, spoofing transactions are normally organized in groups, which can be better captured in group-level *global* feature learning in transaction graphs. We model the grouped behavior in a graph aggregating process. In particular, for two connected nodes v_i and v_j, if one of which is a transaction canceled after buying/selling and the other is a completed transaction, we label the edge of these two nodes as a suspicious spoofing edge. Then, we aggregate the connected sub-graphs into a grouped super node by these suspicious spoofing edges. Subsequently, we obtain a new aggregated transaction graph, denoted as $G' = (V', E')$. Recall that if two transactions share the same trader with time period t_1 or two transactions are placed within time period t_2 ($t_1 > t_2$), we create an edge between two nodes. By controlling the t_1 and t_2, the transaction relations are well modeled by a graph so that the transaction graphs could support both the *local* context (in a graph G) and *global* context (in aggregated G') feature learning.

Given an aggregated graph G', let $V' = \{v'_1, v'_2, \cdots, v'_{n'}\}$ denote the node group, where each v'_i can be either an aggregated hyper-vertex or a standalone vertex not affiliated with any group. Here, n' signifies the total number of aggregated hyper-vertex. We utilize M' to encode the relationships between the aggregated gangs and individual participators. The M'_i encompasses all indices of vertices within community v'_i. To learn the vertices feature value of X' of G', we aggregate node features within each group by performing element-wise summation. Specifically, the feature of the group v'_i is defined as the average of embedded features of nodes within v'_i, represented as $\frac{1}{|M'_i|} \sum_{j \in M'_i} Embed_{v_j}$, and X' comprises the learned features of n' aggregated vertices. Subsequently, X' and G' are fed into the community-centered encoder, comprising two GAT layers and a single-layer MLP. Analogous to the Multi-layer Graph Learning approach, $X'^{(2)}$ undergoes processing through the GAT layers to yield $X'^{(3)}$ via the one-layer MLP.

Afterward, we then employ the group-level loss function for the spoofing detection task, which is denoted as $\mathcal{L}_{\text{group}}$. The loss function aimed at refining the learning of group representations. Differing from $\mathcal{L}_{\text{node}}$ for node classification and $\mathcal{L}_{\text{edge}}$ for edge classification, $\mathcal{L}_{\text{group}}$ is specifically designed to improve the identification of organized spoofing within transactions, and its formulation is outlined as follows:

$$\mathcal{L}_{\text{group}} = \frac{1}{n'} \sum_{i=1}^{n'} \hat{y_i} log(\hat{p_i}) + (1 - \hat{y_i}) log(1 - \hat{p_i}) \tag{7}$$

where $\hat{y_i}$ is the true value of aggregated vertex $\hat{v_i}$, $\hat{p_i}$ is the model prediction result of group $\hat{v_i}$, $\hat{y_i}$ is 1 if group $\hat{v_i}$ is a hyper vertex (aggregated nodes), otherwise $\hat{y_i}$ is 0. We then train the model via joint optimization, the loss function $\mathcal{L}_{\text{group}}$ aims to effectively detect organized spoofing transactions.

3.5 Prediction Network Layer

Finally, we combine the loss $\mathcal{L}_{\text{node}}$, the loss $\mathcal{L}_{\text{edge}}$ and the grouped representation loss $\mathcal{L}_{\text{group}}$ into get the model's final loss L_{model}:

$$\mathcal{L}_{\text{model}} = \lambda_1 \mathcal{L}_{\text{node}} + \lambda_2 \mathcal{L}_{\text{edge}} + \lambda_3 \mathcal{L}_{\text{group}} \tag{8}$$

where λ_1, λ_2 and λ_3 are hyper parameters which subject to $\sum_{i=1}^{3} \lambda_i = 1$ and are determined by cross validation. By combining $\mathcal{L}_{\text{node}}$, $\mathcal{L}_{\text{edge}}$, and $\mathcal{L}_{\text{group}}$, the detection network can more effectively detect organized spoofing. During model training, we optimize using the standard stochastic gradient descent (SGD) algorithm and employ the Adam optimizer for parameter learning.

4 Experiments

4.1 Datasets

We collect a real-world dataset from a commodity exchange, comprising 50,000 trading record slices from January 2018 to December 2023. We use 80% of the data for training and 20% for testing. This dataset includes order data, order book data, and transaction data, with a total of 38 feature dimensions. Order data captures traders' intentions to execute, buy, sell or cancel orders for specific assets at particular prices. Transaction data shows the successful matching of buy and sell orders, and the order book data is the uncompleted buy and sell orders in the queue, including their price and quantity. The details of the node features and edge features are shown in Table 1. Since spoofing detection is a binary classification task, we evaluate model performance using Recall, F1 score, Accuracy, Precision, and Area Under the Curve (AUC).

4.2 Baseline Models

To compare, we utilize both widely-used industry-level approaches and the latest graph learning methods to assess the effectiveness of our proposed method. The industry-level methods and graph learning baselines are outlined below:

- Logistic Regression (LR) [7]: a linear model that makes predictions by modeling the relationship between input features and categories as probabilities.
- Random Forest (RF) [29]: an ensemble learning algorithm composed of multiple decision trees that improve the prediction performance of a single tree through voting or averaging.

Table 1. The details of the feature description.

	Feature Description
Node	User ID, buy flag, sell flag, buying time, selling time, buying order ID, selling order ID, transaction completed flag, transaction price/volume/amount, open price, current price, current position, current buying/selling volume, Opening to the current highest/lowest/average price, Up/Down limit price, best buying/Selling price, best buying/Selling volume
Edge	User ID of node 1 and node 2, Transaction completed flag of node 1 and node 2, Order placing time of node 1 and node 2, Order canceling time of node 1 and node 2, Time interval between placing and canceling of node 1 and node 2, Order placing time interval between node 1 and node 2, Order cancellation time interval between node 1 and node 2, Whether the connected users are the same user

- Adaboost [12]: an ensemble learning technique that iteratively trains a sequence of weak classifiers and assigns weights to their predictions.
- Gradient Boosting Decision Tree (GBDT) [28]: an ensemble learning algorithm that iteratively trains a decision tree model and each iteration attempts to correct the residuals of the previous model.
- Hybrid Multi-layer perceptron (HMLP) [9]: a neural network that integrates diverse neuron types to enhance model representation and performance.
- GCN [24]: a graph convolutional network designed specifically for handling graph-structured data, spreading information among nodes via convolution operations to accomplish tasks like node classification and link prediction.
- GraphSAGE [8]: a graph neural network that can perform representation learning on nodes by sampling and aggregating local neighborhood features of nodes.
- GAT [21]: a graph neural network that dynamically learns the relationship weights between nodes by introducing an attention mechanism, thereby achieving node-level representation learning and task prediction.
- EigenGCN [14]: a graph convolutional network that uses the pooling operator to design the pooling layer, and combines it with the traditional GCN convolution layer to build a graph neural network framework for graph classification.
- RetaGNN [11]: a pioneering graph neural network built on relational temporal attention, capable of classification and prediction devoid of reliance on content and auxiliary data.

4.3 Experimental Parameter Settings

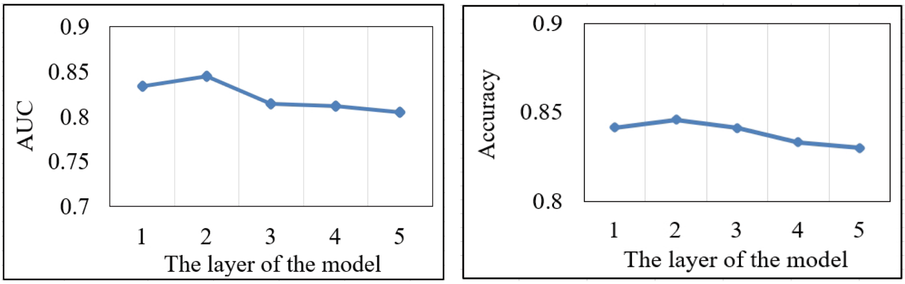
We implement our model in PyTorch framework with an NVIDIA Tesla V100-32GB GPU. During model training, we set the parameter C to 0.4 for LR. For RF, Adaboost, and GBDT, we adjust the parameter *n_estimators* to 150, 100, and 100, respectively. In the HMLP model, we configure two layers with hidden layer dimensions set to 40 and 20. For all graph-based models, we set the time window t_1 to 30 s and t_2 to 3 s. The number of layers l is set to 2, each containing 30 hidden layer units and the ReLU activation function, while the activation function of the last layer is sigmoid. In our proposed GPEGNN, the learning rate is set to 0.001, the batch size to 64, and λ_1, λ_2, and λ_3 to 0.3, 0.3, and 0.4, respectively. The parameters of all comparative baseline models are chosen via cross-validation, and some hyperparameters of the neural network models determined empirically.

Table 2. The spoofing detection performance comparison with all baselines.

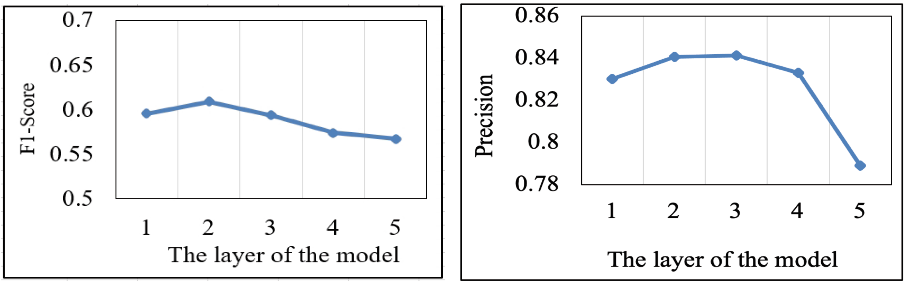
Method	F1-Score	Accuracy	Precision	Recall	AUC
LR	0.5209	0.7656	0.5680	0.4810	0.6746
RF	0.4630	0.8203	0.4440	0.4837	0.6841
Adaboost	0.5187	0.7382	0.4645	0.4513	0.6787
GBDT	0.5205	0.8038	0.7105	0.4106	0.6760
HMLP	0.5459	0.8282	0.8231	0.4084	0.6893
GCN	0.5542	0.8311	0.8226	0.4178	0.7184
GraphSAGE	0.5715	0.8337	0.8104	0.4414	0.7754
GAT	0.5744	0.8348	0.8145	0.4436	0.7438
EigenGCN	0.5719	0.8352	0.8239	0.4379	0.7465
RateGNN	0.5685	0.8345	0.8241	0.4340	0.8101
GPEGNN	**0.6088**	**0.8459**	**0.8408**	**0.4772**	**0.8449**

4.4 Spoofing Detection Experiment

We evaluated the performance of different models in the spoofing detection task, and the detail experimental results are shown in Table 2. The table clearly shows that our proposed model GPEGNN outperforms all baselines, including both the state-of-the-art graph-based models and the traditional industry-level models. For both the AUC and F1 results, our results improve over the second best model (RateGNN) by more than 3%. Among all baseline models, the industry-level conventional machine learning methods (LR, RF, Adaboost, and GBDT) could not achieve satisfactory results. This is because the conspiracy spoofing behavior patterns are too complicated for a traditional machine learning model to address, which also strongly demonstrates the motivation of our proposed model that employs a graph learning model to better extract organized features. As we can see, graph-based models (GCN, GraphSAGE, GAT, EigenGCN, RetaGNN, and GPEGNN) are significantly better than other methods because graph learning can extract information about relationships between trading orders, and these learned relationship information could contribute to better spoofing detection. Among all baselines, GPEGNN is the best model even compared with the state-of-the-art graph-based models because our model not only learns the node-level and edge-level *local* relationships between orders, but also designs a community-centered encoder for *global* context learning on the transaction graph to better extract the organized criminal activities. The result of spoofing detection experiments strongly demonstrates the effectiveness of our proposed method.



(a) AUC (b) Accuracy

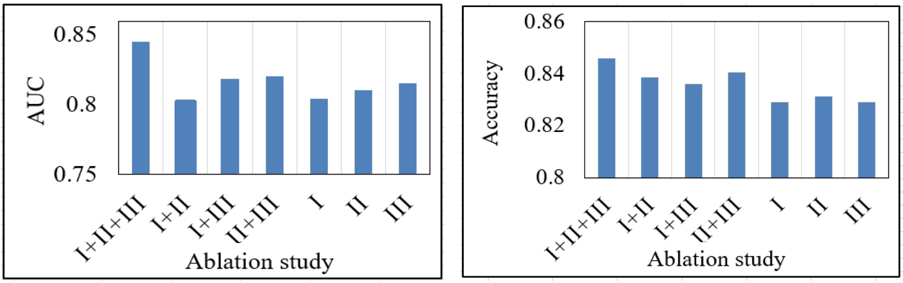


(c) F1 (d) Precision

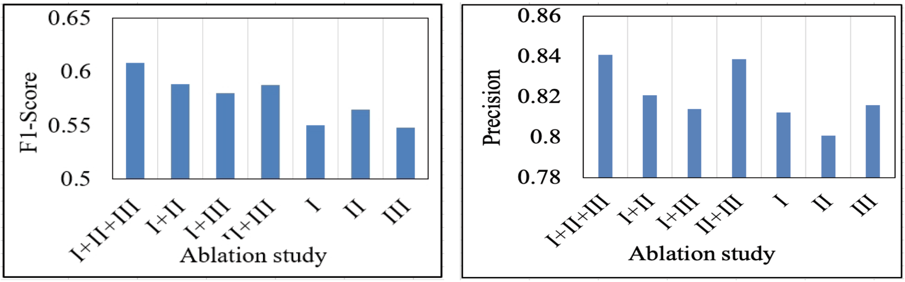
Fig. 3. The results of parameter sensitivity experiments. We vary the number of graph learning layers from 1 to 5 and the result results suggest that two layers of graph feature learning achieves the best performance.

4.5 Parameter Sensitivity Experiment

To deeply analyze the stability of our model across different parameter settings, we study the impact of the number of GCN layers in the hyperparameters of the proposed model (GPEGNN) on the overall performance. We set the number of layers in the model to 1, 2, 3, 4, and 5, and then compare the experimental results of AUC, Accuracy, F1-Score, and Precision. The details are reported in Fig. 3(a), 3(b), 3(c) and 3(d), respectively. As we can see, when the number of layers is 2, the model shows the best performance, indicating that a moderate number of layers can achieve a good balance between the complexity and representation capabilities of the model. On the contrary, when the number of layers is 1, the model does not learn enough relationship information between features, resulting in performance degradation. When the number of layers increases to 5, the deepening hierarchical structure may lead to overfitting or gradient vanishing, thus gradually diminishing the effectiveness. The increase in network depth results in excessive parameters, making it prone to overfitting. Additionally, gradient vanishing makes it difficult for the network to learn meaningful features, impacting both the model's generalization ability and training effectiveness.



(a) AUC (b) Accuracy



(c) F1 (d) Precision

Fig. 4. The ablation study experimental results. We divide the model into three modules, I (Node-level *local* context Representations), II (Edge-level *local* context Representations), and III (Group-level *global* context Representations). The result demonstrates the effectiveness of our proposed model in joint learning both the *local* and *global* context representations.

4.6 Ablation Study

To verify whether each module of the model is effective, we also performed ablation experiments. According to the functional design and purpose of each module in the model, We divide the model into three modules, I (Node-level local context Representations), II (Edge-level local context Representations), and III (Group-level global context Representations). Then we make different combinations of the three modules to generate different models, evaluate the effect of each module by observing the performance of different models. The experimental results are shown in Fig. 4(a) to 4(d). In the figure, model(I) means that only module I is used, I+II means that both module I and module II are used, and I+II+III means that all modules are utilized, which is also the proposed GPEGNN. These figures vividly demonstrates the repercussions of removing individual modules from the model, showcasing varying degrees of performance degradation. This emphasizes the pivotal role played by each module in bolstering spoof detection capabilities. Notably, module III exhibits a more substantial impact compared to module II, with module II exerting a greater influence than module I. Such a hierarchical effect underscores the critical importance of integrating group-level *global* context representations into the realm of spoofing detection. The overall performance of our approach and ablation study illustrate the effectiveness of our proposed method. This validation further solidifies the efficacy and robustness of our proposed methodology, affirming its capability to effectively discern and combat fraudulent activities within complex transactional networks.

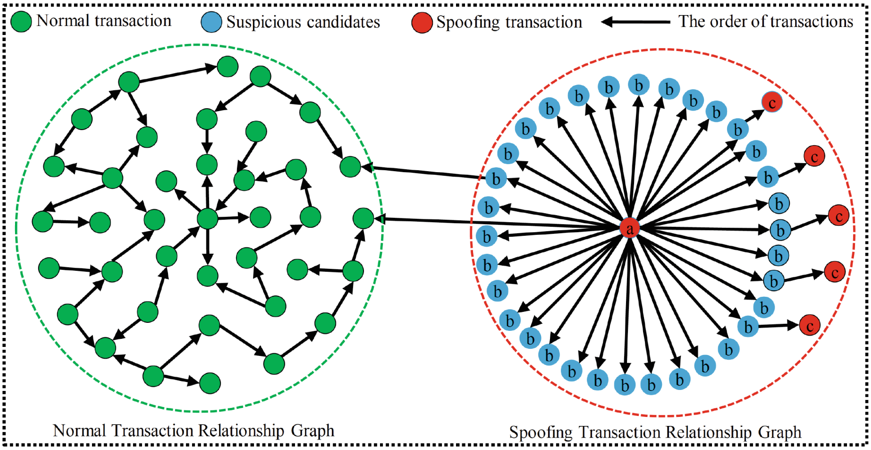

Normal transaction
Suspicious candidates
Spoofing transaction
The order of transactions
a
b
c
Normal Transaction Relationship Graph
Spoofing Transaction Relationship Graph

Fig. 5. The case studies of spoofing transactions in the real-world industry level application scenarios. The green, blue, and red nodes are observed as normal, suspicious candidates and spoofing transactions, respectively. (Color figure online)

4.7 Case Study in Industry-Level Application Scenarios

In this section, we conduct case studies when our proposed model is deployed in the industry-level application scenarios of a commodity exchange. Firstly, we convert the futures trading records into trading graphs, where each node represents a transaction, and the edges between nodes represent the relations of transactions, as shown in Fig. 5. The green nodes represent the most normal transactions, the blue nodes represent suspicious candidate transactions that are quickly canceled after placing a transaction, the red nodes represent any spoofing transactions, and the black edges represent the relations between any two transactions. Spoofing traders usually use different trading accounts to place a large number of transactions in a short time and then quickly cancel the transactions. For example, after the completion of transaction a in the picture, a large number of transactions and quick cancellations of transactions are carried out in a short time (b). During the transaction cancellation period, this causes the price to rise and fall instantly. At this time, the spoofing transaction c in the figure is performed to obtain profits. As depicted in Fig. 5, all these transactions are associated with a group. Utilizing the model proposed in this paper enhances the detection of such group spoofing instances, validating the effectiveness of our approach in detecting organized spoofing.

5 Conclusion

To tackle the challenging and real-world spoofing problem in the financial market, we proposed a novel group perceptual-enhanced deep graph learning approach for spoofing detection for better extracting swiftly updated conspiracy patterns. In particular, we devise a multi-layer graph attention and self-attention for *local* context learning, and a community-centric encoder for *global* context learning on the transaction graphs, which are jointly optimized in the detection network. The proposed method is deployed and evaluated industry-level application scenarios in a commodity exchange. The outcome demonstrates GPEGNN's superiority over existing state-of-the-art baselines. Additionally, the enhanced performance of our proposed method exposes the vulnerabilities of organized spoofing traders, contributing to the robustness of the financial trading ecosystem.

References

1. Bandyopadhyay, S.K., Dutta, S.: Detection of fraud transactions using recurrent neural network during COVID-19: fraud transaction during COVID-19. J. Adv. Res. Med. Sci. Technol. **7**, 16–21 (2020)
2. Boulkenafet, Z., Komulainen, J., Hadid, A.: Face spoofing detection using colour texture analysis. IEEE Trans. Inf. Forensics Secur. **11**, 1818–1830 (2016)
3. Cao, Y., Li, Y., Coleman, S., Belatreche, A., McGinnity, T.M.: Adaptive hidden markov model with anomaly states for price manipulation detection. IEEE Trans. Neural Networks Learn. Syst. **26**, 318–330 (2014)

4. Cartea, Á., Jaimungal, S., Wang, Y.: Spoofing and price manipulation in order-driven markets. Appl. Math. Finance **27**, 67–98 (2020)
5. Cheng, D., Wang, X., Zhang, Y., Zhang, L.: Graph neural network for fraud detection via spatial-temporal attention. IEEE Trans. Knowl. Data Eng. **34**, 3800–3813 (2020)
6. Dou, Y., Liu, Z., Sun, L., Deng, Y., Peng, H., Yu, P.S.: Enhancing graph neural network-based fraud detectors against camouflaged fraudsters. In: ACM International Conference on Information and Knowledge Management, pp. 315–324 (2020)
7. Gong, J., Sun, S.: A new approach of stock price prediction based on logistic regression model. In: National Industrial Security System, pp. 1366–1371 (2009)
8. Hamilton, W., Ying, Z., Leskovec, J.: Inductive representation learning on large graphs. In: Conference on Neural Information Processing Systems, pp. 1–19 (2017)
9. Heidari, A.A., Faris, H., Aljarah, I., Mirjalili, S.: An efficient hybrid multilayer perceptron neural network with grasshopper optimization. Soft Computing **23**(17), 7941–7958 (2019)
10. Hilal, W., Gadsden, S.A., Yawney, J.: Financial fraud: a review of anomaly detection techniques and recent advances. Expert Syst. Appl. **193**, 116429 (2022)
11. Hsu, C., Li, C.T.: RetaGNN: relational temporal attentive graph neural networks for holistic sequential recommendationn. In: International World Wide Web Conference, pp. 2968–2979 (2021)
12. Huang, X., Li, Z., Jin, Y., Zhang, W.: Fair-adaboost: extending adaboost method to achieve fair classification. Expert Syst. Appl. **202**, 117240 (2022)
13. Kang, L., Mu, T.J., Ning, X.: Conspiracy spoofing orders detection with transformer-based deep graph learning. In: International Conference on Advanced Data Mining and Applications, pp. 489–503 (2023)
14. Ma, Y., Wang, S., Aggarwal, C.C., Tang, J.: Graph convolutional networks with eigenpooling. In: Special Interest Group on Knowledge Discovery and Data Mining, pp. 723–731 (2019)
15. MacKenzie, D.: Spoofing: law, materiality and boundary work in futures trading. Econ. Soc. **5**, 1–22 (2022)
16. Mendonça, L., De Genaro, A.: Detection and analysis of occurrences of spoofing in the brazilian capital market. J. Finan. Regul. Compliance **28**, 369–408 (2020)
17. Olychyk, A.: A spoof of justice: double jeopardy implications for convictions of both spoofing and commodities fraud for the same transaction. Am. Univ. Law Rev. **65**, 239–274 (2015)
18. Psiaki, M.L., Humphreys, T.E.: GNSS spoofing and detection. Proc. IEEE **104**, 1258–1270 (2016)
19. Tang, J., Hua, F., Gao, Z., Zhao, P., Li, J.: Gadbench: revisiting and benchmarking supervised graph anomaly detection. In: Advances in Neural Information Processing Systems, pp. 1–26 (2024)
20. Tao, X., Day, A., Ling, L., Drapeau, S.: On detecting spoofing strategies in high-frequency trading. Quant. Finance **22**, 1405–1425 (2022)
21. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y.: Graph attention networks. In: International Conference on Learning Representations, pp. 1–12 (2017)
22. Wang, C., Yu, F., Zhang, Z., Zhang, J.: Multiview graph learning for small-and medium-sized enterprises' credit risk assessment in supply chain finance. Complexity **2021**, 6670873 (2021)
23. Wang, D., et al.: A semi-supervised graph attentive network for financial fraud detection. In: IEEE International Conference on Data Mining, pp. 598–607 (2019)

24. Wu, F., Souza, A., Zhang, T., Fifty, C., Yu, T., Weinberger, K.: Simplifying graph convolutional networks. In: International Conference on Machine Learning, pp. 6861–6871 (2019)
25. Xiang, S., et al.: Semi-supervised credit card fraud detection via attribute-driven graph representation. In: Association for the Advancement of Artificial Intelligence, pp. 14557–14565 (2023)
26. Yu, J., et al.: Group-based fraud detection network on e-commerce platforms. In: ACM SIGKDD Conference on Knowledge Discovery and Data Mining. pp. 5463–5475 (2023)
27. Zang, Y., et al.: Don't ignore alienation and marginalization: correlating fraud detection. In: International Joint Conference on Artificial Intelligence, pp. 4959–4966 (2023)
28. Zhang, T., He, W., Zheng, H., Cui, Y., Song, H., Fu, S.: Satellite-based ground PM2. 5 estimation using a gradient boosting decision tree. Chemosphere 268, 128801 (2021)
29. Zhang, W., Wu, C., Zhong, H., Li, Y., Wang, L.: Prediction of undrained shear strength using extreme gradient boosting and random forest based on bayesian optimization. Geosci. Front. **12**, 469–477 (2021)

Self-SLAM: A Self-supervised Learning Based Annotation Method to Reduce Labeling Overhead

Alfiya M. Shaikh[1(✉)], Hrithik Nambiar[1], Kshitish Ghate[2],
Swarnali Banik[1], Sougata Sen[1], Surjya Ghosh[1],
Vaskar Raychoudhury[3], Niloy Ganguly[4], and Snehanshu Saha[1]

[1] Computer Science and Information Systems and APPCAIR, BITS Pilani K K Birla Goa Campus, Zuarinagar, India
`{2023proj027,f20190100g,p20210016,sougatas,surjyag,snehanshus}@goa.bits-pilani.ac.in`
[2] Language Technologies Institute, Carnegie Mellon University, Pittsburgh, PA 15213, USA
`kghate@andrew.cmu.edu`
[3] Computer Science and Software Engineering, Miami University, Oxford, OH 45056, USA
`raychov@miamioh.edu`
[4] Computer Science and Engineering, IIT Kharagpur, Kharagpur, India
`niloy@cse.iitkgp.ac.in`

Abstract. In recent times, Deep Neural Networks (DNNs) have been effectively used to tackle various tasks such as emotion recognition, activity detection, disease prediction, and surface classification. However, a major challenge in developing models for these tasks requires a large amount of labeled data for accurate predictions. The manual annotation process for a large dataset is expensive, time-consuming, and error-prone. Thus, we present SSLAM (Self-supervised Learning-based Annotation Method) framework to tackle this challenge. SSLAM is a self-supervised deep learning framework designed to generate labels while minimizing the overhead associated with tabular data annotation. SSLAM learns valuable representations from unlabeled data that are applied to the downstream task of label generation by utilizing two pretext tasks with a novel $log - cosh$ loss function. SSLAM outperforms supervised learning and Value Imputation and Mask Estimation (VIME) baselines on two datasets - Continuously Annotated Signals of Emotion (CASE) and wheelchair dataset. The wheelchair dataset is our novel unique surface classification dataset collected using wheelchairs showcasing our framework's effectiveness in real-world scenarios. All these results reinforce that SSLAM significantly reduces the labeling overhead, especially when there is a vast amount of unlabeled data compared to labeled data. The code for this paper can be viewed at the following link: https://github.com/Alfiya-M-H-Shaikh/SSLAM.git

---

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-70378-2_8.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
A. Bifet et al. (Eds.): ECML PKDD 2024, LNAI 14949, pp. 123–140, 2024.
https://doi.org/10.1007/978-3-031-70378-2_8

Keywords: Self-supervised Learning · Annotation Overhead · *log − cosh* loss function · Limited Labelled Data

1 Introduction

Recently, Deep Neural Networks (DNN) have been found effective in different domains including healthcare, activity recognition, surface classification, and human behavior understanding (e.g., emotion recognition, elderly monitoring) [7,15,22]. These systems collect data from the physiological signals and IMU (Inertial Measurement Unit) sensor and then employ a DNN model for the intended task. However, a major challenge in achieving optimal performance by utilizing DNN models is the requirement for a substantial volume of labeled data as the manual annotation process is fatigue-inducing, error-prone, and time-consuming [1,9,12]. At present time, we are surrounded by a large number of pervasive devices (e.g., smartphones, smartwatches, IoT devices) that generate a lot of data; a majority of which remains unlabeled. Consequently, despite the abundance of data, we are unable to fully leverage the potential of this extensive dataset due to the substantial overhead involved in annotation. Hence, the development of efficient strategies for annotating large volumes of data is essential.

In this paper, we aim to address the problem of automatic annotation of a large volume of continuous sensor data streams for socially relevant problems such as detecting wheelchair-accessible path characteristics from the built environment using smartphone-embedded motion sensors. Wheelchair users while undertaking their daily activities, will move through various built surfaces, such as concrete sidewalks, asphalt, granite tiles, cobblestones, etc. in the outdoor environment and carpet, linoleum, mosaic, etc. in the indoor environment. We captured the vibration generated by different surfaces through the accelerometer and gyroscope sensors in the user's smartphone and then used a specialized AI framework to classify the surfaces based on their characteristic vibration patterns. The data collection process for this unique dataset has been extensively documented in our previous work [25]. Often sidewalks are not accessible by wheelchair users depicted in Fig. 1 due to obstacles such as broken/uneven surfaces, steep slopes, high-pile slippery cobblestones (with deep gaps in between) as well as sidewalks with no access ramps. E.g., cobblestones are recognized as grossly inaccessible while concrete sidewalks are considered accessible. However, this problem is a challenging one given the numerous different types of surfaces available in different countries as well as the different types of wheelchairs used by the people. Several wheelchair-related parameters (such as manual or power, tire material, weight, number of wheels, height from the ground at which the smartphone is attached, etc.) are responsible for producing different vibration data streams for the same surface type. Moreover, the user's body weight, height, and disability type can also impact the nature of vibration. Overall, it is non-trivial to manually annotate the different types of data collected in this project across 6 different countries on 3 different continents from 48 different surfaces using 50 wheelchair users on 6 different manual and power wheelchairs.

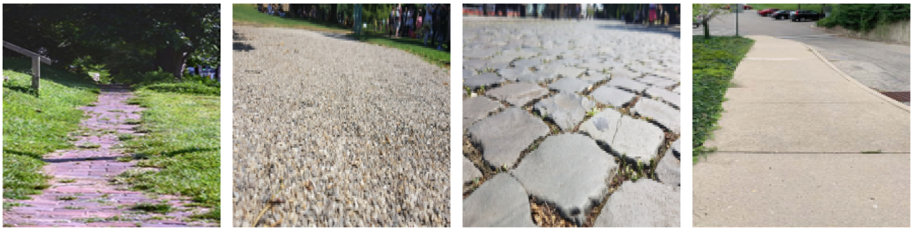


Fig. 1. Non-accessible sidewalks; surface classification required

Various annotation strategies are proposed in the existing literature. First, self-report or expert-driven techniques are utilized wherein the signal fragments are annotated by an (or a group of) experts [23], and distinct unification approaches (e.g., majority voting [13]) are applied to come up with a single rating (or label). For example, the dataset named CASE (*Continuously Annotated Signals of Emotion*) [19] involved participants who used a joystick to provide continuous annotations of their emotions, specifically *valence* and *arousal*, based on the Circumplex Model of emotion [16]. These approaches demand significant user effort and are not easily adaptable to larger scales. The second approach to annotate the signal used an auxiliary modality from a given modality [3]. In this paper, signals from an IMU sensor are annotated automatically, leveraging the availability of acoustic data. However, the dependency on another modality is the major drawback of these approaches. The third approach uses a human-in-the-loop annotation strategy that includes the concept of Active Learning. For example, in [9,18], a human annotator is included in the loop who recognizes a group of seed samples (with available annotations) to train a base model, which gives outcome for all the remaining unlabeled instances. Next, the outcome from the model is considered depending on the model's confidence, or the human expert is conferred for the annotation. The key challenges include seed instance identification, involvement of human experts, and lack of clarity (by the human expert) in understanding the problem encountered by the learner [9,18].

However, we can design an intelligent annotation approach leveraging the apriori knowledge from the domain experts and the *intrinsic properties* of the dataset clusters to reduce human engagement significantly. Thus, we propose the Self-SLAM (SSLAM) annotation framework to label datasets with minimal expert intervention. The framework constitutes a self-supervised algorithm that employs two pretext tasks developed using a contrastive sampling method [24]. We employ pretext tasks to train the encoder in a self-supervised manner, optimize the resultant representations using a parameterized activation function, and then apply a label-noise resilient *log-cosh* loss function for reconstruction. Though this function is similar in structure to the standard loss functions like Mean Squared Error (MSE) and Mean Absolute Error (MAE), it has a desirable analytical property called Lipschitzness that helps to deal with the label noise. This makes the proposed framework robust, and essential to ensure label quality.

The performance of SSLAM was evaluated for two different use cases: (a) emotion annotation and (b) surface classification. First, we evaluated label generation performance to annotate emotion continually on publicly available continuous emotion CASE [19] dataset. The dataset consists of continuous valence-arousal annotations of emotional and physiological responses measured through multiple sensors. SSLAM provides more accurate valence and arousal predictions than a supervised approach leveraging unlabeled data and minimal labeled samples. It outperforms another self-supervised learning framework (VIME) [24] on the same number of labeled and unlabeled data by 20.8% and 17.7% (for valence and arousal, respectively).

We evaluated SSLAM on a subset of our surface vibration dataset collected from wheelchair users. This dataset includes manual wheelchair-induced vibration data from 47 participants across 15 distinct indoor and outdoor surfaces in the USA and China. In this dataset, SSLAM outperforms a supervised learner and VIME by 4.25% and 7.9%, respectively, with the same amount of labeled and unlabeled data. In summary, our paper demonstrates that SSLAM outperforms classical machine learning algorithms such as Logistic Regression, Multilayer Perceptron (MLP), and XGBoost, and a self-supervised learning algorithm (VIME). In summary, our paper's key contributions are:

- We proposed a self-supervised framework SSLAM to significantly reduce annotation overhead and demonstrate improvements over the existing baselines using a parameterized Elliot activation function and a new loss function.
- We collected and shared a novel and unique wheelchair-induced surface vibration dataset that enriches the available resources and facilitates further research.
- We present a new reconstruction loss called $log-cosh$ in the SSLAM encoder setup, provide an explanation of its suitability as a viable alternative to MSE loss, and highlight its implications of being robust to label noise and outliers and its relevance to the SSLAM framework.
- We provide empirical evidence on both wheelchair and publicly available CASE datasets to demonstrate that the proposed method is applicable for different use cases such as surface classification and continuous emotion annotation respectively.

2 Dataset Description

2.1 Wheelchair

As described in Sect. 1, the wheelchair dataset is a collection of surface-induced vibration data caused by the movement of manual wheelchairs in both built and natural environments. Data is collected using an Android smartphone attached tightly to the handrest of a collapsible manual wheelchair. When participants self-propelled the wheelchair across various surfaces, the accelerometer and gyroscope sensors capture the vibration at a sampling rate of 100 Hz. We collected

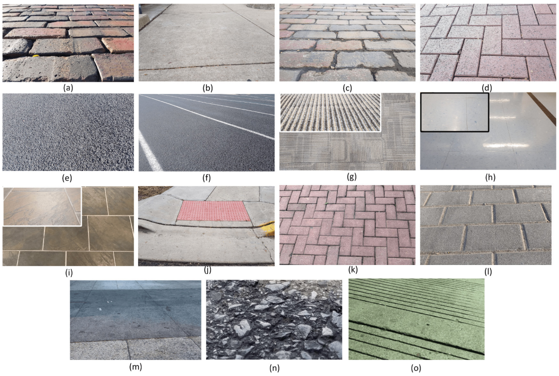


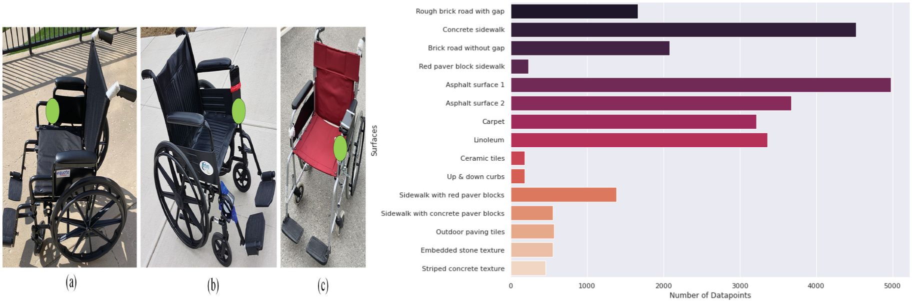
Fig. 2. Surfaces used for data collection: **in the USA:** (a) Rough brick road with gap, (b) Concrete sidewalk, (c) Brick road without gap, (d) Red paver block sidewalk, (e) Asphalt surface 1, (f) Asphalt surface 2, (g) Carpet, (h) Linoleum, (i) Ceramic tiles, (j) Up & down curbs **in China:**, (k) Sidewalk with red paver blocks, (l) sidewalk with concrete paver blocks, (m) Outdoor paving tiles, (n) Embedded stone texture, (o) Striped concrete texture (Color figure online)

data from 16 different surfaces in the USA and China as depicted in Fig. 2. Our data collection involved 42 participants and 2 wheelchairs in the USA and 5 participants and 1 wheelchair in China. The manual wheelchairs used in the USA and China for data collection are presented in Fig. 3.1.

The final clean dataset includes 22 time-domain features representing vibrational and gyroscopic data. Overall, we have collected 27,000 data points that can be used for further analyses of surface classification. This dataset includes 15 surface types/classes, of which, 3 classes have a relatively less number of data points as displayed in Fig. 3.2, making the classification task challenging. Also, since the dataset is manually annotated, there is a possibility that the dataset contains some amount of label noise.

2.2 CASE

The Continuously Annotated Signals of Emotion (CASE) [19] dataset contains continuous emotion annotations provided by the participants while watching various videos. This dataset also includes participant's recorded physiological reactions to the videos. These physiological measurements were synchronized and sampled at 1000 Hz from Electrocardiograph (ECG), Blood Volume Pulse (BVP), Galvanic Skin Response (GSR), Respiration (RSP), Skin Temperature (SKT), and Electromyography (EMG) sensors. This dataset is based on the 2D circumplex model of emotion that depicts different valence and arousal levels on the coordinate X-axis and Y-axis respectively. The participants used a novel Joystick-based Emotion Reporting Interface (JERI) on this 2D plane to report annotations sampled at 20 Hz. The participants included 15 males and 15 females aged between 22 and 37 from different cultural backgrounds.



(1) (2)

Fig. 3. (1) Chairs (a) and (b) were used in the USA, and (c) in China. Green dots indicate where the smartphone was attached. (2) Wheelchair dataset class distribution. (Color figure online)

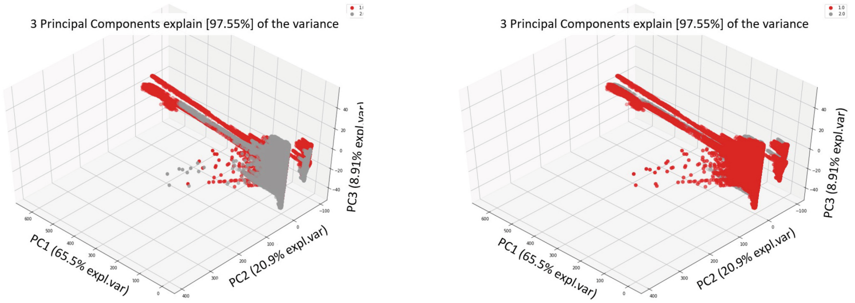


Fig. 4. CASE dataset distribution based on classes 1 and 2, representing low and high levels of (a) arousal and (b) valence respectively.

Our final dataset consists of 8 real-valued features corresponding to the physiological reactions of the participants and has two classes valence and arousal with low (≤ 5) and high (> 5) levels. Also, we have converted the raw annotation scores to low and high valence and arousal values such that they map to one of the four quadrants of the circumplex plane [16]. Though the dataset contains outliers as shown in Fig. 4, we demonstrate our method to be robust to label noise and outliers.

3 SSLAM: Self-supervised Label Generation Framework

Our proposed framework incorporates a novel activation function and loss function as an improvement over the current state-of-the-art self-supervised framework for tabular data (VIME) [24]. We employ two pretext tasks that are, feature vector estimation and mask vector estimation to train an encoder in a self-supervised manner as shown in Fig. 5. These tasks employ two predictors using the input vector's encoder representations. The task of the first predictor model is to recover the original input feature vector from its corrupted variant produced using a mask vector. The task of the second predictor is to predict the mask vector. The pretext tasks are solved using the below models,

(i) Mask vector estimator, $s_m : \mathcal{Z} \rightarrow [0,1]^d$, takes the encoder embedding $\mathbf{z}$ as input and predicts a mask vector $\hat{\mathbf{m}}$.

(ii) Feature vector estimator, $s_r : \mathcal{Z} \rightarrow \mathcal{X}$, takes the encoder embedding $\mathbf{z}$ as input and predicts $\hat{\mathbf{x}}$ for the input feature vector $\mathbf{x}$.

Mask vector estimation task uses a mask vector generator to produce a binary mask vector $\mathbf{m} = [m_1, \ldots, m_d]^\top \in \{0,1\}^d$ where m_i is randomly sampled from a Bernoulli distribution with a probability p_{mask}. The pretext generator $g_m : \mathcal{X} \times \{0,1\}^d \rightarrow \mathcal{X}$ utilizes a mask vector $\mathbf{m}$ and samples $\mathbf{x}$ from the large unlabeled dataset $\mathcal{D}_u$ as input, and generates a corrupted sample $\tilde{\mathbf{x}}$. The corrupted feature is given by, $\tilde{\mathbf{x}} = g_m(\mathbf{x}, \mathbf{m}) = \mathbf{m} \odot \overline{\mathbf{x}} + (1 - \mathbf{m}) \odot \mathbf{x}$ where the j-th feature of $\overline{\mathbf{x}}$ is sampled from the empirical distribution $\hat{p}_{X_j} = \frac{1}{N_u} \sum_{i=N_l+1}^{N_l+N_u} \delta\left(x_j = x_{i,j}\right)$. The pretext generator g_m is also a stochastic function whose randomness comes from $\overline{\mathbf{x}}$. Together this randomness makes reconstructing $\mathbf{x}$ from $\tilde{\mathbf{x}}$ a difficult task for the neural networks. The following optimization problem, $\min_{e, s_m, s_r} \mathbb{E}_{\mathbf{x} \sim p_X, \mathbf{m} \sim p_{\mathbf{m}}, \tilde{\mathbf{x}} \sim g_m(\mathbf{x}, \mathbf{m})} \left[l_m(\mathbf{m}, \hat{\mathbf{m}}) + \alpha \cdot l_r(\mathbf{x}, \hat{\mathbf{x}}) \right]$ where $\hat{\mathbf{m}} = (s_m \circ e)(\tilde{\mathbf{x}})$ and $\hat{\mathbf{x}} = (s_r \circ e)(\tilde{\mathbf{x}})$, is used to train the encoder e and the pretext predictive models.

$l_m(\mathbf{m}, \hat{\mathbf{m}}) = -\frac{1}{d} \Big[\sum_{j=1}^{d} m_j \log \Big[(s_m \circ e)_j (\tilde{\mathbf{x}}) \Big] + (1 - m_j \Big) \log \Big[1 - (s_m \circ e)_j (\tilde{\mathbf{x}}) \Big]$ is the first loss function which is the sum of the binary cross-entropy losses for each dimension of the mask vector. The second loss function l_r is the proposed novel $log - cosh$ reconstruction loss, $l_r(\mathbf{x}, \hat{\mathbf{x}}) = \frac{1}{d} \left[\sum_{j=1}^{d} logcosh \left(x_j - (s_r \circ e)_j (\tilde{\mathbf{x}}) \right) \right]$. We propose a parameterized version of the Elliot activation function to be used in the hidden layers of the encoder to yield a better representation, which is computed as follows: $f\left(\omega_j^{\text{in}}, \beta_j, x_i, \lambda\right) = k_1 + \frac{k_2 \cdot \left(\omega_j^{\text{in}} \cdot x_i + \beta_j\right) \cdot \lambda}{1 + \left| \left(\omega_j^{\text{in}} \cdot x_i + \beta_j\right) \cdot \lambda \right|}$ where λ is the slope of the function and k_1 and k_2 are the parameters learned through back-propagation during the training of this network.

The encoder here is a neural network that maps the input data to a fixed-length vector representation. The multilayer neural network used by SSLAM encoder framework has one hidden layer with a novel activation function. The difficulty of the pretext tasks can easily be controlled through the multiple hyper-parameters of the framework such as the probability p_{mask} can be tuned to adjust the proportion of the corrupted features. The hyper-parameter α is also tuned to weigh the loss from the two pretext tasks. VIME [24] has proposed the optimal values for these parameters using cross-validation. Due to the way the encoder has been trained, the representations z contain information about imputing corrupt features and identifying the corrupted features. This informative representation of the input data, reduces model complexity to minimize the losses in comparison to the raw input feature data, resulting in more accurate predictions.

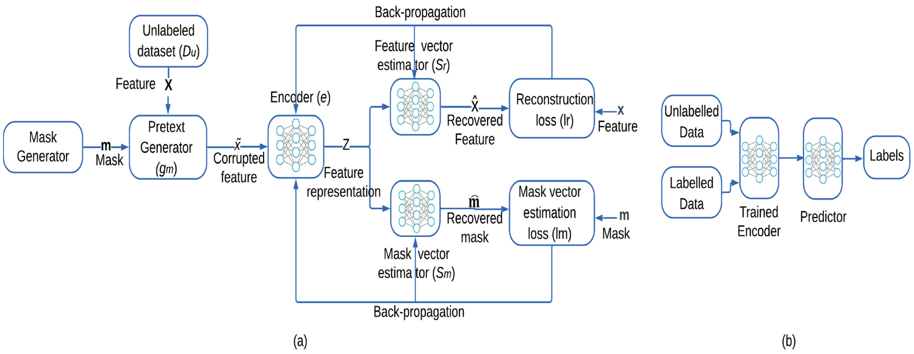


Fig. 5. (a) Block diagram illustrating the SSLAM framework. (b) Generating labels employing a trained encoder and predictor network.

The availability of labeled and unlabeled data is the primary consideration for our proposed framework. We elaborate on the explanation of our framework using the CASE dataset. The CASE dataset has 1.5 million instances of annotated emotion data for valence and arousal classes. Significant expenses are associated with annotating these data points, which we aim to reduce using our label-generation framework. A large proportion of unlabeled data is required for our method, thus, for our study, we split our dataset in the ratio of 1:9 labeled and unlabeled data points. The SSLAM framework utilizes these data in the following way: the encoder takes unlabeled data as input and converts it into informative homogeneous representations. It is then trained to minimize the cross-entropy and reconstruction loss functions associated with the mask and feature vector estimation tasks respectively. To adequately recover the input features x, we require the encoder to output latent representation z. To achieve this, the correlation between the input features of x needs to be captured. This is exactly what the encoder does. s_m can utilize the inconsistencies between feature values to identify the masked features, while s_r can learn from the correlated non-masked features to attribute the masked features. The encoder, therefore, learns that if a particular feature has a different correlation from the others, it may be masked and corrupted.

This information is useful for the next downstream task of transforming the remaining labeled data points into better homogeneous and informative representations. These transformed representations are then fed into the predictive model, to better predict the class labels of the input test data. We, thus apply this framework to our split of unlabeled and labeled data to generate new labeled data points. These artificially generated labels can be added back to the original labeled set and the process can be iterated to produce more annotated data.

To summarise, our data goes through the following steps in the framework:

- Acquire labeled and unlabeled data points where the proportion of unlabeled data points is considerably larger.

- The encoder is fed with unlabeled data points to learn better representations of the data by solving two pretext tasks.
- Post training, the encoder is fed with labeled data to generate a homogeneous and informative representation for further downstream classification tasks.
- The encoder representations train a predictive model using the labeled data.
- The learned representations are then utilized to predict new class labels on test data.
- This newly generated labeled data can be mixed with the original labeled dataset and the process can be iterated over to produce more labels.

3.1 Log-Cosh Loss in SSLAM Framework

Mathematical Framework: We now justify the proposal of Loss, $L(x) = log(cosh(x))$ in an encoder setup. Using $log - cosh$ as the reconstruction loss in the encoder setup is supported by additional analytical properties, such as convexity, smoothness, robustness to outliers, etc. Let $L(x)$ be the loss function with x being the input to the loss function. Then for the symmetric version of the loss function, $L(x) = log(cosh(x))$.

MAE and MSE as Siblings: The expression for the loss function is as follows: $E(x, y) = \sum_{i=1}^{m} log(cosh(y_j - w^T x_j))$ for training examples (x_j, y_j) for $j = 1$ to m, where y_j is the actual value of the jth training example from the dataset. Using Taylor Series approximation it can be shown that $E(x, y) = \sum_{i=1}^{m} log(cosh(y_j - w^T x_j))$ is mathematically equivalent to Mean Absolute Error (MAE) and Mean Squared Error (MSE) respectively for large x away from 0 or for small x nearer to 0. Since, MSE is the preferred reconstruction loss in VIME, we show the impact of the proposed loss function in comparison to the current SoTA, VIME. Since, we know that MAE is 1- Lipschitz [11]. For large x, our loss function behaves like MAE, thus we can argue that like MAE, $log - cosh$ is robust to outliers. Our loss function therefore inherits identical robustness to label noise as MAE. For small $x, log - cosh(x)$ inherits properties of MSE. Consequently, our proposed $log - cosh$ combines the smoothness of MSE and the robustness of MAE, making itself highly suitable for machine learning applications such as the self-supervised approach proposed here. This establishes $log - cosh$ as a suitable alternative for MSE ((similar to VIME) in the encoder setup of the SSLAM framework.

How did the Loss Function Come About? The preceding discussion establishes the $log - cosh$ function and explains its effectiveness in various contexts. However, it does not necessarily prove its relevance as a reconstruction loss in the encoder setup. The primary objective of deep learning is to gain knowledge about the manifold structure present in the data (i.e. natural high dimensional data that converges to a non-linear low dimensional manifold). It also involves understanding the probability distribution associated with the manifold. An encoder learns low dimensional data and represents data as a parametric manifold i.e. a piece-wise linear map from latent to the ambient space.

Logcosh(x) in VAE - A Distributional Insight: We define the encoder and decoder as:

- Encoder $\varphi : \chi \rightarrow \mathcal{F}$ maps Σ to its latent representation $D = \varphi(\Sigma)$ homeomorphically.
- Decoder ψ: $\mathcal{F} \rightarrow \varphi$ maps z to reconstruction $\tilde{x} = \psi(z) = \psi \circ \varphi(x)$

$\varphi \circ \psi = argmin_{\varphi,\psi} \int_{\chi} L(x, \psi \circ \varphi(x))\, dx$, where χ is the ambient space, $\mathcal{F}$ is the latent space, L is the loss function and Σ is a topological space $\Sigma \subset \bigcup_{\alpha} U_{\alpha}$. We constructed distributions using pseudo-hyperbolic Gaussian, resulting in the reconstruction loss for Variational AutoEncoders (VAEs), defined as $logcosh(x)$, serving as our loss function.

Pseudo-Hyperbolic Gaussian: The strategy to generate the pseudo-hyperbolic Gaussian ((Wrapped gaussian distribution $G(\mu, \Sigma)$ on hyperbolic space $\mathbf{H}$)) is as follows:

- Sample a $\boldsymbol{v}$ from normal distribution N(0,Σ) defined over $\mathbf{R}^n$.
- Interpret $\boldsymbol{v}$ as an element of $T_{\mu}\mathbf{H}^n \subset \mathbf{R}^{n+1}$ by rewriting $\boldsymbol{v}$ as v=[0,$\boldsymbol{v}$].
- Parallel transport vector v to $u \in T_{\mu}\mathbf{H}^n \subset \mathbf{R}^{n+1}$ along the geodesic from μ_0 to μ.
- Map u to $\mathbf{H}^n$ using $exp(u) = cosh(||u||_L) + sinh(||u||_L)\frac{u}{||u||_L}$

Reconstruction Loss is thus $-\mathbf{E}_{q_{z|x}} log(p_{\theta}(x|z))$. Replacing $p_{\theta}(x|z)$ with pdf of Hyperbolic secant distribution: $= -log(\frac{1}{2}sech(\frac{\pi x}{2})) = log(2cosh(\frac{\pi x}{2})) = log(cosh(y))$ where $y = \frac{\pi x}{2}$. Since the metric at the tangent space coincides with the Euclidean metric, several distributions can be produced by applying the construction strategy such as $logcosh(x)$.

4 Evaluation

4.1 Experimental Configuration

To evaluate the performance of SSLAM, we test it on two tabular datasets from the domain of affective computing: CASE and wheelchair. For all our experiments, we randomly divide our dataset into an - (a) 85-15% and (b) 80-20% train-test split. Later, we split our training data into 10-90% labeled and unlabeled data. We evaluate our proposed model against four baseline models, three of which were used as baselines in VIME. SSLAM is different from a classical supervised classification problem and therefore most of the SOTA baselines don't apply to the setting proposed here. The first baseline model is a simple MLP trained using only labeled data in a supervised manner. Our second baseline is a simple logistic regression model. XGBoost, a tree-based classification method is our third baseline. Our final baseline is a self-supervised model VIME [24] with state-of-the-art performance results for classification tasks in the tabular domain. The self-supervised models are pre-trained on the unlabeled data and used along with the labeled data for classification tasks.

We have used the same encoder architecture in SSLAM as in VIME as depicted in Table 1. Based on the experiments conducted in VIME we have

Table 1. Architecture details of SSLAM

Module	Layer Details	Layer Dimensions
Input	-	8 × 1
Encoder	[dense] × 1 + Parameterized Elliot	(8,8)
Feature vector estimator	[dense] × 1 + Linear	(8,8)
Mask vector estimator	[dense] × 1 + Sigmoid	(8,8)
Predictor	[dense] × 1 + ReLu [dense] × 4 + ReLu [dense] × 1 + Sigmoid	(8,100) (100,100) (100,2)
Output	-	2

tuned the model parameters p_{mask} and α to 0.3 and 2 respectively. The activation function corresponding to the feature vector estimation is set to linear activation function whereas the layer corresponding to mask estimation has a sigmoid activation function. The reconstruction loss and mask estimation loss are the novel $log - cosh$ and binary cross-entropy losses respectively. The encoder is trained using an RMSprop optimizer with a learning rate of 0.001 on both loss functions. Our baseline Multilayer Perceptron (MLP) and feedforward neural network (FNN) used in the predictor network has five hidden layers each with hidden dimension 100. These hidden layers are set to have a ReLU activation function while the output layer has a Softmax activation function. Both are trained using an Adam optimizer with a learning rate of 0.001 on the categorical cross-entropy loss function. The supervised feedforward neural model is fine-tuned with early stopping (patience 50), and we allocate 10% of the training data as the validation split. All models are trained with a batch size of 128. We train the feedforward neural predictor network for 100 epochs and the encoder for 10 epochs. To enhance our model's performance, we utilized a parameterized Elliot activation function in the encoder.

Fine-Tuning Hyperparameters: We have performed several experiments with varying the hyperparameters of the predictor FNN on both datasets. First, we conducted experiments to vary the dimensions of the hidden layers in the FNN to 100 and noted its performance for the classification tasks as shown in Appendix A.1. We found that for both datasets used in this study 100 neurons in the hidden layer is the best choice. To avoid incurring high computational costs, we do not exceed this number.

Next, for each dataset, we experiment with the number of hidden layers in FNN. Figure in Appendix A.1 and Table 2 indicate that 5 is the optimal choice for both CASE and wheelchair datasets. Again, we restrict our experiments to 5 layers. Ultimately, this study aims to illustrate the comparative performance of the proposed model over the baselines on both datasets, which is achieved by

Table 2. Performance comparison for different number of layers for each of the datasets

Datasets	Number of Hidden Layers				
	1	2	3	4	5
CASE: Valence	77.2329	82.1172	86.0457	87.5545	**90.0178**
CASE: Arousal	76.8658	82.1246	85.4126	87.2266	**89.7985**
Wheelchair	61.6047	64.7465	65.8745	66.1679	**71.6529**

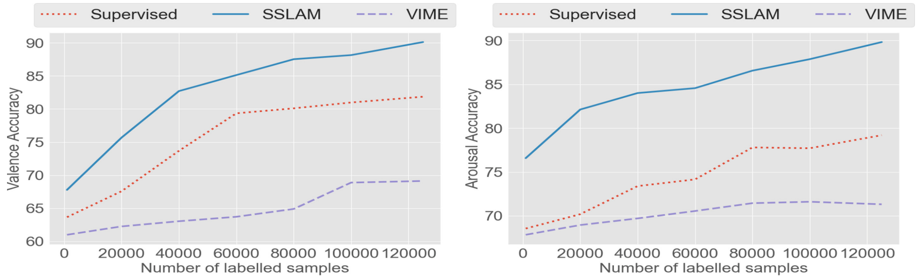


Fig. 6. Comparison of Accuracy of predictions of (a)Valence and (b)Arousal across different sizes of labeled CASE dataset made by SSLAM and other baselines.

our experimental setup. We have executed each of the models 5 times utilizing different random train/validation/test splits and seeds and the average of these results has been reported. As with previous studies on tabular data, we use accuracy as our evaluation metric in all experiments.

4.2 Results: CASE Dataset

In this section, we will be discussing the results of the SSLAM on the CASE dataset, and we will also be comparing its performance with the baselines that we had defined earlier. The classification results of Valence and Arousal accuracy on 85-15% train-test split are indicated in Table 3. The MLP baseline produces 81.3% accuracy for Valence, which is better than logistic regression, XGBoost and VIME. But, our proposed framework vastly outperforms these baselines, producing an accuracy of 90.01%. Also, SSLAM outperforms all of these baselines including VIME for Arousal by generating an accuracy of 89.79%. We obtain similar results on the 80-20% split as displayed in Table 3 with SSLAM outperforming all baselines on both Valence (89.49%) and Arousal (88.84%). In Fig. 6, we compare the performance of (valence and arousal prediction accuracy) of a supervised MLP, VIME and SSLAM against the increasing number of labeled data points (x-axis). The proposed approach (SSLAM) outperforms both the baselines, i.e. the supervised MLP and the self-supervised VIME on the CASE dataset.

Table 3. Comparison of Accuracy for predicting Valence and Arousal on CASE dataset using the 85-15% and 80-20% splits

Model Type	Accuracy using 85-15% split		Accuracy using 80-20% split	
	Valence	Arousal	Valence	Arousal
MLP	0.8130 ± 0.0040	0.7993 ± 0.0013	0.8034 ± 0.0021	0.7911 ± 0.0057
Logistic Regression	0.6901 ± 0.0012	0.6806 ± 0.0010	0.6877 ± 0.0028	0.6899 ± 0.0025
XGBoost	0.7330 ± 0.0009	0.7339 ± 0.0022	0.8467 ± 0.0041	0.7343 ± 0.0015
VIME	0.6917 ± 0.0051	0.7213 ± 0.0042	0.7197 ± 0.0027	0.7093 ± 0.0027
SSLAM	**0.9001 ± 0.0024**	**0.8979 ± 0.0045**	**0.8949 ± 0.0046**	**0.8884 ± 0.0073**

Table 4. Comparison of Accuracy for predicting Valence and Arousal on Wheelchair dataset using the 85-15% and 80-20% splits

Model Type	85-15% split		80-20% split	
	Accuracy	F1 score	Accuracy	F1 score
MLP	0.6740 ± 0.0211	0.6704 ± 0.0054	0.6347 ± 0.0294	0.6309 ± 0.0231
Logistic Regression	0.4463 ± 0.0020	0.4063 ± 0.0012	0.4307 ± 0.0054	0.4237 ± 0.0073
XGBoost	0.6305 ± 0.0089	0.6304 ± 0.0029	0.6216 ± 0.0019	0.6193 ± 0.0147
VIME	0.6366 ± 0.0393	0.6065 ± 0.0381	0.6283 ± 0.0317	0.6267 ± 0.0318
SSLAM	**0.7165 ± 0.0054**	**0.7120 ± 0.0067**	**0.7074 ± 0.0059**	**0.7040 ± 0.0093**

4.3 Results on Wheelchair Dataset

We have used the same experimental setup for the Wheelchair dataset analysis as for the CASE dataset. The comparison outcomes of our model with the baselines on the wheelchair data for the 85-15% and 80-20% splits are presented in Table 4. We observe that the SSLAM model outperforms all other baselines for both train-test splits. SSLAM achieves an accuracy of 71.65% on the 85-15% split, while on the 80-20% split, it achieves an accuracy of 70.74%. The wheelchair dataset has unbalanced classes; thus, we also report the weighted F1 score. From the above two tables, it is clear that the F1 score represents a similar trend where SSLAM outperforms the other two baselines.

In the case of the wheelchair dataset, all the models have performed poorly due to the limited size of the dataset. For all our models we use the same amount of labeled and unlabeled data. Results from the CASE dataset demonstrate that as we increase the number of unlabeled samples, the encoder's representations improve thus resulting in optimal classification accuracy. The wheelchair dataset is relatively small in size, which means that the number of unlabeled samples ($\leq$ 100,000) is not enough to help the encoder generalize well and learn good representations of the inputs. As a result, the performance of the models is not significantly improved. Additionally, the dataset's class imbalance problem further hinders the model's performance.

5 Discussion

We demonstrated that the SSLAM framework outperformed other baselines on both datasets. Our proposed methodology is best suited when dealing with large amounts of unlabeled data where annotating the data is tedious and expensive. This is often the case in real-world scenarios such as annotating surface-induced vibration data for wheelchair users and emotional data from physiological sensors. Thus, SSLAM can efficiently be employed to generate meaningful representations from the unlabeled samples and to generate labels reducing large annotation overhead.

The **Role of $log - cosh$ in SSLAM:** is evident from the performance of the framework in comparison to the standard MSE in the encoder set-up. The use of $log - cosh$ delivers significant improvements across both datasets - CASE and Wheelchair. The largest increase in performance has been observed in the CASE dataset. This dataset has an outlier problem and $log - cosh$ being robust to outliers overcomes this considerably.

6 Relevant Literature

We discuss our research literature in three parts: continuous emotion annotation techniques and their drawbacks, limited data annotation emotion recognition methods, and the effectiveness of different self-supervised approaches on tabular data.

Continuous Emotion Annotation: In the existing literature, the most widely adopted approach of emotion annotation using self-report is the post-interaction or post-stimuli one, where the participants after watching the video provide emotion self-reports based on a standard scale (e.g., Self-assessment Manikin (SAM) [2]). However, in the post-stimuli approach capturing intra-video subtle nuances and time-aligning all the emotions is challenging. To address these issues, researchers use continuous emotion annotation strategies, where participants continuously provide emotion annotations as they watch the videos using a mouse, a joystick or another similar device [5,8,28]. Similarly, the CASE [19] dataset involved participants who used a joystick to provide continuous annotations of their emotions, specifically *valence* and *arousal*, based on the Circumplex Model of emotion [16]. Yet, the challenges with these approaches are the following - (a) for emotion annotations they require the users to utilize an auxiliary device, (b) due to the continuous nature of emotion annotation and video consumption in parallel, the cognitive load increases and the viewing experience degrades.

Recognizing Emotions with Limited Data: Numerous studies in affective computing have attempted to tackle the issue of the restricted availability of labeled data. Chen et al. (2021) [4] proposed a CNN method to tackle the problem of limited samples and imbalanced datasets for emotion recognition on the DEAP dataset through a data augmentation algorithm called the Borderline-SMOTE. They achieved a performance of 97.47% and 97.76% on valence and arousal prediction tasks. Zhang et al. (2022) [29] address the issue of data scarcity in EEG data by proposing a data augmentation method called generative adversarial network-based self-supervised data augmentation (GANSER) to perform emotion recognition. Their model synthesizes simulated EEG signals that do not skew from the underlying data distribution, which helps to perform well on emotion classification tasks. SigRep [6] produces performances for arousal (76.3%) and valence (74.1%) accuracy through a contrastive learning-based self-supervised technique using the data obtained from wearable devices. Tianyi et al. (2020) [27] propose a correlation-based emotion recognition algorithm (CorrNet) that employs an autoencoder to perform automatic feature extraction of signals generated by wearables. The model proposed by Tang et al. (2017) [21] for valence and arousal emotion classification on SEED and DEAP datasets used a denoising autoencoder. Subramanian et al. (2018) [20] learn features from electrocardiogram (ECG) data using a Naive Bayes classifier and Support Vector Machine (SVM). Sarkar et al. (2022) [17] propose a self-supervised multi-task CNN framework to learn ECG representations using pretext tasks.

Self-supervised Learning on Tabular Data: Some recent approaches propose using self-supervised learning techniques that utilize existing unlabeled data to discover broad feature representations specific to the data. In computer vision [10,26] and language modeling tasks [14], these approaches have proven to be fairly successful due to the underlying spatial, syntactic or semantic structure of the image or language data. Regardless, these approaches are not very effective for tabular data and sparse literature exists on handling tabular data using these methods. Recent studies focus on solving pretext tasks. Yoon et al. [24] proposed a self-supervised framework called Value Imputation and Mask Estimation (VIME) which employs two pretext tasks to train an encoder. The pretext generator is fed a random binary mask and unlabeled tabular data samples. This setup results in unlabeled samples that are corrupted by the mask. Given the corrupted heterogeneous inputs to the encoder, it is trained to generate informative homogeneous representations. In this architecture, the encoder representation of the data is fed into the mask and feature estimators, which predict both the binary mask and the original uncorrupted input. These learned transformed representations are further provided to the predictive model to perform the main downstream task.

7 Conclusion and Future Works

We presented a framework SSLAM for self-supervised label generation for annotation overhead reduction. The framework trains an encoder in a self-supervised manner by implementing two pretext tasks using a contrastive sampling method. The structure of VIME inspires our approach, but we distinguish ourselves by employing a novel loss function ($log - cosh$) compared to the denoising autoencoder loss used in VIME in the pre-training phase. Also, in the pre-training phase, we employ the parameterized Elliot activation function in the encoder to generate better representations to ensure more accurate predictions. Since we present our model as an improvement over the VIME, we have employed the same baselines used in VIME. Also, we are comparing against VIME because it is *the* state-of-the-art method. The other SOTA methods are applicable on vision data (such as MixMatch and ReMixMatch). Therefore the efficacy of the proposed method is best compared with VIME.

We evaluated the framework to determine its effectiveness in reducing the continuous annotation overhead on two datasets: wheelchair and CASE. The framework showed better results compared to the state-of-the-art self-supervised approach and the supervised approach. We also observed that the framework can generalize across different use cases, as demonstrated in a large-scale surface classification dataset for wheelchair users. Additional experiments on KEmoCon, MNIST and Fashion-MNIST datasets produce similar SOTA results. Along with further theoretical considerations, we defer the additional details on the generalizability of SSLAM to future work.

In summary, our SSLAM provides superior performance over existing baselines in label generation, particularly when more unlabeled data is available. We attribute this improved performance to our novel reconstruction $log - cosh$ loss that is employed by the encoder. The study results demonstrate the approach's potential to reduce annotation overhead in scenarios with imbalanced labeled and unlabeled data.

Acknowledgments. Snehanshu Saha, Surjya Ghosh and Sougata Sen would like to thank the Anuradha and Prashanth Palakurthi Center for Artificial Intelligence Research (APPCAIR), SERB-DST (SUR/2022/001965) and SERB CRG-DST (CRG/2023/003210), Govt. of India for partially supporting the work. Swarnali Banik gratefully acknowledges the Chanakya Fellowship from AI4CPS Innovation Hub, IIT Kharagpur.

References

1. Abdel Hakim, A.E., Deabes, W.: Can people really do nothing? handling annotation gaps in adl sensor data. Algorithms **12**(10), 217 (2019)
2. Bradley, M.M., Lang, P.J.: Measuring emotion: the self-assessment manikin and the semantic differential. J. Behav. Ther. Exp. Psychiatry **25**(1), 49–59 (1994)
3. Chatterjee, S., Chakma, A., Gangopadhyay, A., Roy, N., Mitra, B., Chakraborty, S.: Laso: exploiting locomotive and acoustic signatures over the edge to annotate imu data for human activity recognition. In: Proceedings of the 2020 International Conference on Multimodal Interaction, pp. 333–342 (2020)
4. Chen, Y., Chang, R., Guo, J.: Effects of data augmentation method borderline-smote on emotion recognition of eeg signals based on convolutional neural network. IEEE Access **9** (2021)
5. Cowie, R., Douglas-Cowie, E., Savvidou, S., McMahon, E., Sawey, M., Schröder, M.: 'FEELTRACE': an instrument for recording perceived emotion in real time. In: ITRW Speech-Emotion (2000)
6. Dissanayake, V., Seneviratne, S., Rana, R., Wen, E., Kaluarachchi, T., Nanayakkara, S.: Sigrep: toward robust wearable emotion recognition with contrastive representation learning. IEEE Access **10**, 18105–18120 (2022)

7. Garcia-Ceja, E., Riegler, M., Nordgreen, T., Jakobsen, P., Oedegaard, K.J., Tørresen, J.: Mental health monitoring with multimodal sensing and machine learning: a survey. Pervasive Mob. Comput. **51**, 1–26 (2018)
8. Girard, J.M., Wright, A.G.: Darma: software for dual axis rating and media annotation. Behav. Res. Methods **50**(3), 902–909 (2018)
9. Hossain, H.S., Khan, M.A.A.H., Roy, N.: Active learning enabled activity recognition. Pervasive Mob. Comput. **38**, 312–330 (2017)
10. Jing, L., Tian, Y.: Self-supervised visual feature learning with deep neural networks: a survey. IEEE Trans. Pattern Anal. Mach. Intell. **43**(11) (2020)
11. Mediratta, I., Saha, S., Mathur, S.: Liparelu: arelu networks aided by lipschitz acceleration. In: 2021 International Joint Conference on Neural Networks (IJCNN), pp. 1–8. IEEE (2021)
12. Muralidharan, A., Gyongyi, Z., Chi, E.: Social annotations in web search. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 1085–1094 (2012)
13. Nowak, S., Rüger, S.: How reliable are annotations via crowdsourcing: a study about inter-annotator agreement for multi-label image annotation. In: Proceedings of the International Conference on Multimedia Information Retrieval, pp. 557–566 (2010)
14. Qiu, X., Sun, T., Xu, Y., Shao, Y., Dai, N., Huang, X.: Pre-trained models for natural language processing: a survey. SCIENCE CHINA Technol. Sci. **63**(10), 1872–1897 (2020)
15. Ronao, C.A., Cho, S.B.: Human activity recognition with smartphone sensors using deep learning neural networks. Expert Syst. Appl. **59**, 235–244 (2016)
16. Russell, J.A.: A circumplex model of affect. J. Pers. Soc. Psychol. **39**(6), 1161 (1980)
17. Sarkar, P., Etemad, A.: Self-supervised ecg representation learning for emotion recognition. IEEE Trans. Affective Comput. (2020)
18. Settles, B.: Active learning literature survey. Technical Report, University of Wisconsin-Madison Department of Computer Sciences (2009)
19. Sharma, K., Castellini, C., van den Broek, E.L., Albu-Schaeffer, A., Schwenker, F.: A dataset of continuous affect annotations and physiological signals for emotion analysis. Sci. Data **6**(1), 1–13 (2019)
20. Subramanian, R., Wache, J., Abadi, M.K., Vieriu, R.L., Winkler, S., Sebe, N.: Ascertain: emotion and personality recognition using commercial sensors. IEEE Trans. Affect. Comput. **9**(2), 147–160 (2016)
21. Tang, H., Liu, W., Zheng, W.L., Lu, B.L.: Multimodal emotion recognition using deep neural networks. In: International Conference on Neural Information Processing, pp. 811–819. Springer (2017)
22. Wang, Y., Nazir, S., Shafiq, M.: An overview on analyzing deep learning and transfer learning approaches for health monitoring. Computational and Mathematical Methods in Medicine **2021** (2021)
23. Yang, J., Fan, J., Wei, Z., Li, G., Liu, T., Du, X.: Cost-effective data annotation using game-based crowdsourcing. Proc. VLDB Endowment **12**(1), 57–70 (2018)
24. Yoon, J., Zhang, Y., Jordon, J., van der Schaar, M.: Vime: extending the success of self-and semi-supervised learning to tabular domain. Adv. Neural. Inf. Process. Syst. **33**, 11033–11043 (2020)
25. Yu, H., Raychoudhury, V., Saha, S., Edinger, J., Smith, R.O., Gani, M.O.: Automated surface classification system using vibration patterns-a case study with wheelchairs. IEEE Trans. Artif. Intell. **4**(4), 884–895 (2023). https://doi.org/10.1109/TAI.2022.3190828

26. Zhai, X., Oliver, A., Kolesnikov, A., Beyer, L.: S4l: self-supervised semi-supervised learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (2019)
27. Zhang, T., El Ali, A., Wang, C., Hanjalic, A., Cesar, P.: Corrnet: fine-grained emotion recognition for video watching using wearable physiological sensors. Sensors **21**(1), 52 (2020)
28. Zhang, T., El Ali, A., Wang, C., Hanjalic, A., Cesar, P.: Rcea: real-time, continuous emotion annotation for collecting precise mobile video ground truth labels. In: Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems, pp. 1–15 (2020)
29. Zhang, Z., Zhong, S.h., Liu, Y.: Ganser: a self-supervised data augmentation framework for eeg-based emotion recognition. IEEE Trans. Affective Comput. (2022)

Multi-intent Driven Contrastive Sequential Recommendation

Yiyuan Zheng[1,2], Beibei Li[3], Beihong Jin[1,2](✉), Rui Zhao[1,2], Weijiang Lai[1,2],
and Tao Xiang[3]

[1] Key Laboratory of System Software (Chinese Academy of Sciences) and State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences,
Beijing, China
`Beihong@iscas.ac.cn`
[2] University of Chinese Academy of Sciences, Beijing, China
[3] College of Computer Science, Chongqing University, Chongqing, China

Abstract. Sequential Recommendation (SR) models with auxiliary tasks of contrastive learning have achieved remarkable progress in recent years, which can effectively mine the self-supervised signals to mitigate the data sparsity problem. However, current contrastive SR models overlook the intricate correlations among different users, leading to the false negative pair problem and adversely affecting recommendation performance. Therefore, in this paper, we propose a multi-intent driven contrastive SR model MICRec. MICRec learns global intent prototypes from the users by a moving-average updating strategy. Then, MICRec introduces two multi-intent guided contrastive losses, including a sequence-level contrastive loss and an intent-level contrastive loss, which both contribute to mining the self-supervised information and building accurate user embeddings. The former optimizes the negative sample set by eliminating the false negative sequence pairs with overlapping intents, and the latter further stabilizes the latent structure by aligning the intents excavated from the original and augmented sequences. Thus, with the multi-intent guided contrastive learning strategy, our model can better understand the correlations between users, leading to a more effective and accurate representation structure in the latent space. MICRec not only achieves superior performance, but also improves the robustness to the interaction noise. The experimental results on three public benchmark datasets show that MICRec outperforms existing SR models in terms of Recall and NDCG.

Keywords: Sequential Recommendation · Contrastive Learning · Implicit Multi-intent Modeling

1 Introduction

In today's world, Sequential Recommendation (SR) models are extensively utilized in various scenarios to offer personalized services. The SR models predict

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
A. Bifet et al. (Eds.): ECML PKDD 2024, LNAI 14949, pp. 141–156, 2024.
https://doi.org/10.1007/978-3-031-70378-2_9

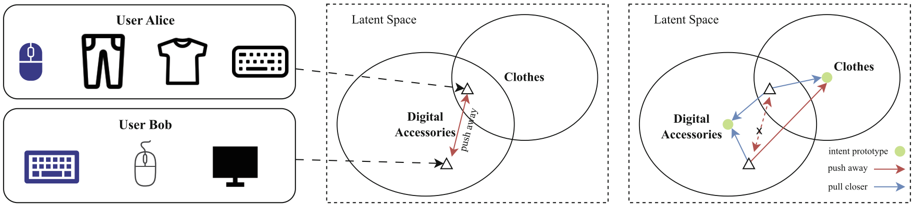


Fig. 1. The left shows the historical interactions of two users. The middle depicts the regular sequence-level contrastive learning, mistakenly pushing users with overlapping intents apart in the latent space. The right illustrates the multi-intent guided contrastive learning in our model, which applies intent-level contrastive learning and eliminates the false negative pairs in sequence-level contrastive learning.

users' next items of interest based on their historical interaction sequences. With the advancement of deep learning, techniques such as RNN, CNN, and self-attention mechanisms have been widely adopted in SR models [10,12,16,20], which makes it possible to provide more accurate and personalized recommendations.

However, in realistic scenarios, data sparsity and data noise problems seriously affect the performance of SR models. To alleviate these problems, contrastive learning is introduced to mine self-supervised information and improve robustness. Existing contrastive SR models typically employ either model-level [19] or data-level [17,22] augmentation to construct two augmented views based on the original sequences, where the model-level augmentation includes conducting dropout twice, adding Gaussian noise, etc., and the data-level augmentation includes cropping, masking, reordering, and inserting items into the sequence, etc.

Usually, contrastive learning is treated as an auxiliary task in contrastive SR models. Most existing contrastive SR models [17,19,22] are based on sequence-level contrastive losses, which often treat all the sequences within a minibatch as negative samples except itself. By pushing away the negative sequence pairs, contrastive learning improves the discrepancy of different sequence embeddings and the overall uniformity of embeddings. However, sequence-level contrastive learning can only learn the consistency between different augmented sequences of the same user, ignoring the correlations among different users at the intent level and simply pushing the sequence pairs of different users away in the latent space. For example, in Fig. 1, users Alice and Bob have an overlapping intent of purchasing digital accessories. However, sequence-level contrastive learning in the middle figure may regard user Alice and user Bob as negative pairs and manage to reduce their similarity in the latent space, which raises the false negative sample problem. The mistakenly pushing away harms the latent structure and finally degrades the recommendation performance.

However, accurately modeling the relevance and difference between users remains a challenge. On the one hand, the absence of supervision signals makes it hard to extract the implicit user intents. On the other hand, the historical interactions of a user can be exceedingly intricate, potentially encompassing multiple intents. As shown in Fig. 1, user Alice simultaneously shows two shopping intents: clothes and digital accessories. Existing contrastive learning methods [5,15] are typically single intent oriented, which cannot properly explore and exploit the implicit correlations among users due to the multiplicity of user intents.

To address the above issues, we propose a multi-intent driven contrastive model MICRec for the SR, which is trained based on global intent prototypes. Specifically, MICRec maintains a set of moving-average-updated prototypes to gather the information from their corresponding users. As shown in the right of Fig. 1, with the implicit intents in hand, we design two strategies to build the intent-level structure in the latent space. First, the false negative sequence pairs whose prototypes overlap partially or fully are eliminated, mitigating the damage of sequence-level contrastive learning to the accurate latent structure. Second, we introduce a multi-intent contrastive loss to further enhance the alignment between the user and his/her assigned multiple intent prototypes. Our contributions can be summarized as follows.

1. We propose a multi-intent driven contrastive SR model MICRec, which designs a lightweight intent prototype identifying and updating strategy to extract the representative intents in the latent space.
2. We introduce two multi-intent guided contrastive losses to accurately establish the intent-level consistency and discrepancy of users by eliminating false negative sequence pairs and enhancing the correlation between users and the intent prototypes.
3. We have conducted experiments on three publicly available datasets. The experimental results show that our model outperforms existing SR models significantly. Compared to the state-of-the-art competitors, our model improves Recall and NDCG by an average of 12.5% and 6.5% on the three datasets, respectively.

2 Related Work

2.1 Sequential Recommendation

Sequential Recommendation (SR) models learn user representations from their historical interaction sequences, assuming that user preferences depend on historical behaviors. Early SR models adopt Markov chains to capture transitions between items. In recent years, SR models have shifted towards leveraging various deep neural networks to enhance sequence representation learning. For instance, GRU4Rec [10] employs recurrent neural networks to capture the sequential information of user interactions. SR-GNN [21] and GC-SAN [23] take the items in each sequence as nodes, construct edges according to the interactions, and then learn the sequence representations with graph neural networks. Recently, attention-based SR models have achieved excellent recommendation performance. For example, STAMP [16] extracts short-term interest from the last item in a user's interaction sequence and calculates long-term interest using an attention mechanism. SASRec [12] leverages transformer blocks as the sequence encoder and further harness the potential of self-attention. BERT4Rec [20] utilizes a bidirectional transformer architecture to integrate information from both past and future interactions to enhance the understanding of user behaviors.

2.2 Contrastive Learning

Contrastive learning has achieved remarkable performance in both computer vision and natural language processing [2–4,8,9]. It treats augmented instances of the same instance as positive pairs, and augmented instances of different instances as negative pairs. However, the less scrupulous contrastive learning methods may result in the destruction of semantic structure. To avoid the collapse of semantic structure, PCL [15] designs the ProtoNCE loss to build the semantic structure discovered by clustering. Some other work focuses on identifying false negative pairs to mitigate the damage to the semantic structure. For example, SupCon [13] utilizes supervised labels directly to solve the false negative pairs problem, pulling together the samples belonging to the same class and pushing the samples from different classes away. FNC [11] constructs a support set to provide an alternative perspective to detect false negative pairs.

Recently, contrastive learning has been introduced into SR. CL4SRec [22] constructs positive sequence pairs by masking, cropping and reordering. CoSeRec [17] generates more informative sequences by introducing two augmentation schemes based on item correlation. DuoRec [19] utilizes supervised information to construct positive pairs, designing both data-level and model-level augmentation techniques. PDMRec [24] proposes contrastive learning strategies that reduce the interference from the positions of items. ICLRec [5] proposes a method that models implicit intents through clustering and assigns an intent to each sequence. It maximizes the agreement between the sequence embedding and its corresponding implicit intent prototype to exploit intent information. DSSRec [18] models multiple intents in a latent space for each sequence. However, ICLRec is unable to model multiple intents and the computation cost of clustering leads to poor timeliness of its prototype updates. DSSRec customizes the intents for each user and underlying correlations among different users are ignored. Compared to existing work, our model focuses on modeling implicit intents shared by each user and building the intent-level structure in latent space to achieve accurate recommendations.

3 Problem Formulation

We denote the user set as $\mathcal{U}$, the item set as $\mathcal{V}$, and the embedding of item $v \in \mathcal{V}$ as $\mathbf{v} \in \mathbb{R}^d$, where d is the dimension of the embedding. For each user $u \in \mathcal{U}$, we sort the items that the user has interacted with by timestamp to obtain the historical interaction sequence $s_u = [v_1^u, v_2^u, ...v_{|s_u|}^u]$. Given the sequence s_u, our goal is to build a model that predicts the next item $v_{|s_u|+1}^u$ that the user u is most likely to interact with. The problem is formulated as:

$$argmax_{v \in \mathcal{V}} P(v_{|s_u|+1}^u | s_u). \tag{1}$$

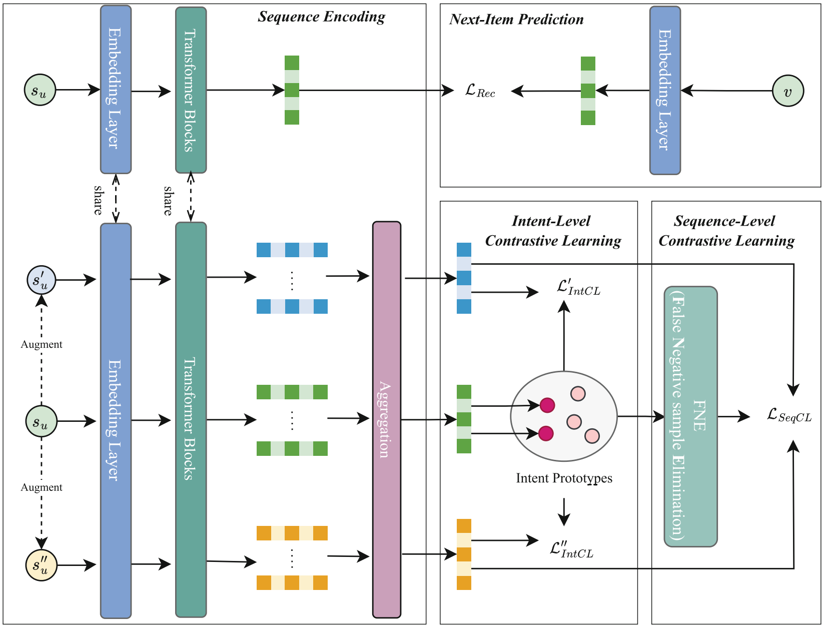


Fig. 2. The architecture of MICRec.

4 Our Model

4.1 Overview

The architecture of MICRec is shown in Fig. 2. Our model consists of the next item prediction and multi-intent guided contrastive learning, i.e., intent-level contrastive learning and sequence-level contrastive learning. The next item prediction and multi-intent guided contrastive learning share the embedding layer and the transformer blocks. During training, our model identifies and updates the intent prototypes by the embeddings of the original sequences. After that, the intent prototypes are utilized in intent-level contrastive losses to exploit self-supervised intent information and build the intent-level structure. The identified multiple intents are also used in the FNE module to eliminate false negative sequence samples in sequence-level contrastive learning.

4.2 Sequence Encoding

The sequential encoder based on Transformer effectively mines user interests, achieving excellent SR performance. The encoder extracts features by stacking self-attention layers and feed-forward network layers.

Given the historical interaction sequence $s_u = [v_1^u, v_2^u, ...v_{|s_u|}^u]$ of user u, we firstly transform it into a sequence of item embeddings, and add the corresponding position embeddings, which are a set of learned vectors. After that, we obtain the output of the embedding layer:

$$\mathbf{S}_u = \left[\mathbf{v}_1^u + \mathbf{p}_1, \mathbf{v}_2^u + \mathbf{p}_2, \ldots, \mathbf{v}_{|s_u|}^u + \mathbf{p}_{|s_u|}\right]. \tag{2}$$

Next, we use the multi-head self-attention block to capture correlations in sequence:

$$\mathbf{S}'_u = \text{Attention}_{\text{MH}}(\mathbf{S}_u) = \text{concat}(head_1; head_2, ...; head_h)\mathbf{W}^O, \tag{3}$$

$$head_i = \text{Attention}(\mathbf{S}_u\mathbf{W}_i^Q, \mathbf{S}_u\mathbf{W}_i^K, \mathbf{S}_u\mathbf{W}_i^V) \qquad i = 1, 2, .., h, \tag{4}$$

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}(\frac{\mathbf{Q}\mathbf{K}^{\mathbf{T}}}{\sqrt{d/h}})\mathbf{V}, \tag{5}$$

where h is the number of heads, $\mathbf{W}_i^Q, \mathbf{W}_i^K, \mathbf{W}_i^V \in \mathbb{R}^{d\times \frac{d}{h}}$, $\mathbf{W}^o \in \mathbb{R}^{d\times d}$. Also, we forbid the connection between $\mathbf{Q}_i$ and $\mathbf{K}_j$ $(j > i)$ in our model. Next, to enhance the fitting ability, a nonlinear feed-forward network layer is added as follows:

$$\mathbf{F}_u = \text{FFN}\left(\mathbf{S}'_u\right) = \text{ReLU}\left(\mathbf{S}'_u\mathbf{W}^{(1)} + \mathbf{b}^{(1)}\right)\mathbf{W}^{(2)} + \mathbf{b}^{(2)}, \tag{6}$$

where $\mathbf{W}^{(1)}, \mathbf{W}^{(2)} \in \mathbb{R}^{d\times d}$, $\mathbf{b}^{(1)}, \mathbf{b}^{(2)} \in \mathbb{R}^d$. After that, $\mathbf{F}_u$ is fed into the multi-head self-attention layer of the next block. To alleviate the problem that deep networks are difficult to converge, we use residual connections between different layers and layer normalization to stabilize the training. Finally, we record the output of the last block as $\mathbf{H}_u = [\mathbf{h}_1^u, \mathbf{h}_2^u, .., \mathbf{h}_{|s_u|}^u]$, where $\mathbf{h}_i^u$ is the output of the i-th time step of the encoder.

We take the output of the $|s_u|$-th time step $\mathbf{h}_{|s_u|}^u$ as the sequence embedding. Particularly, we employ a trick when calculating the score of user-item interactions: normalizing the embedding and using a temperature hyper-parameter to scale the interaction score. For user $u \in \mathcal{U}$, his/her interaction score with item $v_j \in \mathcal{V}$ is predicted as follows:

$$\hat{y}_{uj} = \frac{{\mathbf{v}_j}^T\mathbf{h}_{|s_u|}^u}{||\mathbf{v}_j||_2||\mathbf{h}_{|s_u|}^u||_2} \cdot \frac{1}{\tau}, \tag{7}$$

where τ is the temperature hyper-parameter. The above equation can be seen as scaling the cosine similarity between embeddings. We consider the next item v_i that the user u interacts with as the positive sample and the other items in the item set as negative samples. The recommendation loss is constructed as follows:

$$\mathcal{L}_{\text{Rec}}(u) = -\log \frac{exp(\hat{y}_{ui})}{\sum_{v_j\in\mathcal{V}} exp(\hat{y}_{uj})}. \tag{8}$$

4.3 Intent Prototype Identifying and Updating

We propose a moving-average updating strategy to learn intent prototypes. Specifically, we initialize a set of intent prototypes $\mathcal{P}$, where each normalized prototype $\mathbf{p}_i \in \mathbb{R}^d$ represents an implicit intent and $|\mathcal{P}|$ denotes the number of intent prototypes. We view the prototypes as intent centers in the latent preference space and explore the correlation between users and the prototypes.

We believe that the longer interaction sequences may imply more intents. Therefore, we assume that the number of intents implied by a sequence is proportional to the length of the sequence, rather than fixing the number of intents corresponding to each sequence. We calculate the number of intents corresponding to each sequence by the following equation:

$$k_u = \min(\lceil \frac{|s_u|}{\rho} \rceil), k_{max}), \tag{9}$$

where ρ is used to adjust the number of intents implied in each sequence and k_{max} limits the maximum number of intents in a single sequence. Next, we compute the cosine similarity between the sequence embedding and the intent prototypes by Eq. 10. Note that the intent prototype $\mathbf{p}_i$ is a normalized vector, i.e.,

$$\hat{s}_{ui} = \frac{{\mathbf{p}_i}^T \mathbf{h}^u_{|s_u|}}{||\mathbf{h}^u_{|s_u|}||_2}. \tag{10}$$

After that, we find the top k_u intents with the highest similarity and obtain the set of intents corresponding to the sequence I_u:

$$I_u = \{j \mid \hat{s}_{uj} \in \mathrm{top}(\{\hat{s}_{ui}, \mathbf{p}_i \in \mathcal{P}\}, k_u)\}, \tag{11}$$

where the $\mathrm{top}(\cdot, k)$ function returns the largest k elements in the input set and I_u represents the intents implied in the sequence s_u.

Learning the prototypes by clustering [5,15] may cause a huge computational overhead. Therefore, we update the prototypes with a moving-average style as follows:

$$\mathbf{p}_j = \mathrm{Normalized}(\beta \mathbf{p}_j + (1 - \beta) \cdot \mathrm{Normalized}(\mathbf{h}^u_{|s_u|})) \qquad if\ j \in I_u, \tag{12}$$

where β is a hyper-parameter. The moving-average updating strategy of the intent prototypes can incorporate the information of a single sequence into multiple intents. At the same time, the moving-average updating makes the process of modeling intents more stable, less susceptible to individual noisy data, and effectively reduces the computational overhead, which is capable of being applied with each minibatch.

4.4 Multi-intent Guided Contrastive Learning

Sequence-Level Contrastive Learning. For each historical sequence s_u, we randomly select two data augmentation methods to obtain positive pairs $s'_u = Aug_1(s_u)$ and $s''_u = Aug_2(s_u)$, where $Aug_*(\cdot) \in \{mask, crop, reorder\}$. To improve the effectiveness of contrastive learning, we follow the approach of previous work [5,17] and concatenate the normalized output at all time steps to obtain the augmented sequence representation $\mathbf{h}'_u$ and $\mathbf{h}''_u$.

Specifically, we calculate the similarity between the augmented sequence pairs by $\mathrm{sim}(\mathbf{h}'_u, \mathbf{h}''_u) = \frac{1}{\tau'} \cdot \mathbf{h}'^{T}_u \mathbf{h}''_u$, where τ' is a temperature hyper-parameter. Next, we apply the InfoNCE loss to construct contrastive loss as follows:

$$\mathcal{L}_{\text{InfoNCE}}\left(\mathbf{h}'_u, \mathbf{h}''_u\right) = -\log \frac{e^{\mathrm{sim}\left(\mathbf{h}'_u, \mathbf{h}''_u\right)}}{e^{\mathrm{sim}\left(\mathbf{h}'_u, \mathbf{h}''_u\right)} + \mathcal{T}_{\text{neg}}}, \tag{13}$$

$$\mathcal{T}_{\text{neg}} = \sum_{i \in \mathcal{H}^-} e^{\mathrm{sim}\left(\mathbf{h}'_u, \mathbf{h}'_i\right)} + \sum_{i \in \mathcal{H}^-} e^{\mathrm{sim}\left(\mathbf{h}'_u, \mathbf{h}''_i\right)}, \tag{14}$$

where $\mathcal{H}^-$ denotes the set of negative users, i.e., users in the minibatch excluding u.

FNE Module. The augmented sequences of negative users may introduce false negative pairs that are harmful to the latent structure. Considering that each sequence has multiple implicit intents, whether the two sequences have overlapping intents is taken as a criterion to identify the false negative sequence pairs and improve the quality of the negative sample set. Given the user u, his/her optimized negative user set is shown as follows:

$$\mathcal{N}_u = \{i \mid \text{ if } I_u \cap I_i = \phi\}. \tag{15}$$

Then, we improve the sequence-level contrastive loss based on the optimized set as follows:

$$\mathcal{L}_{\text{SeqCL}}\left(\mathbf{h}'_u, \mathbf{h}''_u\right) = -\log \frac{e^{\mathrm{sim}\left(\mathbf{h}'_u, \mathbf{h}''_u\right)}}{e^{\mathrm{sim}\left(\mathbf{h}'_u, \mathbf{h}''_u\right)} + \mathcal{T}^{+}_{\text{neg}}}, \tag{16}$$

$$\mathcal{T}^{+}_{\text{neg}} = \sum_{i \in (\mathcal{H}^- \cap \mathcal{N}_u)} e^{\mathrm{sim}\left(\mathbf{h}'_u, \mathbf{h}'_i\right)} + \sum_{i \in (\mathcal{H}^- \cap \mathcal{N}_u)} e^{\mathrm{sim}\left(\mathbf{h}'_u, \mathbf{h}''_i\right)}, \tag{17}$$

where $\mathcal{H}^- \cap \mathcal{N}_u$ denotes the optimized negative user set in the minibatch, which eliminates the false negative sequence pairs and alleviates the destruction of the latent structure. The **F**alse **N**egative sample **E**limination is denoted as the FNE module.

Intent-Level Contrastive Learning. Intent-level contrastive learning is employed to improve the coherence of intents conveyed by the augmented sequences and the original sequences, exploiting the self-supervised information of users at the intent level.

The intent set I_u is considered as the positive intents of the user u. The augmented sequences usually keep similar intents to the original sequence. Based on the above view, we construct the following contrastive loss to maximize the consistency between the augmented sequence embeddings and the prototypes of their positive intents:

$$\mathcal{L}'_{\text{IntCL}}\left(\mathbf{h}'_u, I_u\right) = -\frac{1}{|I_u|} \sum_{i \in I_u} \log \frac{e^{\mathrm{sim}\left(\mathbf{h}'_u, \mathbf{p}_i\right)}}{\sum_{\mathbf{p}_j \in \mathcal{P}} e^{\mathrm{sim}\left(\mathbf{h}'_u, \mathbf{p}_j\right)}}, \tag{18}$$

$$\mathcal{L}''_{\text{IntCL}}\left(\mathbf{h}''_u, I_u\right) = -\frac{1}{|I_u|}\sum_{i\in I_u}\log\frac{e^{\text{sim}\left(\mathbf{h}''_u,\mathbf{p}_i\right)}}{\sum_{\mathbf{p}_j\in\mathcal{P}} e^{\text{sim}\left(\mathbf{h}''_u,\mathbf{p}_j\right)}}. \tag{19}$$

Note that the intent prototypes will be frozen from gradient descent, only updated from the moving-average strategy mentioned in Sect. 4.3. Finally, we obtain the overall intent-level contrastive loss for user u as follows:

$$\mathcal{L}_{\text{IntCL}}(u) = \mathcal{L}'_{\text{IntCL}}\left(\mathbf{h}'_u, I_u\right) + \mathcal{L}''_{\text{IntCL}}\left(\mathbf{h}''_u, I_u\right). \tag{20}$$

4.5 Overall Loss

We consider sequence-level contrastive learning and intent-level contrastive learning as two auxiliary tasks and we train our model by optimizing the following joint loss:

$$\mathcal{L} = \begin{cases} \mathcal{L}_{\text{Rec}} + \lambda\cdot\mathcal{L}_{\text{InfoNCE}}, & \text{if } epoch <= E \\ \mathcal{L}_{\text{Rec}} + \lambda\cdot\left(\mathcal{L}_{\text{SeqCL}} + \mathcal{L}_{\text{IntCL}}\right) & \text{else} \end{cases} \tag{21}$$

where λ controls the intensity of the two auxiliary tasks, E is the hyper-parameter controlling the number of warm-up epochs. At the early stage of training, the prototypes are insufficient for providing guidance. Therefore, we apply a two-stage training approach, starting with a warm-up stage by only using $\mathcal{L}_{\text{InfoNCE}}$ as the auxiliary loss. After warming up, the model has the basic modeling capability and intent information. Then we use $\mathcal{L}_{\text{SeqCL}} + \mathcal{L}_{\text{IntCL}}$ as the auxiliary loss until convergence.

4.6 Discussion

Impact of Embedding Normalization. As shown in Eq. 7, MICRec applies embedding normalization before calculating similarities in both the next item prediction and the contrastive learning. Compared to dot product similarity, we find that using the cosine similarity with normalized embeddings can achieve a significant performance improvement in the experiments. This might be because the normalized embeddings can alleviate popularity bias compared to directly using embeddings of free-varying magnitude, which is also revealed in a recent study [1]. Thus, we apply the method to our model.

Time Complexity. In the training phase, the main time consumption comes from two parts, i.e., encoding sequences and identifying the intent prototypes. The time complexity of encoding the original and augmented sequences is $O(3\cdot(Ld^2 + L^2d))$, where L is the maximum length of the input sequence. The time complexity of identifying the intents for a sequence is $O(|\mathcal{P}|\log k_{max})$. In the prediction phase, the intent prototype is not involved, and we do not need to compute the embedding of the augmented sequence, so the overall time complexity reduces to $O(Ld^2 + L^2d)$. Compared to the other contrastive SR models like CL4SRec, MICRec does not increase the computational cost significantly. Meanwhile, the move-average update strategy makes it more efficient than the intent contrastive SR model ICLRec during the training phase.

Table 1. Statistics of the datasets after preprocessing.

Dataset	#Users	#Items	#Interactions	Density
Beauty	22,363	12,101	198,502	0.07%
Toy_and_Games	19,412	11,924	167,597	0.07%
Sports_and_Outdoors	35,598	18,357	270,544	0.04%

5 Experiments

5.1 Experimental Setting

Datasets. Considering that the SR models are widely used in e-commerce scenarios, we choose 3 subcategories of the widely used Amazon dataset[1] to conduct experiments, including Beauty, Toy_and_Games, and Sports_and_Outdoors.

During the dataset preprocessing, users or items appearing less than 5 times are removed. The statistics of the processed datasets are shown in Table 1. We adopt the leave-one-out strategy to split the datasets. For an interaction sequence $s_u = [v_1^u, v_2^u ... v_{|s_u|}^u]$, we leave the last interaction $v_{|s_u|}^u$ for testing and $v_{|s_u|-1}^u$ for validation. The remaining interactions $[v_1^u, v_2^u .. v_{|s_u|-2}^u]$ are used for training.

Evaluation Metrics. We follow the common evaluation metrics in the next item prediction task, including Recall@K and NDCG@K (R@K and N@K for short). In this paper, we present our findings for K values of 10, 20, and 50. We use the ranking results on the whole set of item candidates to evaluate for fairness [14].

Baselines. We choose the following models as competitors, including four regular SR models, i.e., GRU4Rec [10], STAMP [16], SASRec [12], BERT4Rec [20]. Additionally, we compare our model to two contrastive SR models, CL4SRec [22] and DuoRec [19], along with two implicit-intent-based SR models, DSSRec [18] and ICLRec [5].

Implementation Details. For the competitors, GRU4Rec, STAMP, SASRec, BERT4Rec and CL4SRec are implemented based on an open-source framework Recbole[2]. DuoRec[3], DSSRec[4] and ICLRec[5] are implemented with their open-source code. We set the dimension of embeddings to 64, batch size to 512 and dropout to 0.25. The maximum sequence length is set to 100. Particularly, for the model applying the self-attention mechanism, we set the attention head to 2,

---

[1] http://jmcauley.ucsd.edu/data/amazon/index.html.

[2] https://github.com/RUCAIBox/RecBole.

[3] https://github.com/RuihongQiu/DuoRec.

[4] https://github.com/abinashsinha330/DSSRec.

[5] https://github.com/salesforce/ICLRec.

and the number of layers to 2. The remaining hyper-parameters are configured following the specifications outlined in their respective papers.

We implement our model in PyTorch. We initialize the parameters and intent prototypes by a normal distribution in the range $[-0.02, 0.02]$. For the hyper-parameters, we tune the weight of contrastive loss λ, the size of the intent prototype set $|\mathcal{P}|$ and the number of warm-up epochs E in $\{0.01, 0.05, 0.1, 0.5\}$, $\{64, 128, 256, 512\}$ and $\{0, 2, 4, 8\}$, respectively. ρ and k_{max} are set to 4 and 8. The moving-average updating weight β is set to 0.99. The temperature hyper-parameters τ and τ' are both set to 0.1.

We train the models with an early stopping strategy. That is, we stop training if Recall@20 does not improve on the validation set for 10 epochs consecutively. We train our model with three different random seeds, and report the mean value of the results and the confidence interval with a 95% confidence level.

5.2 Performance Comparison

Table 2 gives the experimental results for the overall performance comparison. From the results, we have the following observations. STAMP, which is based on Short-Term Attention, has a better performance compared to the GRU4Rec based on the recurrent neural network on most of the datasets. SASRec achieves the best performance among all regular SR models, indicating the effectiveness of the self-attention mechanism in sequence modeling. However, BERT4Rec performs poorly on most of the datasets, potentially because the masked item prediction training method is not suitable for the next item prediction task.

The contrastive SR model CL4SRec outperforms SASRec due to the effectiveness of introducing sequence-level contrastive learning to mine self-supervised information. DuoRec outperforms CL4SRec in most scenarios, suggesting that combining supervised and unsupervised contrastive learning allows the model to learn better representations. However, despite DSSRec and ICLRec modeling implicit intents, their performance is inferior to that of CL4SRec and DuoRec. This indicates that they have limitations in intent modeling. ICLRec focuses on modeling a single intent for each user, while DSSRec does not incorporate the implicit intents from global user correlations, resulting in a marginal performance impairment.

Our model MICRec significantly outperforms other competitors in all metrics on all datasets. For example, MICRec has 15.6% and 9.14% average improvement over the second-best model in Recall and NDCG on the Sports dataset, respectively. This is attributed to the multi-intent guided contrastive losses, which can effectively exploit self-supervision signals at different levels while maintaining the accurate latent structure.

5.3 Ablation Study

Table 3 shows the ablation experiment results of MICRec. Variants a, b and c remove sequence-level contrastive learning, intent-level contrastive learning

Table 2. Performance comparisons of different models. The best in each row is in bold, and the second-best is underlined. The last column shows the relative improvements compared to the best baseline results.

Dataset	Metric	GRU4Rec	STAMP	BERT4Rec	SASRec	CL4SRec	DuoRec	DSSRec	ICLRec	MICRec	Improv.
Beauty	R@10	0.0633	0.0663	0.0338	0.0785	0.0797	0.0824	0.0627	0.0648	**0.0889**±0.0011	7.85%
	R@20	0.0917	0.0922	0.0565	0.1120	0.1138	0.1147	0.0936	0.1004	**0.1289**±0.0021	12.41%
	R@50	0.1459	0.1416	0.1024	0.1684	0.1714	0.1751	0.1478	0.1591	**0.1945**±0.0020	11.10%
	N@10	0.0338	0.0395	0.0146	0.0380	0.0382	0.0418	0.0328	0.0283	**0.0430**±0.0004	2.95%
	N@20	0.0409	0.0460	0.0203	0.0465	0.0468	0.0499	0.0405	0.0372	**0.0531**±0.0006	6.48%
	N@50	0.0516	0.0557	0.0293	0.0577	0.0582	0.0619	0.0512	0.0488	**0.0661**±0.0004	6.79%
Toys	R@10	0.0478	0.0705	0.0380	0.0842	0.0866	0.0890	0.0712	0.0735	**0.0966**±0.0030	8.58%
	R@20	0.0746	0.0921	0.0592	0.1155	0.1196	0.1203	0.0989	0.1086	**0.1342**±0.0030	11.55%
	R@50	0.1235	0.1308	0.0993	0.1705	0.1718	0.1717	0.1476	0.1655	**0.1962**±0.0017	14.29%
	N@10	0.0249	0.0469	0.0162	0.0403	0.0418	0.0441	0.0397	0.0332	**0.0451**±0.0010	2.34%
	N@20	0.0317	0.0523	0.0216	0.0483	0.0501	0.0520	0.0466	0.0420	**0.0546**±0.0010	5.06%
	N@50	0.0414	0.0599	0.0294	0.0592	0.0605	0.0622	0.0563	0.0532	**0.0669**±0.0007	7.56%
Sports	R@10	0.0334	0.0395	0.0182	0.0438	0.0448	0.0463	0.0311	0.0380	**0.0534**±0.0012	15.26%
	R@20	0.0521	0.0553	0.0326	0.0649	0.0660	0.0684	0.0486	0.0583	**0.0790**±0.0011	15.55%
	R@50	0.0909	0.0858	0.0630	0.1018	0.1052	0.1086	0.0845	0.0997	**0.1260**±0.0028	15.99%
	N@10	0.0171	0.0238	0.0083	0.0202	0.0205	0.0229	0.0165	0.0169	**0.0246**±0.0002	7.42%
	N@20	0.0218	0.0278	0.0119	0.0255	0.0258	0.0284	0.0209	0.0220	**0.0310**±0.0004	9.27%
	N@50	0.0295	0.0338	0.0179	0.0328	0.0335	0.0364	0.0280	0.0302	**0.0403**±0.0008	10.71%

Table 3. Ablation Study of MICRec. (Recall@20)

Model	Beauty	Toys	Sports
MICRec	**0.1289**	**0.1342**	**0.0790**
a) w/o $\mathcal{L}_{\text{IntCL}}$	0.1266	0.1334	0.0775
b) w/o $\mathcal{L}_{\text{SeqCL}}$	0.1250	0.1307	0.0768
c) w/o FNE	0.1266	0.1316	0.0782
d) w/o $\mathcal{L}_{\text{IntCL}}$ and $\mathcal{L}_{\text{SeqCL}}$	0.1222	0.1297	0.0756

and FNE module from MICRec, respectively. Variant d removes the contrastive learning at both levels.

It can be observed that variants a, b, and c all show a certain degree of performance degradation compared to MICRec, indicating that both the multi-intent guided contrastive losses and the FNE module contribute to the recommendation performance, which is due to the fact that the contrastive losses effectively mine the self-supervised information in the sequence-level and intent-level, and the optimized negative sample set can alleviate the disruption caused by false negative sequence pairs. Variant d exhibits inferior performance compared to variants a and b, suggesting that incorporating the multi-intent guided contrastive losses can yield benefits for the performance.

5.4 Robustness Analysis

Robustness Against Interaction Sparsity. The preferences of users with few interactions are more difficult to capture. Therefore, we divide users into four groups according to the interaction frequency, and observe the recommendation

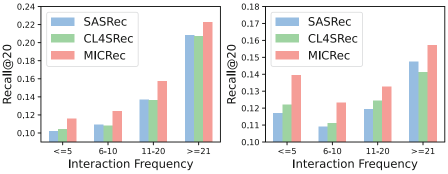


Fig. 3. Performance on the user groups with different interaction frequencies. (Left: Beauty, Right: Toys)

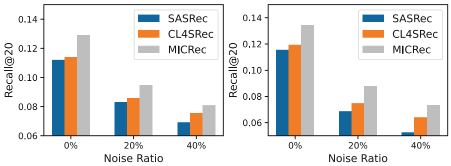


Fig. 4. Performance comparison with different added noise ratios. (Left: Beauty, Right: Toys)

performance of SASRec, CL4SRec and MICRec. Figure 3 shows that the performance of CL4SRec has a stable improvement compared to SASRec on the user group with few interactions, which indicates that contrastive learning can alleviate the data sparsity problem. MICRec demonstrates a notably superior performance compared to SASRec and CL4SRec across all user groups in both datasets. Particularly noteworthy is the greater improvement observed in user groups with limited interactions, which indicates that multi-intent guided two-level contrastive learning can sufficiently exploit the self-supervised information to alleviate the interaction sparsity compared to the regular sequence-level contrastive loss.

Robustness Against Noise. To explore the performance of the MICRec model in the noisy scenarios, we add 20% and 40% of random items to the train dataset, respectively. From Fig. 4, we can find that CL4SRec outperforms SASRec in noisy scenarios. This is because sequence-level contrastive learning brings better uniformity for sequence embeddings, which may avoid the over-fitting of noisy data. MICRec performs better than SASRec and CL4SRec, and the relative improvement increases as the noise ratio increases. This reflects its excellent robustness against noisy data, possibly attributed to the model's capacity to enrich the intent-level structure in the latent space, thus mitigating significant perturbations and maintaining good recommendation performance even in a noisy scenario.

5.5 Impact of Flase Negative Sample Elimination

Given the joint optimization of the next item prediction task and contrastive learning task, false negative pairs could introduce conflicts and disrupt the accurate latent structure during the optimization process. In this section, we investigate the significant role of the FNE module in mitigating conflicts. Specifically, all the variants only apply sequence-level contrastive learning as the auxiliary task and the weights of contrastive learning are all set to 1. The difference is that variant MICRec$_{\text{FNE}}$(8) applies the FNE module after a warm-up stage with the number of warm-up epochs set to 8, while variant MICRec$_{\text{FNE}}$(64) does so with the number set to 64.

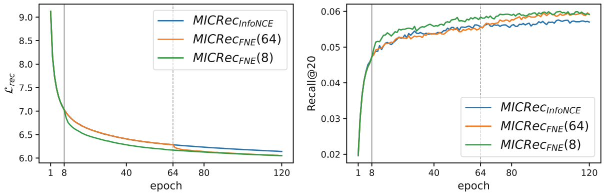


Fig. 5. The curve of recommendation loss $\mathcal{L}_{\text{Rec}}$ on train dataset and the evaluation metric Recall@20 on validation dataset during training on Beauty dataset.

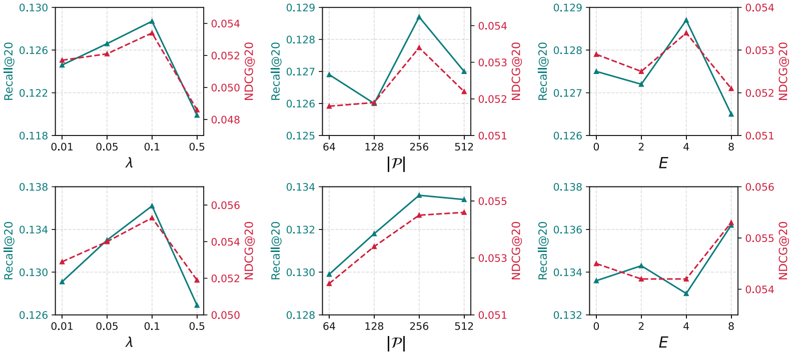


Fig. 6. Hyper-parameter sensitivity of λ, $|\mathcal{P}|$ and E. (Upper: Beauty, Lower: Toys)

Figure 5 shows the curve of next item prediction loss $\mathcal{L}_{\text{Rec}}$ and the metric Recall@20 on the validation dataset during training 120 epochs on the Beauty dataset. It can be observed that the decay rate of the $\mathcal{L}_{\text{Rec}}$ becomes larger and the performance of $\text{MICRec}_{\text{FNE}}(8)$ and $\text{MICRec}_{\text{FNE}}(64)$ gradually exceeds $\text{MICRec}_{\text{InfoNCE}}$ after applying the FNE module. This indicates that the FNE can mitigate potential conflicts when jointly optimizing the recommendation loss and sequence-level contrastive learning loss. It accelerates the convergence of the recommendation task, aids in learning high-quality embeddings, and ultimately improves the recommendation performance.

5.6 Hyper-parameter Sensitivity

We perform experiments on the Beauty and Toys datasets to explore the sensitivity of the weight of contrastive loss λ, the size of intent prototype set $|\mathcal{P}|$, and the number of warm-up epochs E. We fix other hyper-parameters, and tune λ, $|\mathcal{P}|$ and E within {0.01, 0.05, 0.1, 0.5}, {64, 128, 256, 512}, {0, 2, 4, 8}, respectively. From the results in Fig. 6, we can observe that MICRec demonstrates peak performance on both datasets when the weight of the contrastive loss λ is set to 0.1. Notably, both excessively large and small weights diminish the model's performance. The model performs poorly when the value of $|\mathcal{P}|$ is small. This could be attributed to the limited size of the intent prototype set, causing the model to overly emphasize coarse-grained intent information while lacking detailed information. MICRec achieves optimal performance on the datasets Beauty and Toys when the number of warm-up epochs E is set to 4 and 8, respectively. This suggests that employing a warm-up stage assists the model in establishing a foundational latent structure, thereby enhancing the learning process.

6 Conclusion

In this paper, we propose a multi-intent driven contrastive SR model MICRec, which is equipped with multi-intent guided two-level contrastive losses. Our model learns implicit intent prototypes through a moving-average updating strategy, utilizing implicit intents to eliminate false negative sequence pairs and mine self-supervised information at the intent-level. Experiments on several publicly available datasets show that our model outperforms existing SR models.

Acknowledgment. This work was supported by the National Natural Science Foundation of China under Grant No. 62072450 and Meituan.

References

1. Chen, J., Wu, J., Wu, J., Cao, X., Zhou, S., He, X.: Adap-τ: adaptively modulating embedding magnitude for recommendation. In: Proceedings of the ACM Web Conference 2023, pp. 1085–1096 (2023)
2. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: International Conference on Machine Learning, pp. 1597–1607 (2020)
3. Chen, T., Kornblith, S., Swersky, K., Norouzi, M., Hinton, G.E.: Big self-supervised models are strong semi-supervised learners. In: Advances in Neural Information Processing Systems, pp. 22243–22255 (2020)
4. Chen, X., Xie, S., He, K.: An empirical study of training self-supervised vision transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Visio,. pp. 9640–9649 (2021)
5. Chen, Y., Liu, Z., Li, J., McAuley, J., Xiong, C.: Intent contrastive learning for sequential recommendation. In: Proceedings of the ACM Web Conference 2022, pp. 2172–2182 (2022)
6. Dong, X., Jin, B., Zhuo, W., Li, B., Xue, T.: Improving sequential recommendation with attribute-augmented graph neural networks. In: Pacific-Asia Conference on Knowledge Discovery and Data Mining, pp. 373–385 (2021)
7. Dong, X., Jin, B., Zhuo, W., Li, B., Xue, T.: Sirius: sequential recommendation with feature augmented graph neural networks. In: Jensen, C.S., et al. (eds.) DASFAA 2021. LNCS, vol. 12683, pp. 315–320. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-73200-4_21
8. Gao, T., Yao, X., Chen, D.: SimCSE: simple contrastive learning of sentence embeddings. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pp. 6894–6910 (2021)

9. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9729–9738 (2020)
10. Hidasi, B., Karatzoglou, A., Baltrunas, L., Tikk, D.: Session-based recommendations with recurrent neural networks. In: Proceedings of the 4th International Conference on Learning Representations (2016)
11. Huynh, T., Kornblith, S., Walter, M.R., Maire, M., Khademi, M.: Boosting contrastive self-supervised learning with false negative cancellation. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 2785–2795 (2022)
12. Kang, W.C., McAuley, J.: Self-attentive sequential recommendation. In: Proceedings of the IEEE International Conference on Data Mining, pp. 197–206 (2018)
13. Khosla, P., et al.: Supervised contrastive learning. In: Advances in Neural Information Processing Systems, pp. 18661–18673 (2020)
14. Krichene, W., Rendle, S.: On sampled metrics for item recommendation. Commun. ACM **65**(7), 75–83 (2022)
15. Li, J., Zhou, P., Xiong, C., Hoi, S.: Prototypical contrastive learning of unsupervised representations. In: Proceedings of the International Conference on Learning Representations (2020)
16. Liu, Q., Zeng, Y., Mokhosi, R., Zhang, H.: Stamp: short-term attention/memory priority model for session-based recommendation. In: Proceedings of the 24th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, pp. 1831–1839 (2018)
17. Liu, Z., Chen, Y., Li, J., Yu, P.S., McAuley, J., Xiong, C.: Contrastive self-supervised sequential recommendation with robust augmentation. arXiv preprint arXiv:2108.06479 (2021)
18. Ma, J., Zhou, C., Yang, H., Cui, P., Wang, X., Zhu, W.: Disentangled self-supervision in sequential recommenders. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 483–491 (2020)
19. Qiu, R., Huang, Z., Yin, H., Wang, Z.: Contrastive learning for representation degeneration problem in sequential recommendation. In: Proceedings of the 15th ACM International Conference on Web Search and Data Mining, pp. 813–823 (2022)
20. Sun, F., et al.: BERT4Rec: sequential recommendation with bidirectional encoder representations from transformer. In: Proceedings of the 28th ACM International Conference on Information and Knowledge Management, pp. 1441–1450 (2019)
21. Wu, S., Tang, Y., Zhu, Y., Wang, L., Xie, X., Tan, T.: Session-based recommendation with graph neural networks. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 346–353 (2019)
22. Xie, X., et al.: Contrastive learning for sequential recommendation. In: Proceedings of the 38th IEEE International Conference on Data Engineering, pp. 1259–1273 (2022)
23. Xu, C., et al.: Graph contextualized self-attention network for session-based recommendation. In: Proceedings of International Joint Conference on Artificial Intelligence, pp. 3940–3946 (2019)
24. Yu, Y., Jin, B., Song, J., Li, B., Zheng, Y., Zhuo, W.: Improving micro-video recommendation by controlling position bias. In: Amini, M.R., Canu, S., Fischer, A., Guns, T., Kralj Novak, P., Tsoumakas, G. (eds.) ECML PKDD 2022. LNCS, vol. 13713, pp. 508–523. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-26387-3_31

KAT5: Knowledge-Aware Transfer Learning with a Text-to-Text Transfer Transformer

Mohammad Golam Sohrab[1](✉) and Makoto Miwa[1,2]

[1] Artificial Intelligence Research Center, National Institute of Advanced Industrial Science and Technology, Tokyo, Japan
`sohrab.mohammad@aist.go.jp`, `makoto-miwa@toyota-ti.ac.jp`
[2] Toyota Technological Institute, Nagoya, Japan

Abstract. We introduce knowledge-aware transfer learning with a text-to-text transfer transformer (KAT5) by leveraging a text-to-text transfer transformer (T5) in the Wikipedia domain. In standard transfer learning like T5, a model is first pre-trained on an unsupervised data task with a language model objective before fine-tuning it on a downstream task. T5 explores several learning objectives, including masked language model (MLM), random span, and deshuffling, where the model is limited to exploring integrating knowledge during pre-training. Here, we push the limits of this model by grafting knowledge like entity and co-reference information by mapping Wikipedia and Wikidata during pre-training. We align large-scale alignments between Wikipedia abstract and Wikidata triples to facilitate our pre-training KAT5 model. Our approach can match or outperform task-specific models while using the same architecture and hyper-parameters, in particular in entity and relation extraction (CoNLL04, ADE, and NYT datasets), and language generation tasks, including abstractive summarization (XSum, CNNDM), and machine translation. Our code is publicly released on GitHub (https://github.com/aistairc/kat5) under the Apache 2.0 License.

Keywords: Natural language processing · Transfer learning · Language model · Sequence-to-Sequence · Language understanding and generation · Information extraction · Machine translation

1 Introduction

In this work, to better capture the awareness of knowledge in language modeling pre-training, we present a knowledge-aware text-to-text transfer transformer that packs more information into the T5 model [26], which we call KAT5. During transfer learning a model is first pre-trained on a large-scale unsupervised data task and the most successful approaches have been variants of masked language models (MLMs), which are denoising autoencoders that are trained to reconstruct text by masking out a random subset of the input sequence.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
A. Bifet et al. (Eds.): ECML PKDD 2024, LNAI 14949, pp. 157–173, 2024.
https://doi.org/10.1007/978-3-031-70378-2_10

Integrating knowledge like entity or coreference information during transfer learning in NLP is not a common fashion as it needs to label a large-scale dataset. Such large-scale label dataset is not available, therefore, it is common to pre-train the entire model using data-rich unsupervised learning on unlabeled data. Our baseline model, T5 investigates different objective tasks, including masked language model (MLM), random span, and deshuffling, where the model is limited to exploring integrating knowledge during pre-training. Here, we push the limits of this model by grafting knowledge like entity and co-reference information by mapping Wikipedia and Wikidata during pre-training.

We propose a novel knowledge-aware T5 (KAT5) sequence-to-sequence (S2S) method with encoders and decoders that integrates entities and their coreferences as knowledge during pre-training. The KAT5 model is based on existing S2S architectures to allow parameter initialization from publicly available T5 checkpoints. We perform large-scale alignments between Wikipedia abstract and Wikidata triples to facilitate our pre-training KAT5 model and further research on integrating knowledge into large-scale pre-training. We show that initialization with knowledge-aware pre-training is effective for various downstream tasks. We fine-tune and evaluate the KAT5 model in joint entity-relation extraction (CoNLL04, ADE, and NYT dataset) and generation tasks - abstractive summarization (XSum, CNNDM datasets), and machine translation (English to German and English to Romanian datasets). We compare its performance with several recent state-of-the-art models.

Our implementation of the KAT5 model is based on Huggingface transformers. To the best of our knowledge, this is the first model to handle such a large-scale knowledge alignment during pre-training.

2 Related Work

Pre-trained Language Models such as BERT [5], RoBERTa [19], T5 [28], GPT-2 [27], XLNet [36], and XLM [4] are neural networks trained on large-scale datasets that can be fine-tuned on task-specific data. These models have significantly transformed the NLP landscape by showing remarkable success in a wide range of NLP tasks. Here, we draw our little scratch by introducing **K**nowledge-**a**ware T5 (KAT5) to facilitate different NLP tasks.

Text-to-Text Transfer Transformer (T5) [28] - the basic idea underlying this work is to treat every text processing problem as a “text-to-text” problem, i.e. taking text as input and producing new text as output. The model achieves state-of-the-art results on many benchmarks covering summarization, question answering, text classification, and more. We adopt this approach as our direct baseline by grafting knowledge like entity and coreference information during pre-training by initializing our KAT5 with T5-base checkpoints.

Translation between Augmented Natural Languages (TANL) [23] - a framework to solve several structure predictions in a unified way, with a common architecture and without the need for task-specific modules. The TANL framework is applied to joint entity and relation extraction, named entity recognition, relation classification, semantic role labeling, coreference resolution, event extraction and dialogue state tracking. In all cases the model achieve at least comparable results to the current state-of-the-art, and it achieves new state-of-the-art (SOTA) performance on joint entity and relation extraction (CoNLL04, ADE, NYT, and ACE2005 datasets), relation classification (FewRel and TACRED), and semantic role labeling (CoNLL-2005 and CoNLL-2012). This is our baseline approach for joint entity and relation extraction tasks as the model initializes from the T5-base model like our approach during pre-training.

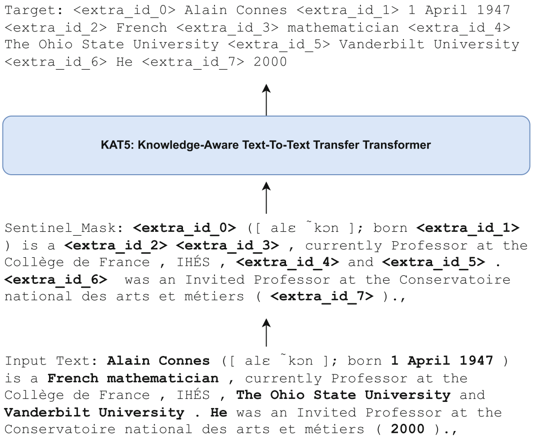
REBEL [13] - a S2S model based on BART-large that performs end-to-end relation extraction for more than 200 different relation types and show that how relation extraction can be simplified by expressing triplets as a sequence of text. We follow the same triplet linearization process of REBEL into our work. REBEL is basically relation extraction-based task-specific model and has shown SOTA performance in most of the relation extraction tasks. Though the model is relation extraction-based and the model parameters are larger than us, here, we compare our KAT5 over the joint relation-extraction tasks.

BART-NAR-BERT (BnB) [33] - a pre-trained non-autoregressive S2S model, which employs BERT as the backbone for the encoder and decoder for natural language understanding and generation tasks. The model outperformed several SOTA models in non-autoregressive benchmark and has shown comparable performance in autoregressive models. Since the model follows a sequence-to-sequence manner, we also compare our model over the generative tasks.

3 Model

3.1 Baseline

We consider the text-to-to transfer transformer – T5 [26] as a baseline encoder-decoder architecture close to the original architecture of transformer [35]. In T5, the input sequence of tokens is mapped to a sequence of embeddings, which is then passed into the encoder. The encoder consists of a stack of "blocks," each comprising two subcomponents: a self-attention layer and a following small feed-forward network. Unlike layer normalization [1], T5 uses a simplified version of layer normalization where the activations are only rescaled and no additive bias is applied. A residual skip connection [32] then adds input from each subcomponent to its output. Dropout [34] is applied within the feed-forward network, on the skip connection, on the attention weights, and at the input and output of the entire stack. The decoder is similar in structure to the encoder, except that it includes a standard attention mechanism after each self-attention layer that attends to the output of the encoder. The self-attention mechanism in the decoder also uses a form of auto-regressive or causal self-attention, which only allows the model to attend to past outputs. The output of the final decoder block is fed into a dense layer with a softmax output, whose weights are shared with the input embedding matrix. All attention mechanisms in the Transformer are split into independent "heads" whose outputs are concatenated before being further processed.



Target: <extra_id_0> Alain Connes <extra_id_1> 1 April 1947 <extra_id_2> French <extra_id_3> mathematician <extra_id_4> The Ohio State University <extra_id_5> Vanderbilt University <extra_id_6> He <extra_id_7> 2000

Sentinel_Mask: **<extra_id_0>** ([alɛ ˜kɔn]; born **<extra_id_1>**) is a **<extra_id_2> <extra_id_3>** , currently Professor at the Collège de France , IHÉS , **<extra_id_4>** and **<extra_id_5>** . **<extra_id_6>** was an Invited Professor at the Conservatoire national des arts et métiers (**<extra_id_7>**).,

Input Text: **Alain Connes** ([alɛ ˜kɔn]; born **1 April 1947**) is a **French mathematician** , currently Professor at the Collège de France , IHÉS , **The Ohio State University** and **Vanderbilt University** . **He** was an Invited Professor at the Conservatoire national des arts et métiers (**2000**).,

Fig. 1. Pre-training tasks of KAT5

3.2 KAT5: Knowledge-Aware Text-to-Text Transfer Transformer

As an unsupervised objective during pre-training, a model needs an objective function that does not require labels but teaches the model generalizable knowledge and will be useful to transfer that knowledge into downstream tasks. Apart from casual language modeling objective for pre-training, recently denoising a.k.a. masked language modeling (MLM) [5] shows better performance and become a standard unsupervised learning objective in many natural language processing (NLP) tasks. In the MLM objective, the model is trained to predict missing or corrupted tokens by adding `<MASK>` in the input sequence. Inspired by BERT's MLM objective, T5 follows a random span masking objective to corrupt 15% of tokens in the input sequence where all consecutive spans of dropped-out tokens are replaced by a single sentinel token, a.k.a. unique mask tokens. We adopt the T5 masking strategies and design an objective that randomly samples and then drops out 15% of entity and coreference related spans in the input sequence using 100 sentinel tokens. Each sentinel token represents a unique mask token starting as $<$ `extra_id_0` $>$, $<$ `extra_id_1` $>$, ..., $<$ `extra_id_99` $>$ for a given input sequence.

Figure 1 shows a knowledge-aware task of KAT5. In this figure, the bold text in the input sequence represents entities where the pronoun `He` indicates the coreference of `Alain connes`. During sentinel masking in KAT5, unique mask tokens are used to corrupt the input text by replacing the entity and coreference spans. Finally, the output sequence consists of the dropped-out entity and coreference spans, delimited by the sentinel tokens used to replace them in the input.

3.3 Pre-training Data Creation

Another key contribution of this paper is to automatically create data for pre-training the KAT5 model. The pre-training data set is a crucial component of the

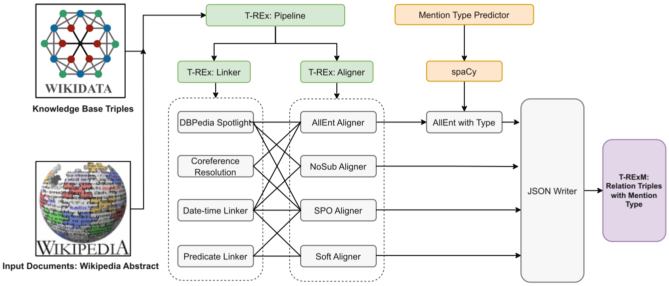


Fig. 2. Knowledge-aware data example aligned from Wikidata

transfer learning pipeline. During pre-training, the model needs a large amount of data that teaches the model generalizable knowledge. The T5 model used the Colossal Clean Crawled Corpus (C4) dataset for pre-training by downloading about 750 GB of text extracted from the Web. In contrast, our KAT5 model is based on integrating knowledge like entity and co-reference information during pre-training; therefore, it is a challenging task to integrate such knowledge in a vast amount of unstructured text like C4 or the entire Wikipedia dump. One possible way is to create such a knowledge-aware pre-training dataset by aligning Wikipedia abstract and Wikipedia hyperlinks with Wikidata entities. We aligned the Wikipedia abstract and Wikidata entities to pre-train the KAT5 model to shed light on this challenging task. We are interested in measuring whether this knowledge-aware dataset can enrich the downstream tasks by leveraging the knowledge during pre-training.

We create the knowledge-aware pre-training dataset by adopting the T-REx implementation[1]. In this implementation, we integrate entity or mention type predictors using the spaCy[2] model to predict all the span types of Wikipedia links. We call T-RExM since an additional mention module is integrated into the T-REx implementation.

T-RExM Details. We follow the T-REx implementation to integrate entity types for creating our T-RExM dataset. T-RExM consists of 11 million triples aligned with 3.09 million Wikipedia abstract which is two orders of magnitude larger than the largest available alignments dataset and covers 2.5 times more predicates. As a source of triples, we use the Wikidata truthy dump[3] containing about 144M triples. Figure 2 shows the knowledge-aware data aligned from Wikipedia. Based on the T-REx implementation, T-RExM follows a pipeline manner, (1) T-REx Linker: it contains the components of Document Reader, Entity Extraction, Coreference resolution, Date-time Linker, and Predicate Linker and (2) T-REx Aligner: it includes AllEnt Aligner, NoSub Aligner, SPO

---

[1] https://github.com/hadyelsahar/RE-NLG-Dataset.

[2] https://spacy.io.

[3] https://dumps.wikimedia.org/wikidatawiki/entities/latest-truthy.nt.bz2.

Aligner, and Soft Aligner. A simple description of main components (among the components) are as follows (for more details we refer to the readers for T-REx creation [7]):

- Document Reader: The T-RExM pipeline is fed with documents from the DBpedia Abstract dataset [2], an open corpus of annotated Wikipedia texts. An English section that contains 4.6M text documents is used. Document Reader includes sentence and word tokenizers to extract the start and end positions of sentences and words in documents.
- Entity Extraction: For each input document, it extracts named entities in the text and links them to their uniform resource identifier (URI) with the DBpedia Spotlight [21] entity linker.
- Coreference Resolution: Stanford CoreNLP co-reference resolution component [20] is used to map a list of possible pronouns of each knowledge base (KB) entity.
- Date and Time Extraction: Stanford temporal tagger Sutime [3] is used to extract temporal expressions and their locations in the documents.
- Predicate Linking: A predicate linker links a sequence of words in a paragraph to its equivalent KB predicate URI if it matches the predicate label or any of its aliases in the KB.
- AllEnt Aligner: Every pair of entities in a sentence is considered in alignment and mapped to their equivalent KB relations.
- SPO Aligner: The subject predicate and object (SPO) aligner aligns triples when the subject and object of a triple are mentioned in a sentence.

During this pipeline manner, we integrate the mention type predictor module on top of AllEnt Aligner to predict the entity type using spaCy model. Based on spaCy model, we align eighteen different entity types[4] and relation labels[5] with 11M triples.

3.4 KAT5 Fine-Tuning

During KAT5 fine-tuning, Like T5, we treat every text processing problem as a test-to-text problem, i.e. giving text as input to the KAT5 model and producing new text as output. Figure 3 shows a fine-tuning approach of KAT5.

4 Experimental Settings

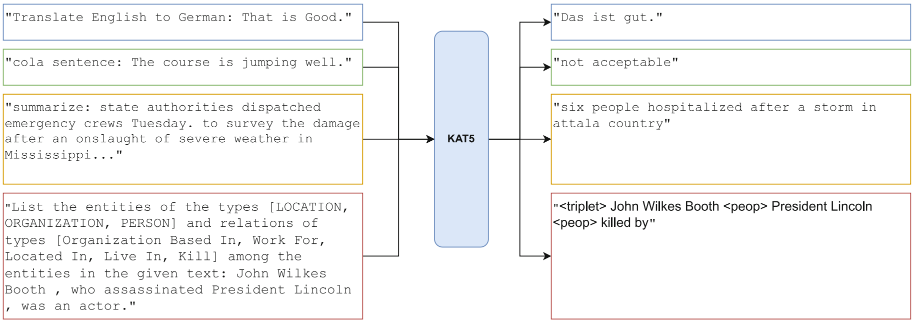
4.1 Datasets

Joint Entity-Relation Extraction Dataset. For joint entity and relation extraction tasks, we evaluate our models on the following datasets.

---

[4] 'CARDINAL', 'DATE', 'EVENT', 'FAC', 'GPE', 'LANGUAGE', 'LAW', 'LOC', 'MONEY', 'NORP', 'ORDINAL', 'ORG', 'PERCENT', 'PERSON', 'PRODUCT', 'QUANTITY', 'TIME', 'WORK_OF_ART'.

[5] Relations label are based on Wikidata properties (https://www.wikidata.org/wiki/Special:ListProperties).



"Translate English to German: That is Good."

"cola sentence: The course is jumping well."

"summarize: state authorities dispatched emergency crews Tuesday. to survey the damage after an onslaught of severe weather in Mississippi..."

"List the entities of the types [LOCATION, ORGANIZATION, PERSON] and relations of types [Organization Based In, Work For, Located In, Live In, Kill] among the entities in the given text: John Wilkes Booth , who assassinated President Lincoln , was an actor."

"Das ist gut."

"not acceptable"

"six people hospitalized after a storm in attala country"

"<triplet> John Wilkes Booth <peop> President Lincoln <peop> killed by"

Fig. 3. Fine-tuning KAT5

CONLL04. The CoNLL04 [29] dataset consists of sentences extracted from news articles - with four entity types `location`, `organization`, `person`, and `other`, and five relation types (`work for`, `kill`, `organization based in`, `live in`, and `located in`. We use the 922/231/288 sentences in the train/validation/test set based on the split by Gupta [11].

ADE. The ADE [12] dataset consists of 4,272 sentences extracted from medical reports– with two `drug` and `disease` entity types and a single relation type `effect`. This dataset has sentences with nested entities. We follow the same settings as TANL [24], conduct a 10-fold cross-validation, and report the average macro-F1 results across all ten splits.

NYT. The NYT dataset [38] is based on the New York Times corpus where we use the preprocessed version of Yu [37]. It consists of three entity types `location`, `organization`, `person` and 24 relation types (such as place of birth, nationality, company etc.). It consists of 56,195/5,000/5,000 sentences in the training/validation/test set.

Summarization Dataset. The XSum [22] and CNNDM datasets are used to evaluate our models.

XSum. Abstractive text summarization aims to produce a short version of a document while preserving its salient information content. We evaluate the models based on the BBC extreme [22] (XSum) dataset. This is a news summarization dataset containing 227K news articles and single-sentence summary pairs. We load the XSum datasets from Huggingface[6] The evaluation metric is ROUGE [17], including ROUGE-1 (R-1), ROUGE-2 (R-2), and ROUGE-L (R-L). We adopted the Google Research re-implementation of ROUGE[7].

---

[6] https://huggingface.co/datasets/EdinburghNLP/xsum.

[7] https://github.com/google-research/google-research/tree/master/rouge.

CNNDM. The CNN/DailyMail (CNNDM) dataset is an English-language dataset containing just over 300k unique news articles as written by journalists at CNN and the Daily Mail. We load the CNNDM datasets from Hugging Face datasets[8] that supports both extractive and abstractive summarization.

Machine Translation Dataset. We evaluate our models using two popular benchmark data sets from the WMT shared tasks on news translation - English (EN) → German (DE) data from WMT 2014 and English→Romanian (RO) data from WMT 2016. We load the WMT datasets from Hugging Face datasets[9,10] and use them directly to train the models without filtering. We evaluate the performance by computing BLEU [25].

4.2 Training

KAT5 Pre-training. To pre-train the KAT5 model, we initialize the model with the T5-base checkpoint[11] and continue pre-training using the knowledge-aware span denoising objective of T5 on the training split of our dataset that was explained in Sect. 3.3. We use a learning rate of 0.001, a linear warm-up of 5k steps, a gradient accumulation of 2 steps, and a maximum sequence length of 512 tokens. The KAT5 model is trained on 1.3B tokens, where we employ a batch size of 65,536 tokens with a maximum step of 200K steps. The original T5 model was trained on 34B tokens over the C4 corpus, which was 26 times larger than our additional pre-training dataset. The KAT5 model is optimized end-to-end using an Adafactor optimizer with a corrupted knowledge-aware span ratio of 15%.

KAT5 Fine-Tuning. During fine-tuning with KAT5 on downstream datasets, we consider two learning settings - (1) Single-task learning: a single model on a single dataset is learned initializing from KAT5 checkpoint. (2) Multi-task learning: Since the model is based on our direct baseline T5 model, therefore, our KAT5 model naturally allows us to train a single model on multiple datasets that can cover many structured prediction tasks. In this setting, we add the dataset name followed by the task separator: (for example, "xsum summarize:" for Xsum dataset) as a prefix to each input sentence and only the summarization datasets are evaluated. We fine-tune on top of KAT5 for a maximum of 10 epochs in all our downstream tasks.

5 Results

In this section, we show that our **K**nowledge-**a**ware T5 (KAT5) can effectively solve the structure prediction tasks that match or exceed the previous state of the

---

[8] https://huggingface.co/datasets/cnn_dailymail.

[9] https://huggingface.co/datasets/wmt14.

[10] https://huggingface.co/datasets/wmt16.

[11] https://huggingface.co/google-t5/t5-base.

Table 1. Performance comparison on the CONLL04, ADE, and NYT datasets. Bold and underlined denotes the best and second-best results within KAT5 and Baseline Models.

Model	Params	CONLL04		ADE		NYT	
		Entity	Relation	Entity	Relation	Entity	Relation
SpERT [6]	110M	88.9	71.5	89.3	78.8	–	–
MRC4ERE [39]		88.9	71.9	–	–	–	–
REBEL [13]	460M	–	71.2	–	81.7	–	91.8
$\text{REBEL}_{\text{pretraining}}$ [13]	460M	–	75.4	–	82.2	–	92.0
- Baseline Model-							
TANL + Single-task [23]	220M	89.4	71.4	90.2	80.6	94.9	90.8
TANL + Multi-dataset [23]	220M	89.8	**72.6**	90.0	80.0	94.7	90.5
TANL + Multi-task [23]	220M	**90.3**	70.0	91.2	**83.8**	94.7	90.7
KAT5 + Single-task	220M	90.0	69.8	**91.9**	81.6	**95.1**	**90.9**

art on multiple datasets. All our experiments start from a pre-trained KAT5 and to keep our model as simple as possible, hyper-parameters are the same across all experiments. In all our experiments we compare our results in two folds: (1) Baseline Model: T5 is the backbone of this model where we directly compare our results and (2) other recent models are applied in the same downstream tasks. To evaluate our model, we adopt TANL [23] evaluation script for joint entity-relation extraction tasks and Hugging Face Transformers evaluation script for summarization and translation tasks.

5.1 Performance on Joint Entity-Relation Extraction

We tackle the joint entity-relation as a generation task where the model output of KAT5 is a triplet that is present in the input text. With the single-task setup in Table 1, the KAT5 outperforms over the TANL which is our direct baseline since TANL framework is initialized with T5 and used the same model parameters. We obtain a +0.6/+1.7/+0.2 and -0.6/+1.0/+0.1 improvement using F1 score in the CONLL4/ADE/NYT datasets for entity and relation extraction tasks respectively. The performance of KAT5 is even better in some tasks in the multi-dataset and multi-task settings. TANL, needs 200 epochs to achieve the stated results in Table 1 where we fine-tune on top of KAT5 for 10 epochs.

In contrast to our baseline approaches, KAT5 shows a better performance over the SpERT [6] and MRC4ERE [39], but shows a little drop in comparison to the Rebel [13] which is a task specific relation extraction model.

Triple Generation Example. Table 2 shows randomly selected generated examples from the ADE dataset. In this table, in the case of Reference 2 and 3 where both T5 and KAT5 models generated the output triple correctly. But in Reference 1, T5 is generated relation type `has effect` and object (<obj>)

Table 2. An example of ADE relation triples output comparing the baseline T5 model output to our KAT5. ☒ and ☑ indicates wrong and correct prediction respectively.

Reference:	1	<triplet> human teratogen <subj> cyclophosphamide <obj> has effect
T5:	☒	<triplet> teratogen <subj> cyclophosphamide <obj> has effect
KAT5:	☑	<triplet> human teratogen <subj> cyclophosphamide <obj> has effect
Reference:	2	<triplet> Lethal anuria <subj> ifosfamide <obj> has effect
T5:	☑	<triplet> Lethal anuria <subj> ifosfamide <obj> has effect
KAT5:	☑	<triplet> Lethal anuria <subj> ifosfamide <obj> has effect
Reference:	3	<triplet> pulmonary toxicity <subj> Gemcitabine <obj> has effect
T5:	☑	<triplet> pulmonary toxicity <subj> Gemcitabine <obj> has effect
KAT5:	☑	<triplet> pulmonary toxicity <subj> Gemcitabine <obj> has effect

Table 3. Performance comparison on the XSum dataset. R-1/2/L stands for ROUGE-1/2/L. Bold and underlined scores denote the best and second-best results.

Model	Params	XSum		
		R-1	R-2	R-L
Transformer [35]		30.7	10.8	24.5
iNAT [14]		27.0	6.9	22.4
NAT [9]		24.0	3.9	20.3
CMLM [8]		23.8	3.6	20.2
LevT [10]		24.8	4.2	20.9
ELMER-Hard [16]		34.5	9.8	26.1
ELMER-Soft [16]		38.3	14.2	29.9
BART [15]		38.8	16.2	30.6
BERT2BERT [30]	220M	37.5	15.2	30.1
BERT-NAR-BERT [33]	220M	32.7	11.6	27.8
BERT-NAR-BERT + additional pre-training [33]	220M	36.1	13.4	30.0
- Baseline Model-				
T5 + fine-tuning + Single-task	220M	39.7	16.5	31.9
KAT5 + Single-task	220M	39.9	16.7	32.1
KAT5 + Multi-task	220M	**40.2**	**17.0**	**32.2**

mention `cyclophosphamide` correctly but failed to generate `human teratogen` mention as subject (<sub>).

5.2 Performance on Summarization

In the multi-task settings of summarization tasks, we add the dataset name followed by the task separator is used (for example, "xsum summarize:" for XSum dataset and "summarize:" for CNNDM dataset) as a prefix to each input sentence. Table 3 shows the abstractive performance comparison of KAT5 over the

Table 4. Performance comparison on the CNN/DailyMail (CNNDM) dataset. Bold and underlined scores denote the best and second-best results within KAT5 and Baseline Models.

Model	Params	CNNDM		
		R-1	R-2	R-L
PTGEN [31]		36.44	15.66	33.42
PTGEN+COV [31]		39.53	17.28	36.38
BERTSUMABS [18]		41.72	19.39	38.76
BERTSUMEXTABS [18]		42.13	19.60	39.18
ROBERTASHARE [30]		40.31	18.91	37.62
BART [15]		44.16	21.28	40.90
- Baseline Model-				
T5 [28]	220M	–	19.24	–
T5 (Re-run)	220M	39.40	17.10	36.57
KAT5 + Single-task	220M	**43.51**	**20.64**	**40.66**
KAT5 + Multi-task	220M	<u>43.44</u>	<u>20.28</u>	<u>40.56</u>

XSum dataset. Both the single- and multi-task settings, the KAT5 outperforms the baseline T5 model. In the original T5 implementation they exclude XSum summarization task. Here, the reported score in Table 3 based on our run by initializing the T5 model. The model also outperforms BART and the recent non-autoregressive BERT-NAR-BERT model. Table 4 shows the performance comparison of KAT5 over the CNNDM dataset. The model shows an improvement over the baseline model but shows a little drop in comparison to the BART model. In the multi-task settings, the performance of CNNDM dataset shows a little drop where XSum dataset shows an improvement in comparison to the single-task settings.

5.3 Performance on Machine Translation

Results of machine translation (MT) experiments are summarized in Table 5. With the single-task setup in MT, the KAT5 model is trained on instruction-based approach and outperforms the baseline T5 that obtains a +0.36/+3.01 improvement using BLEU score in the EN-DE/EN-RO WMT datasets respectively. We also compare our model with the auto-regressive BERT2BERT S2S model and achieve +2.21/+6.75 improvement. Besides, our model also shows an improvement over the vanilla transformer in both EN-DE and EN-RO datasets.

In contrast, we also compare our model on the recent non-autoregressive S2S model like BERT-NAR-BERT. The KAT5 model outperforms all formats of the BERT-NAR-BERT model. Interestingly, our model even shows better performance over the BERT-NAR-BERT initialized with multi-lingual BERT (mBERT) and trained together with original and distilled data from WMT 2014

Table 5. Machine translation experiment results in BLEU scores. Bold and underlined scores denote the best and second-best results.

Model	Params	EN - DE	EN - RO
Transformer [35]		27.30	21.53
BERT2BERT + mBERT [30]	220M	25.80	23.24
BERT-NAR-BERT + random [33]	220M	7.15	4.12
BERT-NAR-BERT + mBERT [33]	220M	6.81	5.92
BERT-NAR-BERT + mBERT + distilled [33]	220M	27.49	18.94
- Baseline Model-			
T5 [28]	220M	<u>27.65</u>	<u>26.98</u>
KAT5	220M	**28.01**	**29.99**

(German) and WMT 2016 (Romanian). We obtain a +0.52/ +11.05 improvement in the EN-DE/EN-RO WMT datasets respectively.

6 Discussion

We present **K**nowledge-**a**ware T5 (KAT5), a novel, simple, and easy-to-implement S2S model by leveraging T5 checkpoint during pre-training. We demonstrate strong performances of joint entity-relation extraction in three datasets (ADE, CONLL04, and NYT), XSum and CNNDM in summarization tasks, and English (EN) → German (DE) and English→Romanian (RO) in machine translation.

KAT5 is a budget training approach since it needs 10 epochs that can achieve similar or somewhat better performance over the each CONLL04, NYT, and ADE datasets where TANL set 200 epochs to achieve the reported score in Table 1 for all the entity-relation extraction datasets. In contrast, the task-specific relation extraction model like REBEL, where the model fine-tunes on top of REBEL for a maximum of 30/42/25 epochs for CONLL04/NYT/ADE datasets respectively.

Most SOTA models train the models for a longer time by setting early stopping criteria, however, this can lead to long and expensive training times. Our experiments show that training for fewer epochs may lead to a small decrease in performance, but it brings the benefit of a more affordable training time. Besides, it also shows that knowledge-aware pre-training is very effective to facilitate the downstream tasks by transferring the pre-training knowledge into the fine-tuning tasks.

7 Conclusion

This paper introduces an efficient **K**nowledge-**a**ware T5 (KAT5) S2S method with encoders and decoders that integrates entities and their coreferences as knowledge during pre-training. To introduce such knowledge-aware approach, we perform large-scale alignments between Wikipedia abstract and Wikidata triples to facilitate our pre-training KAT5 model by leveraging T5 model. Experiment results show that the proposed model outperforms baselines in most of the joint entity-relation, summarization, MT tasks. We have also shown its flexibility in adapting to new domains, by training on just a few epochs. In the future, we plan to extend our KAT5 model into a larger parameter model with more knowledge-aware data by leveraging biomedical dataset along with general domain.

Limitations

During knowledge-aware pre-training, we consider using a document-level by loading the T-RExM dataset aligning with Wikipedia abstract and Wikidata triples. Usually the input length is longer and we set the maximum input sequence length to 512 tokens. During pre-training, therefore, One may further train the models for a longer period with a sentence-level corpus of Wikipedia and Wikidata triples to achieve better context representations.

Another limitation is that, our KAT5 is trained on 1.3B tokens that consist of around 11M triples during knowledge-aware pre-training. It is still a challenging task to align more knowledge during pre-training. One possible direction is to align biomedical domain knowledge (like entity, relation, or co-reference) along with general knowledge from Wikipedia, which can further facilitate the pre-training strategy. Furthermore, hyper-parameter tuning on large language models is computationally very costly, we choose 10 epochs for the fine-tuning tasks without conducting a wide-level of empirical studies.

Acknowledgments. This paper is based on results obtained from a project JPNP20006, commissioned by the New Energy and Industrial Technology Development Organization (NEDO).

Ethics Statement. We use only publicly available datasets and relatively low compute amounts while conducting our experiments to enable reproducibility. We do not perform any studies on other humans or animals in this research.

References

1. Ba, J., Kiros, J.R., Hinton, G.E.: Layer normalization. ArXiv **abs/1607.06450** (2016). https://api.semanticscholar.org/CorpusID:8236317
2. Brümmer, M., Dojchinovski, M., Hellmann, S.: DBpedia abstracts: a large-scale, open, multilingual NLP training corpus. In: Calzolari, N., et al. (eds.) Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016), Portorož, Slovenia, pp. 3339–3343, May 2016. https://aclanthology.org/L16-1532

3. Chang, A.X., Manning, C.: SUTime: a library for recognizing and normalizing time expressions. In: Calzolari, N., et al. (eds.) Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC 2012), Istanbul, Turkey, May 2012, pp. 3735–3740 (2012). http://www.lrec-conf.org/proceedings/lrec2012/pdf/284_Paper.pdf
4. Conneau, A., Lample, G.: Cross-lingual language model pretraining. In: Advances in Neural Information Processing Systems, vol. 32, pp. 7059–7069 (2019). https://proceedings.neurips.cc/paper/2019/file/c04c19c2c2474dbf5f7ac4372c5b9af1-Paper.pdf
5. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: Burstein, J., Doran, C., Solorio, T. (eds.) Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Minneapolis, Minnesota, June 2019, pp. 4171–4186. ACL (219). https://aclanthology.org/N19-1423
6. Eberts, M., Ulges, A.: Span-based joint entity and relation extraction with transformer pre-training. CoRR (2019). http://arxiv.org/abs/1909.07755
7. Elsahar, H., et al.: T-REx: a large scale alignment of natural language with knowledge base triples. In: Calzolari, N., et al. (eds.) Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018). European Language Resources Association (ELRA), Miyazaki, Japan, May 2018. https://aclanthology.org/L18-1544
8. Ghazvininejad, M., Levy, O., Liu, Y., Zettlemoyer, L.: Mask-predict: parallel decoding of conditional masked language models. In: Inui, K., Jiang, J., Ng, V., Wan, X. (eds.) Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 6112–6121, Hong Kong, China, November 2019. Association for Computational Linguistics (2019). https://aclanthology.org/D19-1633
9. Gu, J., Bradbury, J., Xiong, C., Li, V.O., Socher, R.: Non-autoregressive neural machine translation. In: International Conference on Learning Representations (2018). https://doi.org/10.48550/arXiv.1711.02281
10. Gu, J., Wang, C., Zhao, J.: Levenshtein transformer. In: Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R. (eds.) Advances in Neural Information Processing Systems, vol. 32. Curran Associates, Inc. (2019). https://proceedings.neurips.cc/paper_files/paper/2019/file/675f9820626f5bc0afb47b57890b466e-Paper.pdf
11. Gupta, P., Schütze, H., Andrassy, B.: Table filling multi-task recurrent neural network for joint entity and relation extraction. In: Matsumoto, Y., Prasad, R. (eds.) Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics, pp. 2537–2547. https://aclanthology.org/C16-1239
12. Gurulingappa, H., Rajput, A.M., Roberts, A., Fluck, J., Hofmann-Apitius, M., Toldo, L.: Development of a benchmark corpus to support the automatic extraction of drug-related adverse effects from medical case reports. J. Biomed. Inform. **45**(5), 885–892 (2012). https://www.sciencedirect.com/science/article/pii/S1532046412000615

13. Huguet Cabot, P.L., Navigli, R.: REBEL: relation extraction by end-to-end language generation. In: Moens, M.F., Huang, X., Specia, L., Yih, S.W.t. (eds.) Findings of the Association for Computational Linguistics: EMNLP 2021, pp. 2370–2381. ACL. https://aclanthology.org/2021.findings-emnlp.204
14. Lee, J., Mansimov, E., Cho, K.: Deterministic non-autoregressive neural sequence modeling by iterative refinement. In: Riloff, E., Chiang, D., Hockenmaier, J., Tsujii, J. (eds.) Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October–November 2018, pp. 1173–1182. Association for Computational Linguistics (2018). https://aclanthology.org/D18-1149
15. Lewis, M., et al.: BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In: Jurafsky, D., Chai, J., Schluter, N., Tetreault, J. (eds.) Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 7871–7880. ACL, July 2020. https://aclanthology.org/2020.acl-main.703
16. Li, J., Tang, T., Zhao, W.X., Nie, J.Y., Wen, J.R.: ELMER: a non-autoregressive pre-trained language model for efficient and effective text generation. In: Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pp. 1044–1058, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics (2022). https://aclanthology.org/2022.emnlp-main.68
17. Lin, C.Y.: ROUGE: a package for automatic evaluation of summaries. In: Text Summarization Branches Out, Barcelona, Spain, July 2004, pp. 74–81. ACL (2004). https://aclanthology.org/W04-1013
18. Liu, Y., Lapata, M.: Text summarization with pretrained encoders. In: Inui, K., Jiang, J., Ng, V., Wan, X. (eds.) Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hong Kong, China, November 2019, pp. 3730–3740. ACL (2019). https://aclanthology.org/D19-1387
19. Liu, Y., et al.: RoBERTa: a robustly optimized BERT pretraining approach. CoRR **abs/1907.11692** (2019). http://arxiv.org/abs/1907.11692
20. Manning, C., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S., McClosky, D.: The Stanford CoreNLP natural language processing toolkit. In: Bontcheva, K., Zhu, J. (eds.) Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, Baltimore, Maryland, June 2014, pp. 55–60. ACL (2014). https://aclanthology.org/P14-5010
21. Mendes, P.N., Jakob, M., García-Silva, A., Bizer, C.: DBpedia spotlight: shedding light on the web of documents. In: Proceedings of the 7th International Conference on Semantic Systems, I-Semantics 2011, New York, NY, USA, pp. 1–8. Association for Computing Machinery (2011). https://doi.org/10.1145/2063518.2063519
22. Narayan, S., Cohen, S.B., Lapata, M.: Don't give me the details, just the summary! Topic-aware convolutional neural networks for extreme summarization. In: Riloff, E., Chiang, D., Hockenmaier, J., Tsujii, J. (eds.) Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 1797–1807. ACL, Brussels, Belgium, Oct-Nov 2018 (2018). https://aclanthology.org/D18-1206
23. Paolini, G., et al.: Structured prediction as translation between augmented natural languages. In: 9th International Conference on Learning Representations, ICLR 2021 (2021)
24. Paolini, G., et al.: Structured prediction as translation between augmented natural languages. CoRR **abs/2101.05779** (2021). https://arxiv.org/abs/2101.05779

25. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: BLEU: a method for automatic evaluation of machine translation. In: Isabelle, P., Charniak, E., Lin, D. (eds.) Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, Philadelphia, Pennsylvania, USA, July 2002, pp. 311–318. ACL (2002). https://aclanthology.org/P02-1040
26. Radford, A., Narasimhan, K., Salimans, T., Sutskever, I., et al.: Improving language understanding by generative pre-training. OpenAI Blog (2018). https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf
27. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I.: Language models are unsupervised multitask learners. OpenAI blog **1**(8), 9 (2019). https://d4mucfpksywv.cloudfront.net/better-language-models/language_models_are_unsupervised_multitask_learners.pdf
28. Raffel, C., et al.: Exploring the limits of transfer learning with a unified text-to-text transformer. J. Mach. Learn. Res. **21**(140), 1–67 (2020). http://jmlr.org/papers/v21/20-074.html
29. Roth, D., Yih, W.t.: A linear programming formulation for global inference in natural language tasks. In: Proceedings of the Eighth Conference on Computational Natural Language Learning (CoNLL 2004) at HLT-NAACL 2004, pp. 1–8, Boston, Massachusetts, USA, May 6 - May 7 2004. Association for Computational Linguistics (2004). https://aclanthology.org/W04-2401
30. Rothe, S., Narayan, S., Severyn, A.: Leveraging pre-trained checkpoints for sequence generation tasks. Trans. Assoc. Comput. Linguist. **8**, 264–280 (2020). https://aclanthology.org/2020.tacl-1.18
31. See, A., Liu, P.J., Manning, C.D.: Get to the point: summarization with pointer-generator networks. In: Barzilay, R., Kan, M.Y. (eds.) Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 1073–1083, Vancouver, Canada, July 2017. Association for Computational Linguistics (2017). https://aclanthology.org/P17-1099
32. Shafiq, M., Gu, Z.: Deep residual learning for image recognition: a survey. Appl. Sci. **12**(18) (2022). https://www.mdpi.com/2076-3417/12/18/8972
33. Sohrab, M.G., Asada, M., Rikters, M., Miwa, M.: BERT-NAR-BERT: a non-autoregressive pre-trained sequence-to-sequence model leveraging BERT checkpoints. IEEE Access **12**, 23–33 (2024). https://doi.org/10.1109/ACCESS.2023.3346952
34. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. **15**(56), 1929–1958 (2014). http://jmlr.org/papers/v15/srivastava14a.html
35. Vaswani, A., et al.: Attention is all you need. Adv. Neural Inf. Process. Syst. **30** (2017). https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf
36. Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R.R., Le, Q.V.: XLNet: generalized autoregressive pretraining for language understanding. In: Advances in Neural Information Processing Systems, vol. 32, pp. 5753–5763 (2019). https://dl.acm.org/doi/pdf/10.5555/3454287.3454804
37. Yu, B., et al.: Joint extraction of entities and relations based on a novel decomposition strategy. CoRR **abs/1909.04273** (2019). http://arxiv.org/abs/1909.04273

38. Zeng, X., Zeng, D., He, S., Liu, K., Zhao, J.: Extracting relational facts by an end-to-end neural model with copy mechanism. In: Gurevych, I., Miyao, Y. (eds.) Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Melbourne, Australia, July 2018, pp. 506–514. ACL. https://aclanthology.org/P18-1047
39. Zhao, T., Yan, Z., Cao, Y., Li, Z.: Asking effective and diverse questions: a machine reading comprehension based framework for joint entity-relation extraction. In: Bessiere, C. (ed.) Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, pp. 3948–3954, IJCAI 2020 (2020). https://doi.org/10.24963/ijcai.2020/546, main track

Asymmetric Graph-Based Deep Reinforcement Learning for Portfolio Optimization

Haoyu Sun[1], Xin Liu[1], Yuxuan Bian[2], Peng Zhu[1], Dawei Cheng[1,4](✉),
and Yuqi Liang[3]

[1] Department of Computer Science, Tongji University, Shanghai, China
`{the_shy,2051277,pengzhu,dcheng}@tongji.edu.cn`
[2] The Chinese University of Hong Kong, Hong Kong, China
[3] Seek Data Group, Emoney Inc., Shanghai, China
`roly.liang@seek-data.com`
[4] Shanghai Artificial Intelligence Laboratory, Shanghai, China

Abstract. In recent years, existing studies have sought to enhance the effectiveness of portfolio optimization by modeling asset relations. However, employing conventional graph neural network methodologies for effective aggregation and final representation learning of intricately complex financial information within real-world markets proves challenging. This necessitates the optimization of graph structures to enhance the accuracy of parsing and leveraging financial information. In this paper, we propose an asymmetric graph-based deep reinforcement learning for portfolio optimization. Specifically, leveraging the excellent evaluative capabilities of large language models, we decipher multi-dimensional asymmetric relationships between stocks in multi-dimensional data, constructing asymmetric stock relationship graphs based on news and sectors. We then design a multi-dimensional relationship attention mechanism to jointly represent asymmetric graph information and employ deep reinforcement learning for end-to-end portfolio optimization. Extensive experiments on real datasets from China and the United States have demonstrated the superiority of our method over existing state-of-the-art methods. In the industrial observation conducted at a leading financial technology company, we validated the applicability of our method in real-world market scenarios.

Keywords: Graph Neural Networks · Deep Reinforcement Learning · Portfolio Optimization

1 Introduction

Portfolio management involves the diversified management of assets through the selection of multiple stocks based on the investor's requirements, aiming to achieve risk diversification and enhance returns [18]. This holds significant

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
A. Bifet et al. (Eds.): ECML PKDD 2024, LNAI 14949, pp. 174–189, 2024.
https://doi.org/10.1007/978-3-031-70378-2_11

academic research value and finds extensive applications in the financial domain, representing one of the focal points in the field of data science. Presently, existing studies [11,16] have demonstrated that effectively modeling relations between stocks can enhance the performance of portfolio optimization models.

Stocks and their associated enterprises exist within a complex market environment, leading to a diversity of structural relationships among enterprises, including investments, guarantees, and supply chains [17]. Accurately extracting and effectively modeling complex relationships for application in portfolio optimization poses significant challenges [20]. Previous research has explored various approaches to modeling complex relations in the context of stocks. For example, RAT [21] models stock relations using financial leverage, while DeepTrader [19] proposes methods such as feature correlations, causal relations, and industry relations for modeling stock relations. There is also research [6] employing large language models for the analysis and modeling of stocks and markets.

However, these studies are confined to theoretically symmetric graph structures of stock relationships. In asymmetric graph structures, their applicability diminishes due to weakened information propagation and aggregation, resulting in suboptimal outcomes. In reality, financial relationships are not bidirectionally symmetric but manifest unidirectional and asymmetric influences. For instance, leading or large-scale enterprises often play a leading role in steering the overall development of their respective sectors. Similarly, in the stock market, the broader market is significantly influenced by large-cap enterprises [3]. While some studies [20,22] have considered moving beyond pre-defined symmetric relations, they are predominantly confined to the use of a single information source, thus lacking a comprehensive description of relationships between stocks.

As is well known, the utilization of multiple information sources enhances the market's perceptual capabilities [1,28]. However, constructing a multi-source asymmetric relationship for portfolio optimization still poses significant difficulties and challenges. This is because the incorporation of multiple information sources also introduces more noise and redundant information, creating complications for joint representation and subsequent portfolio optimization. While Deep Reinforcement Learning (DRL) provides a foundational paradigm for direct optimization learning, and there are studies [16,18,19,21] employing DRL for portfolio optimization, maintaining the simplicity and high accuracy of information in the graph structure is crucial. The intricate environment of real-world financial markets further amplifies the adverse effects on model training caused by suboptimal information extraction and representation, thereby impacting the ultimate optimization of investment portfolios.

Therefore, we proposed a Asymmetric Graph-based Deep Reinforcement Learning (AGDRL), which can be applied to optimize portfolio management in the context of multi-dimensional data sources. We leverage the advanced knowledge reasoning capabilities of large language models (LLM) to identify and extract meaningful information from noisy multi-dimensional data sources. Subsequently, we construct multi-dimensional asymmetric stock relation graphs, employing joint graph learning of multi-dimensional information and temporal feature information. Within the framework of DRL, we directly optimize investment strategies to achieve superior investment outcomes, including increased returns and reduced risks. In the evaluation conducted within industrial settings, our approach showcased its applicability to real-world trading markets. In conclusion, our contributions can be summarized as follows:

- We propose a novel method that leverages deep reinforcement learning to learn from multi-dimensional asymmetric relationship graphs between stocks, and applies it directly to portfolio optimization.
- We employ large language models to parse fundamental information of enterprises and financial news, thereby constructing multi-dimensional stock relation graphs. Subsequently, we devise a multi-dimensional relationship attention mechanism to learn the graph representation of asymmetric relations. Finally, we employ deep reinforcement learning to accomplish portfolio optimization, with the objective of enhancing returns and mitigating risk.
- Extensive experiments on two real-world datasets have conclusively demonstrated the superior effectiveness of our method compared to state-of-the-art baselines. Additionally, our work was evaluated in a leading financial service provider of China, affirming the adaptability of our method to real market scenarios.

2 Related Work

Exploring and modeling stock relationships represent effective strategies in enhancing the optimization of investment portfolios [12,17]. Existing research [4,11,23] has made many attempts in this field, such as THGNN [20], which employs temporal heterogeneous graphs to characterize stock relationships, and CGM [28], which introduces news sentiment to construct short- and long-term stock relations. However, these methods often focus on predicting stock trends before devising investment strategies, potentially leading to suboptimal results [8]. The DRL based approaches demonstrate advantages in portfolio optimization by enabling end-to-end optimization of investment strategies. Currently, there are various related attempts [2,15], such as those based on news representation [25], dynamic programming [5], and exploration of other factors [10,24,27]. Although many DRL-based approaches [16,19,21] have the capability to directly optimize investment strategies, such as AlphaStock [18] utilizing attention mechanisms to capture inter-stock relationships and end-to-end optimizing investment returns, they assume symmetric relationships when modeling stock relations. HIST [22] made changes, attempting to introduce hidden dynamic factors into the predefined symmetric relationships. However, symmetric graph frameworks fail to capture the unidirectional influence among stocks, leading to increased interference in information aggregation for optimization. Our method is dedicated to effectively leveraging an asymmetric graph framework to learn the unidirectional influence among stocks and applying it to portfolio optimization.

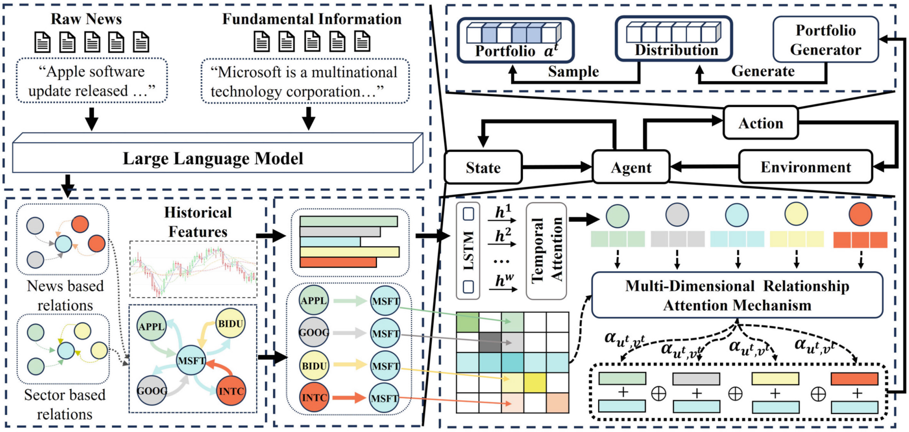


Fig. 1. The overall architecture of the proposed AGDRL. Our method leverages asymmetric graphs to aggregate multidimensional information and employs a deep reinforcement learning framework for end-to-end optimization of investment strategies.

3 The Proposed Method

3.1 Problem Formulation

The objective of portfolio management is to maximize investor returns by observing the market, specifying appropriate investment strategies, and adjusting the portfolio allocation accordingly. It is applicable to the Markov Decision Process $M = \{\mathbf{S}, \mathbf{A}, P, R\}$, where $\mathbf{S}, \mathbf{A}, P, R$ stand for market state, investment action, portfolio rebalancing strategy, and investment reward function, respectively. For time t, the agent decides on the investment portfolio action $\boldsymbol{a}_t \in \mathbf{A}$, and then transitions the state to $\mathbf{s}_{t+1} \in \mathbf{S}$ based on the transition distribution $\mathbf{s}_{t+1} \sim P(\mathbf{s}_{t+1} \mid \mathbf{s}_t, \boldsymbol{a}_t)$. Throughout the investment period, the agent engages in a repeated interactive process to learn the optimal strategy π that maximizes expected returns $J = E_{P(\tau)}[\sum_{t=1}^{|\tau|} \gamma^{t-1} r_t]$, where τ stands for the interaction trajectory and γ for the discounted factor.

3.2 Framework Overview

The proposed AGDRL is a deep reinforcement learning framework designed to efficiently extract and jointly learn from multi-dimensional asymmetric graph information, optimizing investment returns. The structural framework is illustrated in Fig. 1. The State section denotes the data processing procedure; our model has three input sources, namely raw news, fundamental information of enterprises, and time-series features of stocks. The Agent section represents the main network structure for the entire model training and guides the update of the State. The Action part signifies the process of portfolio generation. The Environment interacts with the Agent and Action, optimizing the Agent based on the feedback generated from the portfolio creation strategy.

3.3 Construction of Asymmetric Graphs

The construction of the asymmetric graph consists of two components: the news-based graph $\mathrm{G_n} = (V, E_n)$ and the sector-based graph $\mathrm{G_s} = (V, E_s)$, both of which are updated daily, where V represent stocks, E_n and E_s represent relations between stocks based on news and sectors. Leveraging the capabilities of large language models, we automate the parsing of news text and fundamental information about companies. Specifically, we employ *gpt-3.5 turbo* and design a set of prompts to analyze the relevance between stocks and textual information.

We retrieve relevant news data for the target dataset through the data interface of FinGPT[1] and extract pertinent fundamental information from the introduction on the company's homepage. The construction of $\mathrm{G_n}$ and $\mathrm{G_s}$ is similar; here, we illustrate the construction of $\mathrm{G_n}$ as an example. For each piece of news, the large language model is utilized to assess the degree of influence $r \in (0, 1)$ on stocks associated with it. For a given news event N, the influence degree is used to calculate an asymmetric relationship: the impact degree of stock x is denoted as d_x^N, the impact degree of stock y is denoted as d_y^N, and the impact relationship I_{xy}^N from stock x to y is computed as $I_{xy}^N = d_x^N / d_y^N$. The nodes $V = \{v^{t_1}, v^{t_2}, \ldots, v^T\}$ and edges $E_n = \{e_n^{t_1}, e_n^{t_2}, \ldots, e_n^T\}$ in the news graph dynamically reflect the temporal changes of the asset relations. The weight of the temporal edge e_n^t can be calculated as: $\forall x, y \in v^t$, $\max\limits_{N \in N_t^{l_w^n}} I_{xy}^N$, where $N_t^{l_w^n}$ represents all the news from time t to $t - l_w^n$, and l_w^n represents the look-back window length of news. If the weight is greater than 1, we set it to 1, as the influence from other stocks cannot exceed the impact of a stock on itself.

3.4 Multi-dimensional Relation Representation

For the temporal features of stocks, we employ a Long Short-Term Memory with History state Attention (LSTM-HA) to capture feature relationships within the look-back window length of features l_w^f and represent them. It takes $\mathbf{X}_{in} \in \mathbb{R}^{N \times d_{in} \times T}$ for N stocks with d_{in} features in T days as the initial input. $\mathbf{X}_{in}$ can be represented as a sequence $\{\mathbf{X}^1, \ldots, \mathbf{X}^k, \ldots, \mathbf{X}^{l_w^f}\}$, representing the temporal features. We start by employing a recurrent neural network, specifically LSTM, to capture the temporal feature representation:

$$\mathbf{H}_k^t = \mathrm{LSTM}\,(\mathbf{H}_{k-1}^t, \mathbf{X}^k), k \in [1, l_w^f], \tag{1}$$

where $\mathbf{H}_k^t \in \mathbb{R}^{N \times d_{hid}}$ denotes the hidden state encoded by LSTM at step k and d_{hid} represents the dimension of the hidden layer. Subsequently, we utilize a historical state attention mechanism to capture correlations across multiple hidden layers, thereby enhancing the availability of information and mitigating the loss of intermediate layer information:

---

[1] https://github.com/AI4Finance-Foundation/FinGPT.

$$e_t = \tanh\left([\mathbf{H}_k^t, \mathbf{H}_{l_w^f}^t]\mathbf{W}_1 + \mathbf{X}^{l_w^f}\mathbf{W}_2\right)\boldsymbol{V}_e^{\mathrm{T}}, \tag{2}$$

$$\alpha_k = \frac{\exp(e_k)}{\sum_{i=1}^{l_w^f}\exp(e_t)}, \quad \mathbf{E}^t = \sum_{k=1}^{l_w^f}\alpha_k\mathbf{H}_{l_w^f}^t, \tag{3}$$

where e_t is the score computed for the hidden layer $\mathbf{H}_k^t$ at time t, α_k represents the attention weight between hidden layers $\mathbf{H}_k^t$ and $\mathbf{H}_{l_w^f}^t$, $\mathbf{W}_1$ and $\mathbf{W}_2$ are both the projection weights, $\boldsymbol{V}_e$ is the weight vector. $\mathbf{E}^t \in \mathbb{R}^{N\times d_{hid}}$ denotes the feature representation of all stocks after applying the attention mechanism.

After completing the feature representation for a single stock, we designed a Multi-Dimensional Attention Mechanism to aggregate multi-dimensional relational information of stocks. We first calculate the score α_{x^t,y^t}^i for the edge (x^t, y^t), representing the importance of the one-way edge from stock x to stock y of attention head i at time t:

$$\alpha_{x^t,y^t}^i = \frac{\exp\left(\mathrm{LeakyRelu}(\boldsymbol{a}_{g,i}^{\mathrm{T}}[\boldsymbol{e}_{x^t} \parallel \boldsymbol{e}_{y^t}])\right)}{\sum_{k^t\in v^t}\exp\left(\mathrm{LeakyRelu}(\boldsymbol{a}_{g,i}^{\mathrm{T}}[\boldsymbol{e}_{k^t} \parallel \boldsymbol{e}_{y^t}])\right)}, \tag{4}$$

where $\boldsymbol{e}_{x^t} \in \mathbb{R}^{d_{hid}}$ represents the feature embedding of stock x in time t, $\boldsymbol{a}_{g,i} \in \mathbb{R}^{2d_{hid}}$ denotes the i-th head trainable graph attention weight vector of the specific graph $g \in \{n, s\}$, n representing news and s representing sector. The output $\mathbf{E}_{head_g^i}^t \in \mathbb{R}^{N\times d_{hid}}$ of head i for the graph g can be represented as follows:

$$\mathbf{E}_{head_g^i}^t = \sum_{y^t\in v^t}\sigma\Big(\sum_{x^t\in v^t}\alpha_{x^t,y^t}^i\boldsymbol{e}_{x^t}\boldsymbol{w}_{x^t,y^t}^g\Big), \tag{5}$$

where σ represents the activation function, $\boldsymbol{w}_{x^t,y^t}^g$ represents the weight of the edge (x^t, y^t) in the graph g. The information aggregation is formulated as follows:

$$\mathbf{E}_g^t = \mathrm{Concat}\left(\mathbf{E}_{head_g^1}^t, \ldots, \mathbf{E}_{head_g^{n_h}}^t\right)\mathbf{W}_{g,o}, \tag{6}$$

$$\mathbf{E}_{con}^t = \mathrm{Concat}\left(\mathbf{E}_n^t + \mathbf{E}^t, \mathbf{E}_s^t + \mathbf{E}^t\right) \tag{7}$$

where $\mathbf{E}_g^t$ means the representation aggregated from the non-symmetric relations in graph g, $\mathbf{W}_{g,o}$ represents the attention output projection matrix, n_h represents the number of attention heads, and $\mathbf{E}_{con}^t$ is the final representation aggregated from the news graph $\mathrm{G_n}$ and the sector graph $\mathrm{G_s}$.

3.5 Generator for Portfolio Optimization

We designed a portfolio generator that generates investment distributions based on the final representations of stocks:

$$\boldsymbol{p}_t = \mathbf{W}_p\mathbf{E}_{con}^t + \boldsymbol{b}_p, \tag{8}$$

where $\boldsymbol{p}_t \in \mathbb{R}^{N\times 1}$ represents the generated investment scores, $\mathbf{W}_p$ and $\boldsymbol{b}_p$ are both trainable parameters. Subsequently, we acquire the ultimate investment portfolio, denoted as action, through a sampling process:

$$\mathcal{D}_t = \text{Multinomial}(\boldsymbol{p}_t, N_a), \tag{9}$$

$$\boldsymbol{a}_t^i = \frac{\boldsymbol{p}_t^i}{\sum\limits_{k\in\mathcal{D}_t} p_t^k}, \quad i = 1, 2, \ldots, N_a, \tag{10}$$

where N_a represents the number of stocks for investment. Following the trading rules of qlib[2], we purchase 10% of the total number of stocks. $\mathcal{D}_t$ represents the stocks sampled based on probabilities, $\boldsymbol{p}_t^i$ signifies the investment score for stock i, and $\boldsymbol{a}_t^i$ represents the investment proportion for stock i in time t. This enhances the model's exploration capabilities, thereby increasing the potential for adopting diverse trading strategies. We then employ Actor-Critic reinforcement learning framework to optimize the whole agent model.

Critic Training. Before sampling, the reward function G_t is calculated as:

$$G_t = \frac{\sum_{k=t}^{T} \gamma^{k-t} r_k}{\max\limits_{\tau\in[1,T]} \left(\max\limits_{\lambda\in[1,\tau]} \left(\frac{R_\lambda - R_\tau}{R_\lambda}\right)\right)} \tag{11}$$

where r_t represents the portfolio reward of action $\boldsymbol{a}_t$ and R_τ means the cumulative investment returns at moment τ. We define the reward function as the calmar ratio instead of maximizing returns. The calmar ratio is a comprehensive metric that balances both returns and risks, preventing the model from overly pursuing returns at the expense of ignoring market risks. Then we perform sampling for a designated number of iterations. The MSE objective function of the critic for the sampled mini-batch $\hat{\tau}$ is defined as follows:

$$J(\mu) = \frac{1}{|\hat{\tau}|} \sum_{t=1}^{|\hat{\tau}|} (\text{Critic}(\mathbf{X}^t), G_t)^2 \tag{12}$$

And we use Adam to optimize the critic network parameters μ.

Actor Training. Firstly, we calculate the advantage function A_t of the original trajectory τ as follows:

$$A_t = G_t - \text{Critic}(\mathbf{s}_t), \tag{13}$$

which is regarded as the excess reward of the current state $\mathbf{s}_t$ and action $\boldsymbol{a}_t$. Then we calculate the objective function J_{clip}^{θ} for specific sampled mini-batch $\hat{\tau}$ of the actor as follows:

$$r_t^{\theta_{old}}(\theta) = \frac{\pi_\theta(\boldsymbol{a}_t|\mathbf{s}_t)}{\pi_{\theta_{old}}(\boldsymbol{a}_t|\mathbf{s}_t)}, \tag{14}$$

$$J_{clip}^{\theta} = \sum_{(\mathbf{s},\boldsymbol{a})\in\hat{\tau}} \min(r_t^{\theta_{old}}(\theta) A_t, \text{clip}(r_t^{\theta_{old}}(\theta), 1-\epsilon, 1+\epsilon) A_t), \tag{15}$$

---

[2] https://github.com/microsoft/qlib.

Algorithm 1. AGDRL Training Algorithm

```
Input: Initial Actor parameters $\theta = \theta_{old}$, initial exploration Actor parameters $\pi^0_{old}$,
    initial Critic parameters $\mu$, batch size $L$, re-sampling times $M$
for $k = 0, 1, 2, \dots$ do
    collect the interaction trajectory $\tau$ by running $\pi^k_{old}$ for $T$ time-steps
    Compute discounted cumulative rewards $G_1, \dots, G_T$
    Compute advantage estimates $A_1, \dots, A_T$
    for $i = 1, 2, \dots, len(\tau) * M/L$ do
        Sample the trajectory $\tau$ and gain $mini\text{-}batch_i$
        Compute the current portfolio probability distribution $\boldsymbol{a}^1_t, \dots, \boldsymbol{a}^{N_a}_t$
        Compute the divergence $r^{\theta_{old}}_t(\theta)$ between the exploration actor and the target
            optimized actor
        Update Actor by optimizing the clip policy surrogate $\theta = \arg\max_{\theta} \frac{1}{|L|} J^{\theta}_{\text{clip}}$
        Update Critic by optimizing $\arg\max_{\mu} \frac{1}{|L|} J(\mu)$
    end for
    $\theta_{\text{old}} \leftarrow \theta$
end for
```

where the $r^{\theta_{old}}_t(\theta)$ means the divergence between the exploration actor $\pi_{\theta_{old}}$ [14] and the target optimized actor π_θ, and the clip loss function J^{θ}_{clip} limits the update range of the policy network into $[1 - \epsilon, 1 + \epsilon]$, to avoid excessive adjustments to the strategy and thus achieve a better model fit. Finally, we use Adam optimizerto update the parameters of our proposed actor network. The use of two objective functions facilitates the joint optimization of the agent, resulting in faster convergence and enhanced training stability. The complete AGDRL training algorithm is outlined in Aalgorithm 1.

4 Experiments

4.1 Experimental Setup

Datasets. Our dataset includes 30 constituents of the Shanghai Stock Exchange (SSE50) from the A-share market in China and 80 constituents of the National Association of Securities Dealers Automated Quotations (NDX100) from the U.S. market. The training and validation sets for our experiments cover the period from 2016 to 2020, while the test set spans from 2021 to 2022. We aim to assess the long-term profitability and stability of the model.

Experiment Settings. The training process was run on a server with 32G memory and single NVIDIA Tesla V100 GPU. The entire training process needs about 8 h. We employ a daily trading strategy, where at the beginning of each trading day, stocks are bought according to the investment strategy. At the end of the trading day, stocks are sold, and the profit or loss from day t is reinvested as purchasing capital for day $t+1$. In terms of parameters, we set the input dimension d_{in} to 4, the embedding dimension d_{hid} to 64, the number of attention heads n_h to 4, the size L of mini-batch to 128, the clipping ratio ϵ to

0.25, the look-back window length of news l_w^n to 6, the look-back window length of features l_w^f to 20 and the discounted factor γ to 0.99.

Baselines. We compare our method with the following baselines:

- Market represents the stock market index.
- BLSW [9] is a classical mean reversion strategy. It posits that stocks exhibiting a prolonged downtrend in the past are more investment-worthy.
- CSM [13] is a classical momentum strategy. It posits that stocks showing a sustained upward trend in the past are more investment-worthy.
- LSTM-A [7] is a method based on the LSTM model that utilizes self-attention.
- TRA [12] is a deep learning approach that classifies and predicts stocks based on the study of their characteristics.
- CGM [28] CGM is a deep learning method that utilizes long-term and short-term stock relationships for predicting stock trend.
- THGNN [20] is a deep learning method that constructs short-term heterogeneous graphs for stock prediction.
- FactorVAE [4] is a factor model based on prior-posterior learning approach for stock prediction.
- CTTS [26] is a deep learning method based on CNN and Transformer for long-term modeling of time series features.
- PPO [14] is a reinforcement learning method that uses constraints to ensure the stability of policy updates, thereby improving training efficiency.
- Alphastock [18] is a deep reinforcement learning method that constructs relations of stocks based on attention mechanisms for portfolio management.
- DeepTrader [19] is a deep reinforcement learning model that models market sentiment for controlling investment risk in portfolio management.
- DeepPocket [16] is a deep reinforcement learning model that utilizes graph convolutional networks for portfolio optimization.

Evaluation Metrics. Our metrics include three fundamental metrics: Annualized Rate of Return (ARR) represents the investment returns, Annualized Volatility (AVol) measures the stability of returns over a year and Maximum DrawDown (MDD) assesses the worst-case scenario that may occur. Additionally, we have three composite metrics: Annualized Sharpe Ratio (ASR) describes the additional risk-adjusted returns based on volatility, Calmar Ratio (CR) characterizes the additional risk-adjusted returns based on the maximum drawdown, and Information Ratio (IR) reflects the excess returns under additional risk.

4.2 Overall Results

The backtesting results of each method are summarized in Table 1. It is evident that AGDRL performs the best ARR, ASR, CR, and IR. The highest ARR demonstrates that AGDRL exhibits superior profit-making capabilities. The three composite metrics affirm that our method excels in balancing returns and risks, aligning well with investor preferences. Although our method's Avol and MDD are not the best, they still outperform the majority of methods. Moreover, excessively minimizing risk when pursuing high returns is unrealistic. A

Table 1. Overall performance of all methods on two datasets. The best performance is highlighted in bold. The larger the values of ARR, ASR, CR, IR, the better the performance. Conversely, smaller values for AVol and MDD indicate better performance.

Datasets	SSE50						NDX100					
	ARR	AVol	MDD	ASR	CR	IR	ARR	AVol	MDD	ASR	CR	IR
Market	−0.15	0.20	−0.43	−0.77	−0.36	−0.74	−0.07	0.27	−0.36	−0.26	−0.19	−0.11
BLSW	0.26	0.28	−0.26	0.93	1.02	1.03	0.09	0.28	−0.24	0.33	0.38	0.47
CSM	0.09	0.40	−0.53	0.23	0.18	0.53	−0.13	0.29	−0.43	−0.44	−0.30	−0.30
LSTM-A	0.16	0.34	−0.42	0.46	0.37	0.60	0.06	0.28	−0.31	0.23	0.21	0.37
TRA	0.38	0.34	−0.34	1.10	1.11	1.11	0.13	0.30	−0.31	0.45	0.44	0.57
CGM	0.38	0.34	−0.32	1.11	1.17	1.11	0.06	0.26	−0.29	0.23	0.22	0.36
THGNN	0.37	0.31	−0.25	1.22	1.52	1.19	0.09	0.26	−0.24	0.33	0.36	0.45
FactorVAE	0.14	0.25	−0.17	0.58	0.82	0.67	0.11	0.30	−0.22	0.38	0.51	0.51
CTTS	0.28	0.27	−0.24	1.05	1.18	1.06	0.14	0.31	−0.31	0.45	0.45	0.58
PPO	0.09	0.34	−0.37	0.27	0.25	0.43	0.16	0.27	−0.25	0.58	0.61	0.67
AlphaStock	0.29	0.28	**−0.17**	1.06	1.74	1.07	0.17	0.25	**−0.15**	0.65	1.08	0.73
DeepPocket	0.14	**0.19**	−0.21	0.72	0.64	0.77	0.02	**0.22**	−0.27	0.08	0.07	0.19
DeepTrader	0.30	0.31	−0.34	0.96	0.89	0.97	0.11	0.32	−0.24	0.35	0.48	0.50
AGDRL	**0.46**	0.28	−0.22	**1.64**	**2.10**	**1.49**	**0.25**	0.24	−0.20	**1.03**	**1.24**	**1.04**

reasonable level of risk is more acceptable. We also recorded the return curves on the test set, as shown in Fig. 2 and Fig. 3. It can be observed that models such as AlphaStock and DeepPocket, which strictly control risk, adopt a conservative investment strategy. While their methods avoids significant declines in returns, the upside is also limited, resulting in a suboptimal balance between returns and risks compared to AGDRL. Our method consistently outperforms the baselines throughout the entire investment horizon, benefiting from the enhanced market perception provided by multi-dimensional asymmetric graph information and the excellent decision-making capabilities of reinforcement learning.

4.3 Ablation Study

We investigated the impact of adjusting the graph structure and training architecture of AGDRL on experiment results. The experiment results are recorded in Table 2. AD-NG represents AGDRL without using graph structures, AD-SG replaces the asymmetric graph structure with a symmetric one, AD-NS does not use the sector graph, AD-NN does not use the news graph, AD-NC represents using maximizing returns instead of the calmar ratio as the reward function, and AD-DL replaces DRL with a DL training structure. From the results, it can be observed that AGDRL produces suboptimal effects when certain structures are missing or altered. This indicates the importance of the asymmetric structure, sector and news graph information, our objective function, and the DRL framework. Particularly, optimizing the network with the objective of maximizing returns instead of the calmar ratio results in lower returns. This suggests that

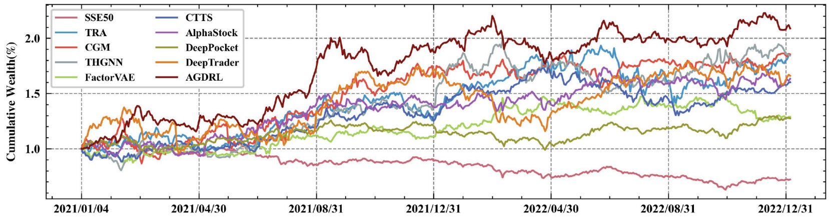


Fig. 2. The test return curves of primary baselines and AGDRL on SSE50.

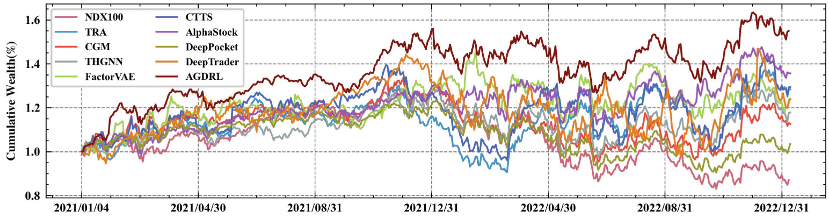


Fig. 3. The test return curves of primary baselines and AGDRL on NDX100.

appropriately considering risks when specifying investment strategies may lead to better investment returns compared to directly pursuing maximum profit.

4.4 Interpretability Analysis

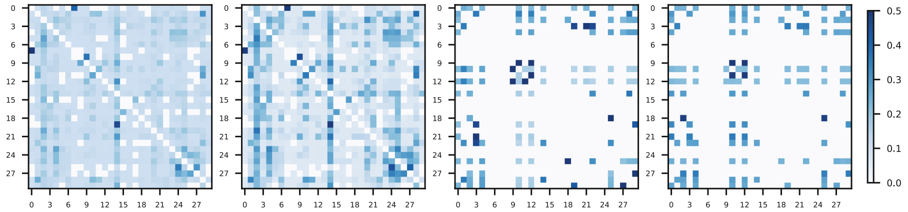
To further investigate the advantages of asymmetric graph structures in relation modeling, we recorded and visualized the attention weights for news and industry graphs, both symmetric and asymmetric, on SSE50, as shown in Fig. 4. For clarity, we did not record self-attention weights for nodes. Figure 4(a) and Fig. 4(b) display the attention weights for symmetric and asymmetric news graphs, respectively. It can be observed that attention weights in symmetric graph structures are evenly distributed with few edges having high weights, whereas in asymmetric graph structures, there is greater diversity in attention weight distribution, indicating the ability of asymmetric relations to allocate different weights based on the importance of relationships. Figure 4(c) and Fig. 4(d) illustrate the attention weights for symmetric and asymmetric industry graphs, respectively. For stable industry structures, attention weights in symmetric graph structures are generally symmetrically distributed, while asymmetric graphs exhibit many asymmetric weights, suggesting the ability of asymmetric relations to distinguish the impact of stocks within industries with higher accuracy.

4.5 Hyperparameter Sensitivity

We recorded the experimental curves using various hyperparameters on SSE50, and the results are illustrated in Fig. 5. As shown in Fig. 5(a), it can be observed

Table 2. Performance evaluation of ablated models on two datasets. The best performance is highlighted in bold

Datasets	Metric	AD-NG	AD-SG	AD-NS	AD-NN	AD-NC	AD-DL	AGDRL
SSE50	ARR	0.18	0.31	0.35	0.38	0.37	0.39	**0.46**
	AVol	0.28	0.29	0.28	0.28	0.28	0.28	**0.28**
	MDD	−0.38	−0.22	−0.24	−0.22	−0.31	−0.24	**−0.22**
	ASR	0.69	1.08	1.24	1.33	1.35	1.38	**1.64**
	CR	0.48	1.38	1.45	1.76	1.20	1.62	**2.10**
	IR	0.77	1.09	1.21	1.28	1.29	1.31	**1.49**
NDX100	ARR	0.11	0.18	0.16	0.20	0.19	0.21	**0.25**
	AVol	0.24	0.25	0.25	0.24	0.25	0.24	**0.24**
	MDD	−0.22	−0.22	−0.23	−0.20	−0.23	−0.20	**−0.20**
	ASR	0.46	0.70	0.63	0.85	0.75	0.88	**1.03**
	CR	0.50	0.80	0.73	1.02	0.85	1.06	**1.24**
	IR	0.56	0.77	0.72	0.90	0.82	0.92	**1.04**



(a) Symmetric News GAT Weights (b) Asymmetric News GAT Weights (c) Symmetric Sector GAT Weights (d) Asymmetric Sector GAT Weights

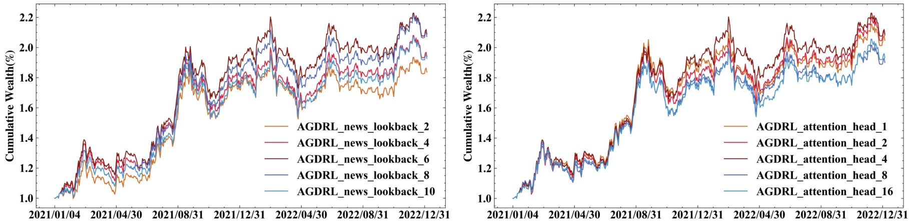
Fig. 4. Visualization of graph attention weights of for news and sector graphs on SSE50.

that both too few and too many days of news look-back window length l_w^n have negative effects. Insufficient days of reviewing news lead to reduced available information due to a smaller quantity of news, while an excessive number of days results in decreased information quality due to the timeliness of news. Through Fig. 5(b), it can be observed that the number of attention heads n_h also has a significant impact on the performance of AGDRL. When the number of heads is too low, the attention mechanism captures insufficient diversity in relationships, and when it is too high, redundant information is generated. Both scenarios can lead to suboptimal experiment results.

5 System Observation

Our work was evaluated in an industrial setting for portfolio optimization across the Chinese A-share market by a leading financial service provider of China.

Our model undergoes offline training every weekend, and the trained model is then used for trading decisions during the week. The trading strategies gen-



(a) News Look-back window length (b) Graph attention head number

Fig. 5. Cumulative return performance of AGDRL in terms of the parameters.

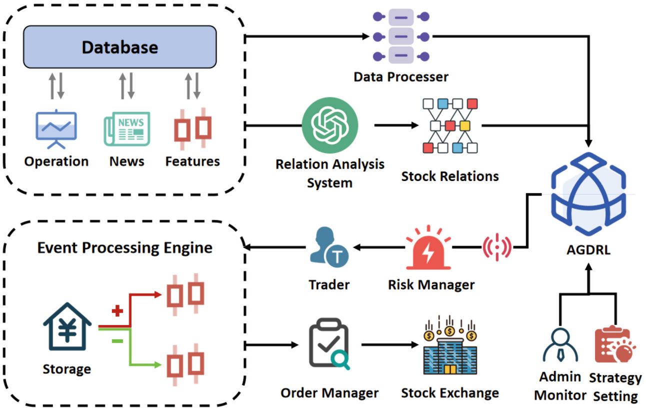


Fig. 6. The industrial system architecture deployed by AGDRL for online trading.

erated by the model are evaluated based on real-time information from the stock exchange. The system architecture is illustrated in the Fig. 6. The server's database system continually records and updates fundamental operational information, financial news for companies. These are integrated before the start of each trading day and processed through a relational parsing system to generate multi-dimensional asymmetric relationships between stocks for model reference. And stock feature data is also fed to the model after being processed by the data processor. Subsequently, trading signals generated by AGDRL are first monitored by the risk management system before being forwarded to the event processing engine. This engine compares existing portfolio information to determine stocks for purchase or sale, which are then fed into the order management system. The management system then calculates investment returns based on information from the stock exchange.

We conducted a six-month observational assessment of the investment performance of AGDRL and we documented the yield curves, as illustrated in Fig. 7. The red curve represents the investment returns of AGDRL, the blue curve represents the market benchmark index, and the orange curve represents the excess returns of AGDRL over the market. The consistently elevated excess returns

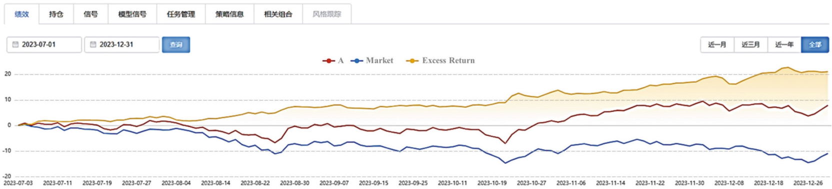


Fig. 7. Observational Evaluation Results of AGDRL across the entire A-share market.

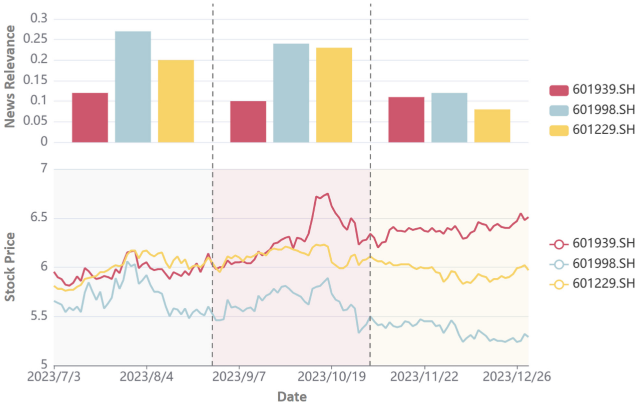


Fig. 8. Case analysis of asymmetric relations. The upper half represents the average news relevance to three stocks during different periods, while the lower half represents the stock price trends of the three stocks.

indicate that AGDRL's investment yield outperforms the market. Even in situations of significant market losses, it manages to incur fewer losses compared to the broader market and eventually achieves positive returns. This demonstrates the applicability of AGDRL to real-world market scenarios.

And we performed case studies to analyze the investment advantages of asymmetric graph structures. We selected three well-known stocks from the banking sector as examples because the banking sector is relatively independent, less influenced by other industries, and exhibits minimal variance in sector based relations, which facilitates our analysis of news information. We recorded the stock trends and mean news relevance of three stocks over three periods, as shown in the Fig. 8. The selected news is related to all three stocks. It can be observed that, in the first two periods, the news relevance of stocks varies greatly. The trend of the least news-relevant stock largely mirrors the most news-relevant one, while another stock is less impacted. In contrast, during the last interval, where news relevance is relatively consistent across the three stocks, their trends are predominantly determined by their individual characteristics. This indicates that asymmetric structures can more accurately model relationships between stocks, reflecting the degree of influence among them.

6 Conclusion

In this paper, we propose a deep reinforcement learning framework based on asymmetric graph information. It leverages the knowledge reasoning capability of large language models to construct asymmetric relation graphs of stocks from complex multi-dimensional information sources, and applies them to end-to-end optimization of investment strategies. This approach, compared to previous symmetric graph methods, enhances the parsing effectiveness of complex financial information and is more suitable for portfolio management tasks in real-world financial scenarios. To our knowledge, this is the first work to apply multi-dimensional asymmetric graph structures to portfolio optimization in real-world markets. Extensive experiments have demonstrated the superiority of our method over the state-of-the-art methods, and the industrial-level observation indicates the applicability of our method to real-world trading systems.

Acknowledgments. This work was supported by the National Key R&D Program of China (Grant no. 2022YFB4501704), the National Natural Science Foundation of China (Grant no. 62102287), and the Shanghai Science and Technology Innovation Action Plan Project (Grant no. 22YS1400600 and 22511100700).

References

1. Cheng, D., Yang, F., Wang, X., Zhang, Y., Zhang, L.: Knowledge graph-based event embedding framework for financial quantitative investments. In: SIGIR 2020, pp. 2221–2230. Association for Computing Machinery, New York (2020)
2. Deng, Y., Bao, F., Kong, Y., Ren, Z., Dai, Q.: Deep direct reinforcement learning for financial signal representation and trading. IEEE Trans. Neural Netw. Learn. Syst. **28**(3), 653–664 (2017)
3. Dimitrios, K., Vasileios, O.: A network analysis of the Greek stock market. Procedia Econ. Finance **33**, 340–349 (2015). The Economies of Balkan and Eastern Europe Countries in the Changed World (EBEEC 2015)
4. Duan, Y., Wang, L., Zhang, Q., Li, J.: FactorVAE: a probabilistic dynamic factor model based on variational autoencoder for predicting cross-sectional stock returns. AAAI **36**, 4468–4476 (2022)
5. Duan, Z., Chen, C., Cheng, D., Liang, Y., Qian, W.: Optimal action space search: an effective deep reinforcement learning method for algorithmic trading. In: Proceedings of the 31st ACM International Conference on Information & Knowledge Management, pp. 406–415 (2022)
6. Fatouros, G., Soldatos, J., Kouroumali, K., Makridis, G., Kyriazis, D.: Transforming sentiment analysis in the financial domain with chatGPT. Mach. Learn. Appl. **14**, 100508 (2023)
7. Feng, F., Chen, H., He, X., Ding, J., Sun, M., Chua, T.S.: Enhancing stock movement prediction with adversarial training. In: IJCAI (2019)
8. Han, L., Ding, N., Wang, G., Cheng, D., Liang, Y.: Efficient continuous space policy optimization for high-frequency trading. In: KDD, pp. 4112–4122 (2023)
9. Jegadeesh, N., Titman, S.: Returns to buying winners and selling losers: implications for stock market efficiency. J. Finance **48**, 65–91 (1993)

10. Li, J., Zhang, Y., Yang, X., Chen, L.: Online portfolio management via deep reinforcement learning with high-frequency data. Inf. Process. Manage. **60**(3), 103247 (2023)
11. Li, W., Bao, R., Harimoto, K., Chen, D., Xu, J., Su, Q.: Modeling the stock relation with graph network for overnight stock movement prediction. In: IJCAI 2020 (2020)
12. Lin, H., Zhou, D., Liu, W., Bian, J.: Learning multiple stock trading patterns with temporal routing adaptor and optimal transport. In: KDD (2021)
13. Poterba, J.M., Summers, L.H.: Mean reversion in stock prices: evidence and implications. J. Financ. Econ. **22**(1), 27–59 (1988)
14. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal policy optimization algorithms (2017). ArXiv, abs/1707.06347
15. Shi, S., Li, J., Li, G., Pan, P., Chen, Q., Sun, Q.: GPM: a graph convolutional network based reinforcement learning framework for portfolio management. Neurocomputing **498**, 14–27 (2022)
16. Soleymani, F., Paquet, E.: Deep graph convolutional reinforcement learning for financial portfolio management-deepPocket. Expert Syst. Appl. **182**, 115127 (2021)
17. Wang, H., Li, S., Wang, T., Zheng, J.: Hierarchical adaptive temporal-relational modeling for stock trend prediction. In: IJCAI (2021)
18. Wang, J., Zhang, Y., Tang, K., Wu, J., Xiong, Z.: AlphaStock: a buying-winners-and-selling-losers investment strategy using interpretable deep reinforcement attention networks. In: KDD. Association for Computing Machinery (2019)
19. Wang, Z., Huang, B., Tu, S., Zhang, K., Xu, L.: DeepTrader: a deep reinforcement learning approach for risk-return balanced portfolio management with market conditions embedding. In: AAAI, pp. 643–650 (2021)
20. Xiang, S., Cheng, D., Shang, C., Zhang, Y., Liang, Y.: Temporal and heterogeneous graph neural network for financial time series prediction. In: CIKM, pp. 3584–3593 (2022)
21. Xu, K., Zhang, Y., Ye, D., Zhao, P., Tan, M.: Relation-aware transformer for portfolio policy learning. In: Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence. IJCAI (2020)
22. Xu, W., et al.: HIST: a graph-based framework for stock trend forecasting via mining concept-oriented shared information (2021). ArXiv, abs/2110.13716
23. Yang, H.W., Zou, Y., Shi, P., Lu, W., Lin, J., Sun, X.: Aligning cross-lingual entities with multi-aspect information, pp. 4431–4441. Association for Computational Linguistics, Hong Kong (2019)
24. Yang, M., Zheng, X., Liang, Q., Han, B., Zhu, M.: A smart trader for portfolio management based on normalizing flows. In: IJCAI (2022)
25. Ye, Y., et al.: Reinforcement-learning based portfolio management with augmented asset movement prediction states. In: AAAI, vol. 34, pp. 1112–1119 (2020)
26. Zeng, Z., Kaur, R., Siddagangappa, S., Rahimi, S., Balch, T., Veloso, M.: Financial time series forecasting using CNN and transformer. In: AAAI (2023)
27. Zhang, Y., Zhao, P., Wu, Q., Li, B., Huang, J., Tan, M.: Cost-sensitive portfolio selection via deep reinforcement learning. IEEE Trans. Knowl. Data Eng. **34**(1), 236–248 (2022)
28. Zhao, L., Li, W., Bao, R., Harimoto, K., Wu, Y., Sun, X.: Long-term, short-term and sudden event: Trading volume movement prediction with graph-based multi-view modeling. In: IJCAI (2021)

Code Summarization with Project-Specific Features

Yu Wang, Xin Liu, Xuesong Lu(✉), and Aoying Zhou

East China Normal University, Shanghai, China
`{wangyu,xinliu}@stu.ecnu.edu.cn`, `{xslu,ayzhou}@dase.ecnu.edu.cn`

Abstract. Code summarization aims to automatically generate natural language descriptions for code snippets, which help people maintain and understand code snippets. Existing code summarization methods are mostly based on the encoder-decoder structure, where the encoder learns latent features from a code snippet and the decoder generates the corresponding summary based on the features. Such methods do not leverage project-specific information and tend to generate general summaries. However, in practice developers want the generated summaries to be project-specific, i.e., being consistent with the existing summaries in the same project on aspects such as sentence patterns and domain concepts. In this work, we investigate project-specific code summarization. We propose a two-stage method CSWPS, which can be seamlessly integrated into any existing encoder-decoder summarization model. In the first stage, CSWPS learns project-specific features from existing summaries in each project using multi-task learning. In the second stage, CSWPS samples from the project-specific features conditioned on the input source code and project information, and extracts the features most relevant to the input code. The features guide the decoder to generate a project-specific summary for the input code. By incorporating CSWPS into existing code summarization models, we can always improve their performance and achieve the new state-of-the-art. We also empirically show that the summaries generated by incorporating CSWPS are more project-specific, via feature visualization and human study. A replication package for this work is available at https://github.com/DaSESmartEdu/CSWPS.

Keywords: Source Code Summarization · Project-Specific · Latent Representation Sampling · Neural Network · Deep Learning

1 Introduction

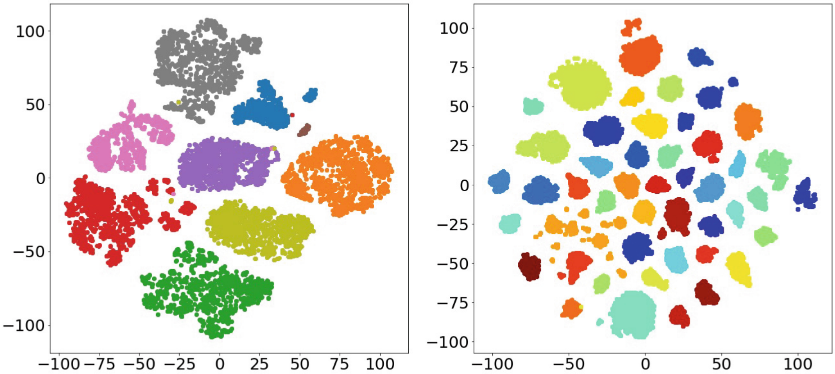
Recent years have witnessed growing research interest in automatic source code summarization [1,9,11,19] due to its beneficial potential in software development

---

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-70378-2_12.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
A. Bifet et al. (Eds.): ECML PKDD 2024, LNAI 14949, pp. 190–206, 2024.
https://doi.org/10.1007/978-3-031-70378-2_12

and maintenance tasks. Generating natural and informative code summaries is not trivial because the intricate syntax structures and the flexible identifier naming in programming languages make it difficult to understand program semantics. In the past few years, various deep learning models have been developed to leverage structural and textual features in the code for generating meaningful and succinct summaries, most of which employ an encoder-decoder structure. The structural features are typically learned using the abstract syntax trees (ASTs) of source code [3,10] and the textual features are learned from the code text sequences [1,8]. Some studies also leverage a hybrid of structural and textual features for better encoding [9,13,18,20].



(a) The PCS dataset (b) The CodeXGLUE dataset

Fig. 1. The t-SNE visualization of the latent representations of the summaries in the PCS and CodeXGLUE dataset.

The models developed in the aforementioned studies are referred to as *general code summarization* (GCS) models [21], as they are trained on code-summary pairs from many software projects. From the practical point of view, recent studies have pointed out that developers may prefer to have a tool that generates code summaries with a consistent style and domain-specific concepts for each specific project [2,21]. The corresponding task is referred to as *project-specific code summarization* (PCS) [21]. For example, some projects prefer to use "return true if" in the summaries rather than use "check whether". Moreover, each project has specific identifier naming, API/function calls and coding patterns, resulting in domain-specific concepts in the summaries. We refer to such properties as ***project-specific features***. To empirically verify the existence of project-specific features in the summaries, we train a feature encoder (see Sect. 3.1) to learn the latent representations of the summaries in two publicly-available datasets of code-summary pairs, namely, PCS [21] and CodeXGLUE [2], respectively. The PCS dataset consists of 9 projects and the CodeXGLUE dataset consists of 47 projects. We use t-SNE to visualize the latent representations. Figure 1 shows the results, where the summaries in the same project are colored the same. The results show clearly that for both datasets, the representations corresponding to the summaries in the same project are much closer to each other in the latent space, indicating that each project has certain project-specific features in its code summaries.

Motivated by the observation, we propose to learn project-specific features from existing summaries and use the features to guide the generation of new summaries. In this way, the newly generated summaries are more likely to be project-specific, i.e., having a style consistent with the existing summaries as well as the domain-specific concepts of the corresponding project. The generated summaries are also expected to have higher quality because of the awareness of project-specific information. To solve our task, we develop a two-stage approach CSWPS for Code Summarization With Project-Specific features. In the first stage, we devise a multi-task learning model to learn the latent project-specific features of the summaries from different projects. In the second stage, to generate the summary for a code snippet, we sample the latent features of existing summaries in the corresponding project that are most relevant to the code snippet and use the features to guide the generation of the summary. Note that CSWPS is a general approach rather than a stand-alone model, which can be seamlessly integrated into any existing code summarization model with an encoder-decoder structure.

Our main contributions are listed as follows:

- We propose a two-stage approach CSWPS for project-specific code summarization. Unlike previous studies [2,21] that focus on the low-resource scenario where the number of summaries in a project is small, we try to produce project-specific summaries by learning features from a bunch of existing summaries.
- The proposed CSWPS can be seamlessly integrated into any code summarization model using an encoder-decoder structure. We conduct extensive experiments to show that by incorporating CSWPS, we promote the performance of existing models and achieve the new state-of-the-art.
- We empirically show that the summaries generated by CSWPS are more project-specific. The visualization shows the latent representations of summaries in the same project are more clustered after incorporating CSWPS. The human study shows the summaries generated using CSWPS are more consistent with existing summaries.

2 Related Work

Current code summarization tasks can be roughly divided into two categories: general code summarization and project-specific code summarization. General code summarization models aim to generate general summaries for many different projects, whereas project-specific models aim to generate consistent summaries for each project.

For general code summarization, some studies consider code as plain text for feature extraction. Lyer et al. [12] use LSTM with attention for code encoding and summarization. Wei et al. [19] explore the duality between code summarization and code generation. Ahmad et al. [1] use Transformer with relative positional encodings. Some studies transform source code into ASTs and graphs for better encoding. Hu et al. [10] convert ASTs into specially formatted sequences and feed them into an LSTM-based encoder. Wan et al. [17] devise two LSTM-based encoders to encode both code and ASTs and use reinforcement learning for summarization. LeClair et al. [13] use convolutional graph neural networks to encode the AST nodes and edges. Tang et al. [15] employ tree-structured attention to encode ASTs. Wang et al. [18] use graph attention networks to encode semantic graphs augmented from ASTs. Other approaches explore different ways to extract semantic features from source code [3,5,7,8].

Recent studies focus on generating project-specific summaries. Bansal et al. [4] use the context of existing source code as project-specific information. Xie et al. [21] use meta-transfer learning to generate summaries for projects with few historical code-summary pairs. Ahmed et al. [2] investigate few-shot training with LLMs like Codex [6]. Our task is different from the above project-specific code summarization tasks. First, we try to learn project-specific features from existing summaries, whereas in [4] the authors learn from existing source code. Second, we aim at the applications where a sufficient number of code-summary pairs are available for the projects, whereas in [2,21] the authors aim at the low-resource scenario.

3 The CSWPS Method

CSWPS is a two-stage approach that can be seamlessly integrated into any existing code summarization model using an encoder-decoder structure. In the first stage, CSWPS adopts multi-task learning to learn the latent representations of the summaries in each project, which captures the project-specific features. In the second stage, CSWPS samples for a code snippet a context feature from the latent summary representations and use the context feature to guide the generation of the corresponding summary. Note that CSWPS learns and samples from the latent representations of the summaries in the training set. Figure 2 shows the overview of CSWPS, which is explained below.

3.1 Stage 1: Learning Latent Summary Representations

To extract the project-specific features entailed in the summaries, in the first stage we train a Transformer encoder to learn the latent representations of the summaries, with the goal of clustering together the representations in the same project and separating the representations in different projects.

As shown in the left part of Fig. 2, we use s_i to denote the ith summary in the training set, which has l_s tokens. We use a Transformer encoder, referred to as SummaryEncoder, to obtain the encoded summary $\boldsymbol{H}_i \in \mathbb{R}^{d \times l_s}$, where d is the encoding size:

$$\boldsymbol{H}_i = \textsc{SummaryEncoder}(s_i). \tag{1}$$

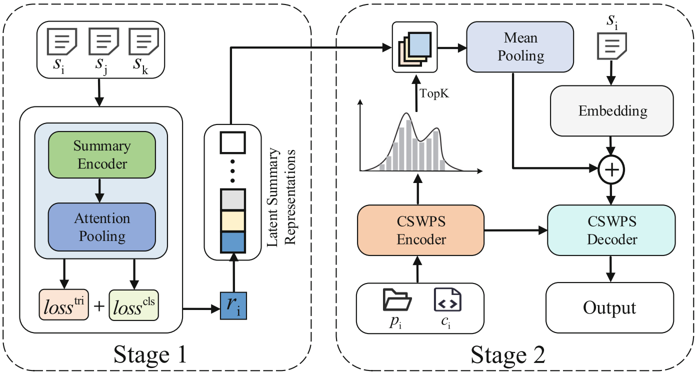


Fig. 2. The overview of CSWPS.

To obtain the latent representation of s_i, we impose a global attention pooling layer on $\boldsymbol{H}_i$, resulting in the representation $\mathbf{r}_i \in \mathbb{R}^{d\times 1}$:

$$a_{ij} = \textsc{Softmax}(\boldsymbol{w}_{attn} \times \boldsymbol{h}_{ij}), \tag{2}$$

$$\tilde{\boldsymbol{r}}_i = \sum_j a_{ij}\boldsymbol{h}_{ij}, \tag{3}$$

$$\boldsymbol{r}_i = \boldsymbol{W}_r \times \tilde{\boldsymbol{r}}_i + \boldsymbol{b}, \tag{4}$$

where $\boldsymbol{h}_{ij} \in \mathbb{R}^{d\times 1}$ denotes the encoding of the jth token in s_i, $\boldsymbol{w}_{attn} \in \mathbb{R}^{1\times d}$, $\boldsymbol{W}_r \in \mathbb{R}^{d\times d}$ and $\boldsymbol{b} \in \mathbb{R}^{d\times 1}$ are the learnable parameters. Then, we design two objectives to learn the latent representation of $\boldsymbol{r}_i$.

Triplet Margin Loss. To obtain discriminative latent summary representations for each project (i.e., latent project-specific features), we employ the Triplet Margin Loss function, which tries to minimize the distance between representations in the same project and maximize the distance between representations in the different projects. Specifically, for each $\boldsymbol{r}_i$, we pair it with two randomly selected representations $\boldsymbol{r}_j$ and $\boldsymbol{r}_k$, respectively, where $\boldsymbol{r}_i$ and $\boldsymbol{r}_j$ are in the same project and $\boldsymbol{r}_k$ is selected from a different project. We regard $\boldsymbol{r}_i$ and $\boldsymbol{r}_j$ as a positive pair and regard $\boldsymbol{r}_i$ and $\boldsymbol{r}_k$ as a negative pair. To calculate the distance between a representation pair, we employ cosine similarity as the metric. Therefore the Triplet Margin Loss function can be formulated as follows:

$$dist_i^{pos} = 1 - \textsc{CosineSim}(\boldsymbol{r}_i, \boldsymbol{r}_j), \tag{5}$$

$$dist_i^{neg} = 1 - \textsc{CosineSim}(\boldsymbol{r}_i, \boldsymbol{r}_k), \tag{6}$$

$$loss_i^{tri} = \textsc{Max}(0, dist_i^{pos} - dist_i^{neg} + m), \tag{7}$$

where m is a hyperparameter denoting the margin. Note that one may use other triplet or contrastive loss functions. We choose the Triplet Margin Loss function because of its high computational efficiency.

Classification Loss. Besides the above unsupervised learning task, we also devise a supervised learning task which classifies the summaries based on the project information. For each latent summary representation $\boldsymbol{r}_i$, we use the corresponding project id p_i as the class label and obtain the classification loss using the cross-entropy function:

$$loss_i^{cls} = -\log \frac{\textsc{Exp}(\boldsymbol{w}_c^i \times \boldsymbol{r}_i)}{\sum_{j=1}^{P} \textsc{Exp}(\boldsymbol{w}_c^j \times \boldsymbol{r}_i)} p_i, \tag{8}$$

where $\boldsymbol{w}_c^i$ are the learnable parameters for label p_i and P is the number of different projects.

Finally, we combine the two types of losses and obtain the total loss for learning the latent representation of the summary s_i:

$$loss_i = \alpha \times loss_i^{tri} + \beta \times loss_i^{cls}, \tag{9}$$

where α and β are the balancing weights of the two losses. After training, SummaryEncoder is used for encoding the summaries in the second stage. Note that if SummaryEncoder is trained using enough projects thereby extracting project-specific features well, it does not need to be retrained every time a new project is added.

3.2 Stage 2: Sampling Latent Summary Representations for Code Summarization

In the second stage, CSWPS can employ any existing encoder-decoder summarization model as backbone and use the project-specific features learned in the first stage to guide the summary generation. Regardless of the specific structure used, we refer to the encoder as CSWPSEncoder and the decoder as CSWPSDecoder. The key module connecting them is a module that samples conditionally the latent summary representations based on the encoded source code and project id.

CSWPSEncoder. As shown in the right part of Fig. 2, we first need to encode the input source code for conditional sampling. In order to obtain more discriminative encodings, we input a code snippet c_i with its project id p_i into CSWPSEncoder and compute the fused encoding. According to encoding code sequences or ASTs in the particular model, we devise two different encoding procedures. If the particular model encodes code sequences (e.g., Transformer), we directly prepend p_i to c_i and input them into CSWPSEncoder:

$$\boldsymbol{E}_i = \textsc{CSWPSEncoder}([p_i; c_i]), \tag{10}$$

where $\boldsymbol{E}_i \in \mathbb{R}^{d\times(l_c+1)}$ denotes the encodings, d is the encoding size and l_c is the number of sub-tokens in c_i.

Some models adopt structure encoders. For example, Chen et al. [7] use a graph encoder and Tang et al. [15] introduce the relationships between AST nodes into the attention computation. For such particular models, we are not able to add p_i into the tree or graph structure. Hence, we first encode c_i and p_i separately and then concatenate the encodings. Specifically, the encoding of p_i is obtained by applying attentions on the encodings of c_i, thereby making p_i and c_i better interact with each other, which is naturally to be done if the model encodes code sequences. Denote by $\boldsymbol{E}_i^c \in \mathbb{R}^{d \times l_c}$ the encodings of c_i and $\tilde{\boldsymbol{e}}_i^p \in \mathbb{R}^{d \times 1}$ the initial embedding of p_i, the encoding $\boldsymbol{e}_i^p$ of p_i is obtained as follows:

$$a_{ij} = \frac{\textsc{Exp}(\tilde{\boldsymbol{e}}_i^p \cdot \boldsymbol{e}_{ij}^c)}{\sum_{k=1}^{l_c} \textsc{Exp}(\tilde{\boldsymbol{e}}_i^p \cdot \boldsymbol{e}_{ik}^c)}, \tag{11}$$

$$\hat{\boldsymbol{e}}_i^p = \sum_j a_{ij} \boldsymbol{e}_{ij}^c, \tag{12}$$

$$\boldsymbol{e}_i^p = \boldsymbol{W}_p \times [\hat{\boldsymbol{e}}_i^p \oplus \tilde{\boldsymbol{e}}_i^p] + \boldsymbol{b}_p, \tag{13}$$

where $\boldsymbol{e}_{ij}^c$ is the jth encoding of $\boldsymbol{E}_i^c$, $\boldsymbol{W}_p \in \mathbb{R}^{d \times 2d}$ and $\boldsymbol{b}_p \in \mathbb{R}^{d \times 1}$ are the learnable parameters. Finally, we concatenate $\boldsymbol{e}_i^p$ and $\boldsymbol{E}_i^c$ to obtain $\boldsymbol{E}_i$

Sampling Latent Summary Representations. Our idea is to leverage the project-specific features of existing summaries to guide the generation of project-specific summaries for the source code in future. Since we obtain the latent representations of existing summaries in the first stage, we can use them to guide the summary generation. However, not every existing summary in a project is relevant to the input code c_i that a summary is generated for. As such, we need to select the most relevant representations to avoid the noisy information. To this end, we use the encoding $\boldsymbol{e}_i^p$ output by CSWPSEncoder as the context vector and sample the most relevant latent representations conditioned on $\boldsymbol{e}_i^p$. The encoding $\boldsymbol{e}_i^p$ can be used as the context vector since it encodes the information of both p_i and c_i, thereby containing the global feature of the project and the source code (See Eq. 10 and 11–13). The sampling process is computed as follows:

$$\hat{\boldsymbol{u}}^i = \boldsymbol{W}_u \times \boldsymbol{e}_i^p + \boldsymbol{b}_u, \tag{14}$$

$$\boldsymbol{u}^i = \textsc{Softmax}(\hat{\boldsymbol{u}}^i), \tag{15}$$

where $\boldsymbol{u}^i \in \mathbb{R}^{n_{p_i} \times 1}$ denotes the probability distribution over existing summaries in project p_i, and n_{p_i} is the number of these summaries. $\boldsymbol{W}_u \in \mathbb{R}^{n_{p_i} \times d}$ and $\boldsymbol{b}_u \in \mathbb{R}^{n_{p_i} \times 1}$ are the learnable parameters.

According to the probability distribution $\boldsymbol{u}^i$, we obtain the top-k relevant latent representations $\boldsymbol{R} = [\boldsymbol{r}_{u_1^i}, \boldsymbol{r}_{u_2^i}, \ldots, \boldsymbol{r}_{u_k^i}]$ with the highest probability. Then we impose a mean pooling layer on $\boldsymbol{R}$ to obtain the context vector $\boldsymbol{z}_i \in \mathbb{R}^{d \times 1}$ for generating the summary.

$$\boldsymbol{z}_i = \frac{1}{k} \sum_{\boldsymbol{r}_{u_j^i} \in \boldsymbol{R}} \boldsymbol{r}_{u_j^i}. \tag{16}$$

CSWPSDecoder. With the context vector $\boldsymbol{z}_i$, we can decode the source code encodings $\boldsymbol{E}_i^c$ to generate the summary more specific to project p_i. Specifically, we linearly transform $\boldsymbol{z}_i$ to $\hat{\boldsymbol{z}}_i$ and add $\hat{\boldsymbol{z}}_i$ into the embedding of each word in the summary s_i. The obtained embeddings are denoted as $\boldsymbol{X}^i \in \mathbb{R}^{d \times l_s}$, where l_s is the summary length. The process is computed as:

$$\hat{\boldsymbol{z}}_i = \boldsymbol{W}_z \times \boldsymbol{z}_i + \boldsymbol{b}_z, \tag{17}$$

$$\hat{\boldsymbol{Z}}_i = \underbrace{[\hat{\boldsymbol{z}}_i; \hat{\boldsymbol{z}}_i; \ldots; \hat{\boldsymbol{z}}_i]}_{l_s}, \tag{18}$$

$$\boldsymbol{X}^i = \textsc{WordEmbed}(s_i) + \textsc{PosEmbed}(s_i) + \hat{\boldsymbol{Z}}_i, \tag{19}$$

where $\boldsymbol{W}_z \in \mathbb{R}^{d \times d}$ and $\boldsymbol{b}_z \in \mathbb{R}^{d \times 1}$ are learnable parameters. WordEmbed and PosEmbed denote word embeddings and positional embeddings, respectively.

The decoder CSWPSDecoder takes $\boldsymbol{E}_i^c$ and the summary embeddings $\boldsymbol{X}_{<t}^i$ before each time step t as input, and predict the summary token $\hat{\boldsymbol{y}}_t^i$ at time step t:

$$\hat{\boldsymbol{y}}_t^i = \textsc{CSWPSDecoder}(\boldsymbol{X}_{<t}^i, \boldsymbol{E}_i^c). \tag{20}$$

The Loss Function. We compute the negative log-likelihood of the generated summary as the loss in the second stage. Denote by $\boldsymbol{\theta}$ the learnable parameters in the second stage, the loss function for the summary s_i is calculated as:

$$loss_i^{gen} = -\sum_{t=1}^{l_s} P(\hat{\boldsymbol{y}}_t^i = \boldsymbol{y}_t^i | p_i, c_i, \boldsymbol{r}_i, s_i^{<t}, \boldsymbol{\theta}), \tag{21}$$

where l_s is the summary length and $\boldsymbol{y}_t^i$ is the ground-truth summary token at time t.

At inference time, for each code snippet c_i in the testing set, we compute its context vector $\boldsymbol{u}^i$ and use it to sample the latent representations of summaries in the training set, which guides the summary generation for c_i.

4 Experiment Settings

4.1 Datasets

In order to learn project-specific features, we need the datasets with project information for the code-summary pairs. To this end, we refer to the works of Xie et al. [21] and Ahmed et al. [2] and use their datasets. Xie et al. construct the PCS dataset that contains 9 famous open-source projects such as Spring-Boot, ExoPlayer and Flink, which can be directly used for our task. On the other hand, we follow [2] and select 47 projects from the CodeXGLUE [14] code summarization benchmark, based on the criteria that each project has more than 500 code-summary pairs. We refer to this dataset as CodeXGLUE. The source code of both datasets is written in Java. The numbers of code-summary pairs in PCS and CodeXGLUE are 32,667 and 51,890, respectively. The detailed summary statistics of PCS and CodeXGLUE can be found in the supplementary materials.

To reduce the vocabulary size, we follow the approach in [1] and split the original source code into sub-tokens. For each dataset, we randomly divide the code-summary pairs in each project with proportion 6:2:2 and form the training, validation and testing part, respectively. After combining all the projects, each dataset has also such a proportion for the training, validation and testing set.

4.2 Hyperparameters in Our Model

In both stages, the embedding size, the dimension of the key and value, and the dimension of the feedforward layer are set to 512, 64 and 2,048, respectively, in the Transformer structure. In the first stage, the loss weights α and β are set to 1 and the margin m for the Triplet Margin Loss is set to 0.5. In the second stage, the number of sampled latent project-specific features k is 20. The complete settings can be found in the supplementary materials.

4.3 Evaluation Metrics

We adopt three commonly used metrics to evaluate the performance, namely, BLEU, METEOR and ROUGE-L. The score for each metric is calculated as the average of all generated summaries in the testing set. Higher scores indicate better performance for all metrics. To prevent random results, we train the model for 4 runs with random seeds and report the average results.

4.4 The Comparative Models

We select ten models developed in recent years for general code summarization. The models are DeepCom [10], Dual Model [19], Hybrid-DeepCom [11], Rencos [22], Transformer [1], GN-Transformer [7], SiT [20], AST-Trans [15], GypSum [18] and SCRIPT [9]. Remember that our main model uses SCRIPT as the backbone for the second stage.

In addition, we conduct experiments using MPCos [21] and few-shot training with LLMs [2], respectively. Although the two studies aim at the low-resource scenario, we report their results since they both generate project-specific summaries. MPCos trains a meta-transfer learning module to learn the initial weights for prefix-tuning, which uses separate prefixes for each project to promote project-specific transfer learning. For few-shot training, Ahmed et al. [2] design a prompting method and send the prompt to the Codex model [6].

5 Experimental Results

This section evaluates the effectiveness of CSWPS. Specifically, we aim to answer the following four research questions:

- **RQ1:** How does our main model perform compared to state-of-the-art models?

- **RQ2:** Can CSWPS improve the performance of existing code summarization models?
- **RQ3:** How do different modules pertaining to project-specific features affect the performance of code summarization?
- **RQ4:** Do the summaries generated using CSWPS preserve better the project-specific features?

5.1 RQ1: Our Main Model Vs. SOTA Models

We compare our main model that employs SCRIPT [9] as backbone (SCRIPT+CSWPS) with the 12 models described in Sect. 4.4. For all the models except MPCos and few-shot training with LLMs, we utilize the original implementations released by the authors. The authors of MPCos provide the implementation only for the meta-transfer learning module. We implement the code summarization module and the prefix-tuning module according to the descriptions in [21]. For the budget reason, we run two-shot training with CodeLLaMA-7B [16] instead of Codex [6] as used in [2]. We choose CodeLLaMA-7B because it is reported to perform better than Codex on many code-related tasks.

Table 1. Our main model compared to the state-of-the-art models. The improvements are statistically significant, verified by a 2-tailed Student's t-test; p-value < 0.05.

Models	PCS			CodeXGLUE		
	BLEU	METEOR	ROUGE-L	BLEU	METEOR	ROUGE-L
DeepCom [10]	17.21	10.01	24.46	18.10	10.25	28.88
Dual [19]	33.61	23.55	47.04	27.51	18.05	42.78
Hybrid-DeepCom [11]	26.47	19.32	40.05	24.82	17.13	39.47
Rencos [22]	38.44	27.28	50.71	28.24	19.29	43.67
Transformer [1]	38.03	26.59	50.89	29.22	19.29	43.83
GN-Transformer [7]	37.33	26.27	50.23	29.27	18.99	44.47
SiT [20]	39.12	27.58	52.26	31.71	20.41	46.61
AST-Trans [15]	34.79	25.02	46.79	28.41	19.56	36.90
GypSum [18]	38.22	27.46	52.04	30.86	21.14	46.73
SCRIPT [9]	39.48	27.71	52.64	31.67	20.50	46.83
MPCos [21]	14.01	10.43	23.81	14.06	10.10	22.11
CodeLLaMA-7B [2]	19.24	16.43	35.66	18.28	14.50	31.00
SCRIPT+CSWPS	**40.95** (↑1.47)	**28.62** (↑ 0.91)	**54.22** (↑ 1.58)	**32.70** (↑ 1.03)	**21.81** (↑ 1.31)	**48.48** (↑ 1.65)

The results are reported in Table 1. We observe that our main model SCRIPT+CSWPS performs the best on all metrics for both datasets. Specifically, it improves the scores of SCRIPT (the previous SOTA) by more than 1 on all metrics except METEOR for PCS, which is also close to 1. The results of MPCos and CodeLLaMA-7B are similar to that in [21] and [2], indicating that we correctly re-implement their methods. However, MPCos and CodeLLaMA-7B do not perform as well as other models, which is not surprising because of their low-resource settings. The results indicate that when there exist a sufficient number of historical summaries in a project, using CSWPS to learn the project-specific features from the summaries to guide the generation of future summaries can yield much better performance than meta-transfer learning or few-shot training with LLMs.

5.2 RQ2: Can CSWPS Improve Existing Models?

We implement CSWPS in 7 representative encode-decoder models for code summarization, i.e., integrating the feature sampling module into their decoder, respectively. To demonstrate the effect on different types of models, we choose Transformer [1], which solely leverages code sequences (Text), DeepCom [10] and AST-Trans [15], which solely leverage ASTs (AST), and Hybrid-DeepCom [11], GN-Transformer [7], GypSum [18] and SCRIPT [9], which leverage both code sequences and ASTs (Hybrid).

Table 2. The performance of the models after incorporating CSWPS. The improvements are statistically significant, verified by a 2-tailed Student's t-test; p-value < 0.05.

Models incorporating CSWPS		PCS			CodeXGLUE		
		BLEU	METEOR	ROUGE-L	BLEU	METEOR	ROUGE-L
Text	Transformer	38.64 (↑ 0.61)	27.18 (↑ 0.59)	51.46 (↑ 0.57)	30.46 (↑ 1.24)	20.14 (↑ 0.85)	45.56 (↑ 1.73)
AST	DeepCom	18.21 (↑ 1.00)	10.96 (↑ 0.95)	26.65 (↑ 2.19)	18.97 (↑ 0.87)	10.92 (↑ 0.67)	30.40 (↑ 1.52)
	AST-Trans	35.21 (↑ 0.42)	25.23 (↑ 0.21)	47.39 (↑ 0.60)	28.62 (↑ 0.21)	19.94 (↑ 0.38)	37.15 (↑ 0.25)
Hybrid	Hybrid-DeepCom	27.02 (↑ 0.55)	19.97 (↑ 0.65)	40.54 (↑ 0.49)	25.88 (↑ 1.06)	17.93 (↑ 0.80)	41.40 (↑ 1.93)
	GN-Transformer	38.28 (↑ 0.95)	26.80 (↑ 0.53)	51.29 (↑ 1.06)	29.89 (↑ 0.62)	19.55 (↑ 0.56)	45.49 (↑ 1.02)
	GypSum	39.11 (↑ 0.89)	28.09 (↑ 0.63)	53.10 (↑ 1.06)	31.57 (↑ 0.71)	21.53 (↑ 0.39)	47.54 (↑ 0.81)
	SCRIPT	40.95 (↑ 1.47)	28.62 (↑ 0.91)	54.22 (↑ 1.58)	32.70 (↑ 1.03)	21.81 (↑ 1.31)	48.48 (↑ 1.65)

The results are reported in Table 2. The scores in the parentheses show the improvements compared to the original model. We observe that by incorporating CSWPS, all models are improved on all metrics for both datasets, indicating CSWPS can improve existing models using an encoder-decoder structure. Although designed for GCS, incorporating the project-specific features makes these models generate summaries more consistent with existing ones in each project, thereby yielding improved summarization performance.

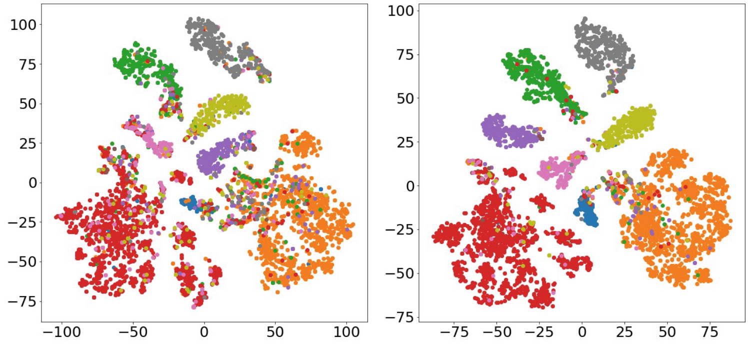
5.3 RQ3: The Effect of the Modules Pertaining to Project-Specific Features

We remove or replace three important modules related to learning or incorporating project-specific features from our main model SCRIPT+CSWPS, respectively, and show the summarization performance of the remaining model. First, we remove the Triplet Margin Loss for extracting project-specific features from existing summaries in the first stage, and only keep the Classification Loss. The resulted model is referred to as SCRIPT+CSWPS (w/o Triplet). Second, we remove the project id p_i from the input of the encoder in the second stage, so that the project-specific features are sampled only by the source code encoding. The resulted model is referred to as SCRIPT+CSWPS (w/o Project Id). Finally, we replace the conditional sampling of project-specific features by uniform sampling at random in the second stage. The resulted model is referred to as SCRIPT+CSWPS (w/o Cond. Sampl.). The results are reported in Table 3.

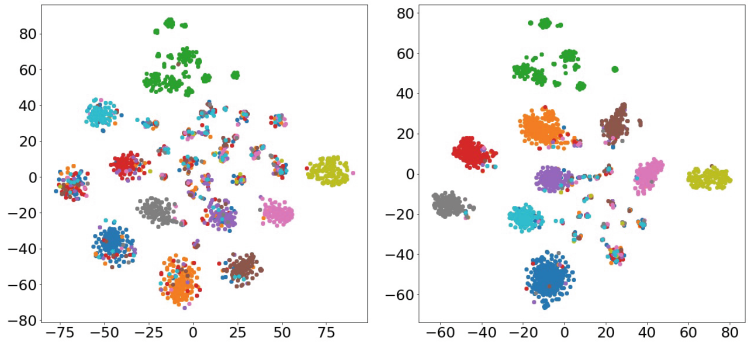
Table 3. The performance after removing the modules pertaining to project-specific features.

Models	PCS			CodeXGLUE		
	BLEU	METEOR	ROUGE-L	BLEU	METEOR	ROUGE-L
SCRIPT+CSWPS (our model)	**40.95**	**28.62**	**54.22**	**32.70**	**21.81**	**48.48**
SCRIPT+CSWPS (w/o Triplet)	40.42	28.48	53.67	32.35	21.36	48.13
SCRIPT+CSWPS (w/o Project Id)	40.53	28.51	53.89	32.44	21.50	48.21
SCRIPT+CSWPS (w/o Cond. Samp.)	40.17	28.11	53.42	32.03	21.25	48.01
SCRIPT [9]	39.48	27.71	52.64	31.67	20.50	46.83

We observe that the performance drops if any of the three modules is removed or replaced, indicating the effectiveness of these modules in learning or incorporating project-specific features for code summarization. Specifically, by replacing the conditional sampling with uniform sampling at random, the summarization performance drops the most. This shows the importance of using the features most relevant to the code to guide the summary generation. On the other hand, all three models perform better than the original SCRIPT model, indicating the effectiveness of incorporating project-specific features, even if they are not leveraged optimally.



(a) PCS (SCRIPT) (b) PCS (SCRIPT+CSWPS)



(c) CodeXGLUE (SCRIPT+CSWPS) (d) CodeXGLUE (SCRIPT+CSWPS)

Fig. 3. The t-SNE visualization of the latent representations of the generated summaries, before and after incorporating CSWPS.

5.4 RQ4: Are the Generated Summaries More Project-Specific?

We conduct two experiments. First, we use the summary encoder in the first stage to encode the summaries generated by SCRIPT and SCRIPT+CSWPS on the testing sets of both PCS and CodeXGLUE. Then, we use t-SNE to visualize the distribution of the latent representations of the generated summaries. We keep only the 10 largest projects in CodeXGLUE for better visualization. The results are shown in Fig. 3, where the points with the same color correspond to the summaries generated in the same project. We observe that for both datasets, the outlier points that appear among or near the points of a different project are much fewer after incorporating CSWPS, indicating that the model using CSWPS can generate summaries preserving better the project-specific features.

In addition, we use K-meas to cluster the latent representations and employ the Silhouette and S-Dbw scores to measure the clustering performance before and after incorporating CSWPS. A higher Silhouette (a lower S-Dbw) score indicates better performance. The results are shown in Table 4. We observe that the both scores show the latent representations of the generated summaries are bet-

Table 4. The performance of clustering the latent representations of the generated summaries, before and after incorporating CSWPS.

Dataset	Model	Silhouette	S_Dbw
PCS	SCRIPT	0.45	0.31
	SCRIPT+CSWPS	**0.50**	**0.23**
CodeXGLUE	SCRIPT	0.49	0.29
	SCRIPT+CSWPS	**0.59**	**0.21**

ter clustered after the model incorporates CSWPS, i.e., the generate summaries preserve better the project-specific features.

Second, we conduct a human study to investigate whether the summaries generated using CSWPS are more consistent with the existing summaries in the same project. We randomly sample 50 summaries in each of the 5 largest projects from both PCS's and CodeXGLUE's training sets, thereby obtaining 500 existing summaries in total. Then we randomly select 10 code snippets in each of the 5 largest projects from both the testing sets, and use SCRIPT and SCRIPT+CSWPS to generate the summaries, respectively, thereby obtaining 100 generated summaries in total. Then, we invite three master students in computer science to evaluate how each generated summary is consistent with the 50 existing summaries in the same project. We particularly ask them to evaluate on two aspects: *sentence similarity* and *content relevance*. Sentence similarity measures how a generated summary conforms with the overall sentence pattern of the 50 existing summaries in the same project (e.g., check whether vs. return true if). Content relevance measures how the content of a generated summary is relevant to the overall contents of the 50 existing summaries in the same project (i.e., they describe related functions.). The evaluators are asked to score each generated summary using the metrics on a scale of 0–3, where a higher score indicates higher similarity or relevance. Then we calculate the average scores of the two metrics.

Table 5. Consistency measures of model outputs. Statistical significance between the two models is computed with a 2-tailed Student's t-test; p-value < 0.05.

Model	Sentence Similarity	Content Relevance
SCRIPT	2.15	1.87
SCRIPT+CSWPS	**2.41**	**2.19**

The results are reported in Table 5. We observe that SCRIPT+CSWPS improves the scores of SCRIPT on both metrics, indicating that the human evaluators agree that the summaries generated using CSWPS are more consistent with the existing summaries in the same project, i.e., the generated summaries are more project-specific. In addition, we observe that the scores of content relevance are lower than those of sentence similarity. This is not surprising because the code functions are usually more diverse than sentence patterns and it is less likely to find an existing summary with the similar function to the generated summary than with the similar sentence pattern.

Overall, the above experiments show that the summaries generated using CSWPS preserve better the project-specific features.

6 Discussion and Conclusion

In this work, we learn project-specific features of existing summaries and sample the features to guide the generation of new summaries. The task is particularly useful for large projects, where the summarization model can learn project-specific features from a bunch of existing summaries and then be used to generate consistent summaries for source code developed in future. The model can be deployed in a CI/CD pipeline to quickly generate project-specific summaries for newly-committed code, thereby facilitating the development and maintenance process for a large project.

One potential limitation is that the experimental datasets contain only the projects written in Java. This is because we compare with many different code summarization models to show the effectiveness of CSWPS, where only Java code adapts to all these models in their official implementations. However, we believe that the CSWPS works also for other common languages, since they have similar grammar rules and very close AST structures.

Our experimental results have shown that by incorporating CSWPS into existing models, we can not only achieve better summarization performance, but also generate more preject-specific summaries. In future, we will try to devise other methods for learning project-specific features and further promote the performance.

Acknowledgments. This work is supported by the grant from the National Natural Science Foundation of China (Grant No. 62277017)

References

1. Ahmad, W., Chakraborty, S., Ray, B., Chang, K.W.: A transformer-based approach for source code summarization. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 4998–5007 (2020)
2. Ahmed, T., Devanbu, P.: Few-shot training LLMs for project-specific code-summarization. In: Proceedings of the 37th IEEE/ACM International Conference on Automated Software Engineering, pp. 1–5 (2022)

3. Alon, U., Brody, S., Levy, O., Yahav, E.: code2seq: generating sequences from structured representations of code. In: 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, 6–9 May 2019 (2019)
4. Bansal, A., Haque, S., McMillan, C.: Project-level encoding for neural source code summarization of subroutines. In: 2021 IEEE/ACM 29th International Conference on Program Comprehension (ICPC), pp. 253–264. IEEE (2021)
5. Chen, F., Kim, M., Choo, J.: Novel natural language summarization of program code via leveraging multiple input representations. In: Findings of the Association for Computational Linguistics: EMNLP 2021, pp. 2510–2520 (2021)
6. Chen, M., et al.: Evaluating large language models trained on code. arXiv preprint arXiv:2107.03374 (2021)
7. Cheng, J., Fostiropoulos, I., Boehm, B.: GN-Transformer: fusing sequence and graph representation for improved code summarization. arXiv preprint arXiv:2111.08874 (2021)
8. Feng, Z., et al.: CodeBERT: a pre-trained model for programming and natural languages. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings, pp. 1536–1547 (2020)
9. Gong, Z., Gao, C., Wang, Y., Gu, W., Peng, Y., Xu, Z.: Source code summarization with structural relative position guided transformer. In: 2022 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER), pp. 13–24. IEEE (2022)
10. Hu, X., Li, G., Xia, X., Lo, D., Jin, Z.: Deep code comment generation. In: Proceedings of the 26th Conference on Program Comprehension, pp. 200–210 (2018)
11. Hu, X., Li, G., Xia, X., Lo, D., Jin, Z.: Deep code comment generation with hybrid lexical and syntactical information. Empir. Softw. Eng. **25**, 2179–2217 (2020)
12. Iyer, S., Konstas, I., Cheung, A., Zettlemoyer, L.: Summarizing source code using a neural attention model. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (2016)
13. LeClair, A., Haque, S., Wu, L., McMillan, C.: Improved code summarization via a graph neural network. In: Proceedings of the 28th International Conference on Program Comprehension, pp. 184–195 (2020)
14. Lu, S., Guo, D., Ren, S., et al.: CodeXGLUE: a machine learning benchmark dataset for code understanding and generation. In: Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual (2021)
15. Tang, Z., et al.: AST-trans: code summarization with efficient tree-structured attention. In: Proceedings of the 44th International Conference on Software Engineering, pp. 150–162 (2022)
16. Touvron, H., et al.: Llama 2: Open foundation and fine-tuned chat models. arXiv preprint arXiv:2307.09288 (2023)
17. Wan, Y., et al.: Improving automatic source code summarization via deep reinforcement learning. In: Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering, pp. 397–407 (2018)
18. Wang, Y., Dong, Y., Lu, X., Zhou, A.: GypSum: learning hybrid representations for code summarization. In: Proceedings of the 30th IEEE/ACM International Conference on Program Comprehension, ICPC 2022, Virtual Event, 16–17 May 2022, pp. 12–23. ACM (2022)
19. Wei, B., Li, G., Xia, X., Fu, Z., Jin, Z.: Code generation as a dual task of code summarization. In: Advances in Neural Information Processing Systems 2019. Neural Information Processing Systems (NIPS) (2019)

20. Wu, H., Zhao, H., Zhang, M.: Code summarization with structure-induced transformer. In: Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pp. 1078–1090 (2021)
21. Xie, R., Hu, T., Ye, W., Zhang, S.: Low-resources project-specific code summarization. In: Proceedings of the 37th IEEE/ACM International Conference on Automated Software Engineering, pp. 1–12 (2022)
22. Zhang, J., Wang, X., Zhang, H., Sun, H., Liu, X.: Retrieval-based neural source code summarization. In: 2020 IEEE/ACM 42nd International Conference on Software Engineering (ICSE), pp. 1385–1397. IEEE (2020)

Unraveling Anomalies in Time: Unsupervised Discovery and Isolation of Anomalous Behavior in Bio-Regenerative Life Support System Telemetry

Ferdinand Rewicki[1,2](✉), Jakob Gawlikowski[1], Julia Niebling[1], and Joachim Denzler[2]

[1] Institute of Data Science, German Aerospace Center (DLR), 07745 Jena, Germany
`{ferdinand.rewicki,jakob.gawlikowski,julia.niebling}@dlr.de`
[2] Institute of Computer Science, Friedrich-Schiller University, 07743 Jena, Germany
`joachim.denzler@uni-jena.de`

Abstract. The detection of abnormal or critical system states is essential in condition monitoring. While much attention is given to promptly identifying anomalies, a retrospective analysis of these anomalies can significantly enhance our comprehension of the underlying causes of observed undesired behavior. This aspect becomes particularly critical when the monitored system is deployed in a vital environment. In this study, we delve into anomalies within the domain of Bio-Regenerative Life Support Systems (BLSS) for space exploration. We analyze anomalies found in telemetry data stemming from the EDEN ISS space greenhouse in Antarctica, using MDI and DAMP, two glassbox methods for anomaly detection based on density estimation and discord discovery respectively. We employ time series clustering on anomaly detection results to categorize various types of anomalies in both uni- and multivariate settings. We then assess the effectiveness of these methods in identifying systematic anomalous behavior. Additionally, we illustrate that the anomaly detection methods MDI and DAMP produce complementary results, as previously indicated by research.

Keywords: Unsupervised Anomaly Detection · Time Series · Multivariate · Controlled Environment Agriculture · Clustering

1 Introduction

Bio-regenerative Life Support Systems (BLSSs) are artificial ecosystems that consist of multiple symbiotic relationships. BLSSs are crucial for sustaining long-duration space missions by facilitating food production and managing essential

---

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-70378-2_13.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
A. Bifet et al. (Eds.): ECML PKDD 2024, LNAI 14949, pp. 207–222, 2024.
https://doi.org/10.1007/978-3-031-70378-2_13

material cycles for respiratory air, water, biomass, and waste. The EDEN NEXT GEN Project, part of the EDEN roadmap at the German Aerospace Center (DLR), aims to develop a fully integrated ground demonstrator of a BLSS comprising all subsystems, with the ultimate goal of realizing a flight-ready BLSS within the next decade. This initiative builds upon insights from the EDEN ISS project, which investigated controlled environment agriculture (CEA) technologies for space exploration. EDEN ISS, a near-closed-loop research greenhouse deployed in Antarctica from 2017 to 2021, focused on crop production, including lettuces, bell peppers, leafy greens, and various herbs. To ensure the safe and stable operation of BLSSs, we explore methods to mitigate risks regarding system health, particularly regarding food production and nourishment shortages for isolated crews. Given the absence of clear definitions for unhealthy system states and the unavailability of ground truth data, we investigate unsupervised anomaly detection methods. Unsupervised anomaly detection targets deviations or irregularities from expected or standard behavior in the absence of labelled training data. Choosing the appropriate method from the plethora of available options is challenging due to differing strengths in detecting certain types of anomalies, as no universal method exists [10].

To address this challenge, we conducted a comparative analysis of six unsupervised anomaly detection methods, differing in complexity in [16]. Three of these methods are classical machine learning techniques, while the remaining three are based on deep learning. The primary questions in this comparison have been: (1) "Is it worthwhile to sacrifice the interpretability of classical methods for potentially superior performance of deep learning methods?" and (2) "What different types of anomalies are the methods capable of detecting?" The findings underscored the efficacy of two classical methods, Maximally Divergent Intervals (MDI) [4] and MERLIN [15], which not only performed best individually but also complement each other in terms of the detected anomaly types.

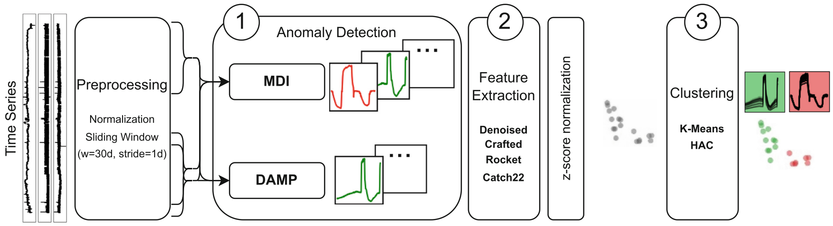


Fig. 1. Overview of our approach to derive different types of anomalous behavior from unlabelled time series. ① MDI and DAMP are applied to time series data to obtain anomalous sequences. ② Features are extracted from sequences and ③ clustering is applied to derive different anomaly types, marked by color.

Building upon our findings in [16], we analyze telemetry data[1] from the EDEN ISS subsystems for the mission year 2020. Our objectives include discovering anomalous behavior, differentiating different types of anomalies, and identifying recurring anomalous behavior. Figure 1 outlines our approach. We apply MDI and Discord Aware Matrix Profile (DAMP), another algorithm for Discord Discovery similar to MERLIN, for anomaly detection, to obtain univariate and multivariate anomalies and extract four sets of features from the anomalous sequences. We finally apply K-Means and Hierachical Agglomerative Clustering (HAC) to obtain clusters representing similar anomalous behavior. Experimental validation addresses five research questions related to the complementarity of MDI and DAMP (RQ1), optimal feature selection (RQ2), superior clustering algorithms (RQ3), types of isolated anomalies (RQ4), and identification of recurring abnormal behavior (RQ5).

2 Related Work

While the field of anomaly detection has witnessed great interest and an enormous number of publications in recent years, particularly in endeavors focusing on timely anomaly detection. Scant attention has been directed towards the critical task of categorizing and delineating various forms of anomalous behavior. Sohn et al. (2023) discriminate anomalous images into clusters of coherent anomaly types using bag-of-patch-embeddings representations and HAC with Ward linkage [21]. In [16] we evaluated six anomaly detection methods with varying complexity regarding their ability to detect certain shape-based anomaly types in univariate time series data. We showed, that the density estimation-based method Maximally Divergent Intervals (MDI) [4] and the Discord Discovery method MERLIN [15] not only deliver the best individual results in this study but yield complementary results in terms of the types of anomalies they can detect. Tafazoli et al. (2023) recently proposed a combination of the Matrix profile, which is the underlying technique behind discord discovery, with the "Canonical Time-series Chracteristics" (Catch22) features, that we employ as one of our methods for feature extraction [22]. Ruiz et al. (2021) experimentally compared algorithms for time series classification, and found that "the real winner of this experimental analysis is ROCKET [Author's note: Random Convolutional Kernel Transform]" [19] as it is the best ranked and by far the fasted classifier in their study [19]. We use ROCKET as one method to derive features from time series. Anomaly detection has also garnered interest in the CEA domain. While the works [1,2,9] proposed various methods for anomaly detection in the CEA and Smart Farming domain, other studies have explored anomaly detection's utility in enhancing greenhouse control [6] or monitoring plant growth [7,23]. However, there has been limited effort to extract potential systematic behavior from anomaly detection results. This work aims to address this gap

---

[1] The code to reproduce our results is available at https://gitlab.com/dlr-dw/unraveling-anomalies-in-time/-/tree/v1.0.0.

by contributing towards the derivation of systematic behaviors from anomaly detection outcomes in the telemetry data of the EDEN ISS space greenhouse.

3 Methodology

In the following section we introduce our pipeline outlined in Fig. 1 to derive different types of anomalous behavior from unlabeled time series. We start by defining time series data and subsequences, followed by the introduction of the methods for anomaly detection and feature extraction. Finally, we present the measures we use to evaluate the quality of clustering results. While we evaluate our approach purely based on our specific use case, its generic nature allows it to be applied to different time series mining tasks.

Time series are sequential data that are naturally ordered by time. We define a regular time series as an ordered set of observations made at equidistant intervals based on [15]:

Definition 1. *The regular* ***time series*** $\mathcal{T}$ *with length* $N \in \mathbb{N}$ *is defined as the set of pairs* $\mathcal{T} = \{(t_n, \mathbf{x_n}) | t_n \leq t_{n+1}, 0 \leq n \leq N-1, t_{n+1} - t_n = c\}$ *with* $\mathbf{x_n} \in \mathbb{R}^D$ *being the data points having* D *behavioral attributes and* $t_n \in \mathbb{N}, n \leq N$ *the equidistant timestamps the data refer to. For* $D = 1$*,* $\mathcal{T}$ *is called univariate, and for* $D > 1$*,* $\mathcal{T}$ *is called multivariate.*

As time series are usually not analyzed en bloc, we define a subsequence as a contiguous subset of the time series:

Definition 2. *The* ***subsequence*** $\mathcal{S}_{a,b} \subseteq \mathcal{T}$ *of the times series* $\mathcal{T}$*, with length* $L = b - a + 1 > 0$ *is given by* $\mathcal{S}_{a,b} := \{(t_n, \mathbf{x_n}) | 0 \leq a \leq n \leq b < N\}$*. For multivariate time series* $\mathcal{T}$*,* $\mathcal{S}^{(i)}_{a,b}$ *with* $i \in \mathbb{N}$ *refers to the subsequence* $\mathcal{S}_{a,b}$ *in dimension* $1 \leq i \leq D$ *For brevity, we often omit the indices and refer to arbitrary subsequences as* $\mathcal{S}$*.*

3.1 Anomaly Detection

In the following, we understand anomalies as collective anomalies, i.e. special subsequences S, that deviate notably from an underlying concept of normality. We focus on collective anomalies as we are interested in prolonged environmental issues that can significantly impact plant health and operational efficiency. We selected MDI and DAMP as they yielded not only the best individual but also complementary results in [16].

MDI [4] is a density-based method for offline anomaly detection in multivariate, spatiotemporal data. We focus on temporal data in this study, providing definitions pertinent to this context. For comprehensive definitions, including spatial attributes, refer to [4]. MDI identifies anomalous subsequences in a multivariate time series $\mathcal{T}$ by comparing the probability density $p_{\mathcal{S}}$ of a subsequence $\mathcal{S} \subseteq \mathcal{T}$ to the density p_{Ω} of the remaining part $\Omega(\mathcal{S}) := \mathcal{T} \setminus \mathcal{S}$. These distributions are modeled using Kernel Density Estimation or Multivariate Gaussians. MDI quantifies the degree of deviation $\mathcal{D}(p_{\mathcal{S}}, p_{\Omega})$ an unbiased variant of the Kullback-Leibler divergence. The most anomalous subsequence $\tilde{\mathcal{S}}$ is identified by solving the optimization problem: $\tilde{\mathcal{S}} := \arg\max_{\mathcal{S}\subseteq\mathcal{T}} \mathcal{D}(p_{\mathcal{S}}, p_{\Omega(\mathcal{S})})$. MDI locates this most anomalous subsequence $\tilde{\mathcal{S}}$ by scanning all subsequences $\mathcal{S} \subseteq \mathcal{T}$ with lengths between predefined parameters $L_{min}, L_{max} \in \mathbb{N}$ and estimating the divergence $\mathcal{D}(p_{\mathcal{S}}, p_{\Omega(\mathcal{S})})$, which serves as the anomaly score. The anomalous subsequences are selected by ranking all subsequences based on their anomaly score and applying Non-maximum suppression. To adapt to large-scale data, MDI employs an interval proposal technique based on Hotelling's T^2 method [13]. This technique selects interesting subsequences based on point-wise anomaly scores rather than conducting full scans over the entire time series, motivated by the rarity of anomalies in time series by definition [4]. We set L_{min} and L_{max} to 144 (0.5 days) and 288 (1 day) empirically.

DAMP [11] is a method for both offline and "effectively online" anomaly detection by discord discovery. The term "effectively online" was introduced by [11] to classify algorithms that are not strictly online but where "the lag in reporting a condition has little or no impact on the actionability of the reported information" [11]. Given a subsequence $\mathcal{S}_{a_1,b_1}$ and a matching subsequence $\mathcal{S}_{a_2,b_2}$ with $b_1 - a_1 = b_2 - a_2 = L$, $\mathcal{S}_{a_1,b_1}$ is a *non-self match* to S_{a_2,b_2} with distance d_{a_1,a_2} if $|a_1 - a_2| \geq L$. $dist(\cdot,\cdot)$ denotes the z-normalized Euclidean distance. The discord $\tilde{\mathcal{S}}$ of a time series $\mathcal{T}$ is defined as the subsequence with the maximum distance $d(\tilde{\mathcal{S}}, M_{\tilde{\mathcal{S}}})$ from its nearest non-self match $\mathcal{M}_{\tilde{\mathcal{S}}}$. To ascertain the discord of a time series, DAMP approximates the left matrix profile $P^{L(\mathcal{T})}$, a vector storing the z-normalized Euclidean distance between each subsequence of $\mathcal{T}$ and its nearest non-self match occurring before that subsequence. DAMP comprises a forward and a backward pass. In the backward pass, each subsequence is assessed to determine if it constitutes the discord of the time series. Meanwhile, the forward pass aids in pruning data points that do not qualify as discord based on the best-so-far discord distance. We set L to 288 (1 day) empirically.

3.2 Feature Extraction

The objective of our analysis is to identify particular anomaly types specific to the EDEN ISS telemetry dataset. Given the exploratory nature of this analysis, we examine four distinct feature extraction methods, which we refer to below as "feature sets" and elaborate on in this section.

Denoised Subsequences. As a first feature set, we utilize the raw subsequences identified as anomalous by MDI or DAMP. These subsequences vary in length from 18 to 447 data points[2] in the univariate and from 13 to 289 in the multivariate case. To compare sequences of differing lengths, we employ Dynamic Time Warping (DTW) [5]. To enhance comparability, we apply moving average smoothing with a window size of five data points to eliminate high frequencies. The window size has been set empirically. In subsequent discussions, we refer to this feature set as `Denoised`.

---

[2] We excluded anomalies with a length of fewer than five data points from our analysis.

Handcrafted Feature-Vectors. For the second feature set, we derive a nine-dimensional vector comprising simple statistical and shape-specific features. This vector encompasses the first four moments, i.e. mean, variance, kurtosis, and skewness, alongside the sequence length, the minimum and maximum values, and the positions of the minimum and maximum within the sequence. Following the computation of these feature vectors, we employ z-score normalization to standardize the features to a zero mean and unit standard deviation. In the following, we will refer to this feature set as `Crafted`.

Random Convolutional Kernel Transform (ROCKET). [8] generates features from time series using a large number of random convolutional kernels. Each kernel is applied to every subsequence, yielding two aggregate features: maximum value (similar to global max pooling) and proportion of positive values (PPV) [8,19]. Pooling, akin to convolutional neural networks, reduces dimensionality and achieves temporal or spatial invariance, while PPV captures kernel correlation. ROCKET employs 10,000 kernels with lengths $l \in \{7, 9, 11\}$ and weights $\mathbf{w} \in \mathbb{R}^l$ sampled from the standard normal distribution $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. We apply Principal Component Analysis (PCA) to the z-normalized transformation outcome and use the first 10 components, also z-normalized, as final features to mitigate dimensionality issues. We reduced the number of kernels to 1000, finding no significant alteration in results. We refer to this feature set as `Rocket`.

Canonical Time Series Characteristics (catch22). [12] comprise 22 time series features derived from an extensive search through 4,791 candidates and 147,000 diverse datasets. These features, tailored for time series data mining, demonstrate strong classification performance and minimal redundancy [12]. They encompass various aspects such as distribution of values, temporal statistics, autocorrelation (linear and non-linear), successive differences, and fluctuation. We apply these features to each subsequence, resulting in a 22-dimensional feature vector per sequence. Features with a normalized variance exceeding a threshold (set empirically at 0.01) are selected and the chosen features are then z-normalized. We will refer to this feature set as `Catch22`.

3.3 Time Series Clustering

Time series clustering involves partitioning a dataset $\mathfrak{D}$ containing time series $\mathcal{T}^{(1)}, \ldots, \mathcal{T}^{(|\mathfrak{D}|)}$ into K disjoint subsets $C_k, k = 2, \ldots, K$, where each subset contains similar time series. Similarity is measured using distance measures like Euclidean Distance or Dynamic Time Warping (DTW) [5]. In this study, we compare K-Means clustering [14] with Hierarchical Agglomerative Clustering (HAC) to identify clusters of similar anomalous subsequences.

K-Means clustering [14] partitions a set of n observations into K clusters by assigning each observation to the cluster with the nearest mean, minimizing within-cluster variance. The centroids $\mu_k, k = 1, \ldots, K$ serve as cluster prototypes. However, vanilla K-Means may not yield optimal results as it randomly selects initial centroids, making it sensitive to seeding. To address this, [3] proposed K-Means++, which selects centroids with probabilities proportional to their contribution to the overall potential. K-Means++ is now a standard initialization strategy for K-Means clustering, including in our experiments.

HAC partitions a set of n observations into a hierarchical structure of clusters. It begins by treating each data point as a separate cluster and then merges the closest clusters iteratively. The choice of a linkage criterion, determining the dissimilarity measure between clusters, is crucial. In our experiments, we adopt the *Unweighted Pair-Group Method of Centroids (UPGMC)* linkage, which calculates the distance between clusters based on the distance between their centroids. Other common linkage criteria include *Single*, *Complete* and *Ward* linkage, which respectively use the minimum (Single) or maximum (Complete) distance between points from different clusters as linkage criterion or minimize the within cluster variance (Ward).

3.4 Quality Measures

The **Silhouette Score (SSC)** [18] is the standard measure for evaluating clustering results and quantifies both cohesion and separation within clusters. It is calculated by averaging over the Silhouette Coefficients SSC_C for each cluster C, defined as:

$$SSC_C = \frac{1}{|C|} \sum_{\mathcal{S} \in C} \frac{idist(\mathcal{S}) - wdist(\mathcal{S})}{\max(wdist(\mathcal{S}), idist(\mathcal{S}))} \; . \tag{1}$$

Here, $wdist(\mathcal{S})$ represents the mean distance of object $\mathcal{S} \in C$ to all other elements within its own cluster C (within-cluster distance), while $idist(\mathcal{S})$ denotes the smallest mean distance to elements in another cluster (inter-cluster distance). [18] SSC ranges from -1 to 1, where 1 indicates well-separated clusters, 0 suggests overlapping clusters, and -1 implies misclassification of objects.

To evaluate the quality of clustering outcomes, we introduce the **Synchronized Anomaly Agreement Index (SAAI)**. Let

$$A = \{\mathcal{S}^{(i)}_{a,b} | i, a, b \in \mathbb{N}, i \leq D, a < b, s(\mathcal{S}^{(i)}_{a,b}) > th\} \tag{2}$$

be the set of univariate anomalies in the time series $\mathcal{T} = \{\mathcal{T}^{(1)}, ..., \mathcal{T}^{(D)}\}$ where $s(\cdot) : \{\mathcal{S} | \mathcal{S} \subseteq \mathcal{T}\} \rightarrow [0,1]$ denotes a anomaly score function, and $th \in [0,1]$ represents the threshold for labeling a subsequence anomalous. Furthermore, let

$$A_S = \{(\mathcal{S}^{(i)}_{a_i,b_i}, \mathcal{S}^{(j)}_{a_j,b_j}) | \mathcal{S}^{(i)}_{a_i,b_i}, \mathcal{S}^{(j)}_{a_j,b_j} \in A, i < j, iou([a_i,b_i],[a_j,b_j]) > th_{iou}\} \tag{3}$$

be the set of synchronized, i.e. temporally aligned, anomalies with $iou([a_i,b_i], [a_j,b_j])$ representing the time-interval *Intersection over Union* of two subsequences $\mathcal{S}^{(i)}_{a_i,b_i}$ and $\mathcal{S}^{(j)}_{a_j,b_j}$. The threshold parameter $th_{iou} \in [0,1]$ determines the degree of temporal alignment. Additionally, let

$$A^*_S \subseteq A_S \tag{4}$$

denote the set of temporally aligned anomalies assigned to the same cluster, where $c(S^{(i)}_{a_i,b_i}) = c(S^{(j)}_{a_j,b_j})$, with $c(\mathcal{S}^{(i)}_{a_i,b_i})$ indicating the cluster of subsequence $\mathcal{S}^{(i)}_{a_i,b_i}$. The SAAI of a clustering solution of univariate anomalous subsequences in the set of time series $\mathcal{T} = \{\mathcal{T}^{(1)}, ..., \mathcal{T}^{(D)}\}$ is defined as:

$$SAAI := \lambda \frac{|A^*_S|}{|A_S|} - (1-\lambda)(\frac{1}{K} + \frac{n_{\mathbb{1}}}{K}) + (1-\lambda) \text{ , } \lambda \in [0,1] \text{ .} \tag{5}$$

Here, the first term $\frac{|A^*_S|}{|A_S|}$ evaluates the ratio of temporally aligned anomalies in the same cluster among all temporally aligned anomalies. The second term serves as regularization, accounting for small cluster sizes ($\frac{1}{K}$) and clusters containing only a single anomaly ($\frac{n_{\mathbb{1}}}{K}$), where $n_{\mathbb{1}}$ represents the number of single-element clusters. The parameter λ allows adjusting the influence of the penalty term. $(1-\lambda)$ is added to scale the value of SAAI to the interval $[0,1]$, enabling the comparison of SAAI values with different weights λ. In our experiments detailed in Sect. 4, we set $\lambda = 0.5$.

The rationale behind this measure is, although we lack knowledge of the real anomaly clusters, we hypothesize that temporally aligned anomalies in similar measurements - such as those from the same sensor types - should cluster together, as they likely represent the same anomaly. Higher values indicate better clustering solutions. In the supplementary materials A we provide examples and further information to interpret SAAI.

The **Gini-Index** [20] is a metric for statistical dispersion or imbalance. Given a set of discrete values $X = x_1, x_2, \ldots, x_K$, it is defined as:

$$G = \frac{\sum_{i=1}^{K} \sum_{j=1}^{K} |x_i - x_j|}{2n \sum_{i=1}^{K} x_i} \text{ .} \tag{6}$$

The Gini-Index ranges from 0 to 1, with lower values indicating more equal distribution and higher values suggesting greater inequality. We use the Gini-Index to evaluate cluster size imbalance across various clusterings by applying Eq. 6 to the cluster sizes $x_1 = |C_1|, \ldots, x_K = |C_K|$ of each solution.

4 Experimental Results

The experimental results were obtained by first applying MDI and DAMP in the uni- and multivariate case. Data from one subsystem is represented by one multivariate time series. We extract features from the detected anomalous subsequences and cluster them with number of clusters $2 \leq K \leq 20$. All experiments were run on an Intel Xeon Platinum 8260 CPU with 20GB of allocated memory Table S1 in the supplemental materials lists the hyperparameter settings we used for our experiments.

4.1 Dataset

The *edeniss2020* dataset [17] comprises equidistant sensor readings from 97 variables derived from the four subsystems of the EDEN ISS greenhouse, namely

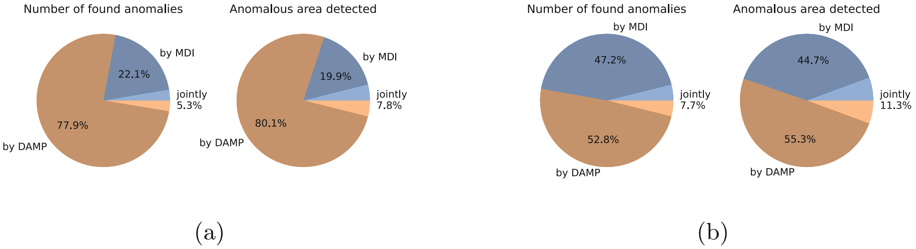


Fig. 2. Amount of (a) univariate and (b) multivariate anomalies found by MDI and DAMP. The highlighted area marks anomalies found by both algorithms.

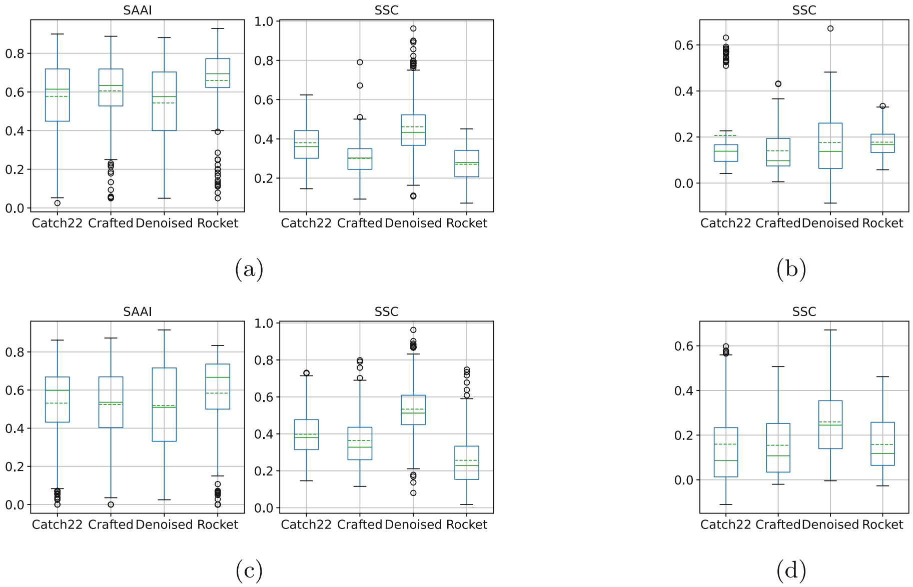
the Atmosphere Management System (AMS), Nutrient Delivery System (NDS), Illumination Control System (ICS) and Thermal Control System (TCS). Our analysis focuses on data from the year 2020, representing EDEN ISS's third operational year. Table S2 in the supplemental materials C lists the measurements per subsystem. The data is captured at a sampling rate of 1/300 Hz and covers the range from 01/01/2020 – 12/30/2020. Every of the 97 univariate time series has a length of 105120 data points. The readings from the AMS pertain to air properties in the greenhouse and service section, while those from the NDS relate to nutrient solution tanks and pressure measurements in the pipes connecting tanks and growth trays. ICS temperature readings are taken above each growth tray and beneath the corresponding LED lamps.

4.2 RQ1: Are the Results of MDI and DAMP Complementary?

In [16] we found, that density estimation- and discord discovery based methods, specifically MDI and MERLIN, yield complementary results in anomaly detection. To validate this claim, we analyzed the anomalies in EDEN ISS telemetry data, found by MDI and DAMP. Our comparison focused on the number and duration of detected anomalies. The results, depicted in Fig. 2, show that DAMP accounts for the majority (77.9%) of the detections in the univariate case, while 22.1% of the detected subsequences are found by MDI. Only 5.3% of the sequences detected as anomalous are found by MDI and DAMP simultaneously. Considering the length of the detected subsequences (i.e. the anomalous area), only 7.8% of the subsequences detected as being anomalous are detected jointly. In the multivariate scenario, DAMP identifies 52.8% and MDI 47.2% of the subsequences, with 7.7% being detected by both. Even though we can not assume all detections being true positives, these findings confirm the complementary nature of MDI and DAMP, with DAMP finding slightly longer anomalies compared to MDI. Hence, using both methods allows us finding a larger variety of anomalous behavior instead of using just one (Fig. 2).

4.3 RQ2: Which Features Yield the Best Results?

To determine which features yield better results, we clustered anomalies for each sensor type (univariate anomalies) or subsystem (multivariate anomalies) with



(a)

(b)

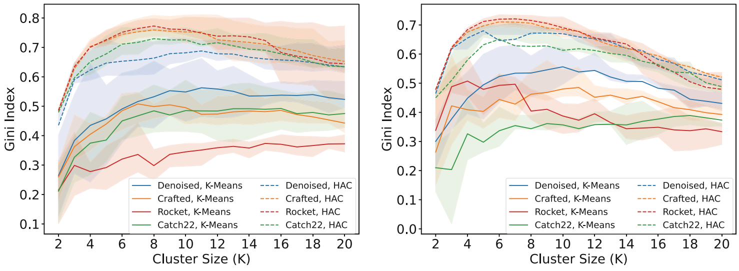
(c)

(d)

Fig. 3. (a), (c) *SAAI* and *SSC* aggregated over all sensor types for the four different feature extraction methods for (a) K-Means and (c) HAC for univariate anomalies. (b), (d): *SSC* over all subsystems in the multivariate case.

varying cluster numbers $K = 2$ to $K = 20$. We then aggregated the sensor-type or subsystem specific outcomes over the number of clusters K. We assessed the quality based on the temporal alignment of clusters using $SAAI$ (Eq. 5) for univariate anomalies, and cluster separation and cohesion using SSC (Eq. 1) for uni- and multivariate anomalies. The results are illustrated as box plots in Fig. 3.

Univariate Anomalies. Analyzing the $SAAI$ distribution for the four feature sets with K-Means clustering (Fig. 3a), `Rocket` features exhibit the highest median $SAAI$ of 0.69, followed by `Crafted` features with 0.63 and `Catch22` with 0.61. `Denoised` subsequences yield a median $SAAI$ of 0.58. With HAC, `Rocket` features show the highest median $SAAI$ of 0.67, followed by `Catch22` with 0.6, `Crafted` with 0.54 and `Denoised` features with a median $SAAI$ of 0.51. `Denoised` features display the highest variability in $SAAI$ results, while `Rocket` features demonstrate the lowest variability for both K-Means clustering and HAC. Regarding cluster separation and cohesion (right plots in Fig. 3), `Denoised` features return the highest median SSC of 0.43 for K-Means and 0.51 for HAC, while `Rocket` features yield the lowest median SSC of 0.28 for K-Means and 0.23 for HAC. Although the discrepancy between $SAAI$ and SSC results might initially seem contradictory, it can be better understood when considering different cluster imbalances, as it will be discussed in Sect. 4.4. For `Denoised`, `Crafted`, and `Catch22` features, higher Silhouette scores are observed for HAC compared to K-Means, indicating higher cluster imbalances. Since SSC



(a) (b)

Fig. 4. Gini-Index of cluster sizes for K-Means Clustering and HAC in (a) the univariate and (b) the multivariate case. Lines represent mean values, while the opaque bands span the second and third quartiles.

is the mean of Silhouette Coefficients and does not consider cluster sizes, higher SSC values may result from many small but dense clusters and few large and dispersed ones, compared to a more even distribution across clusters.

Multivariate Anomalies. Since $SAAI$ is not defined in the multivariate case, we evaluate SSC results, aggregated across all subsystems, which are presented in Fig. 3b and 3d. For K-Means clustering, `Rocket` features exhibit the highest median SSC value of 0.17, whereas for HAC, `Denoised` features demonstrate the highest SSC, with a value of 0.24.

Based on these results we favor `Rocket` features as this feature set shows superior in terms of $SAAI$ when clustering univariate anomalies and the highest median SSC in the multivariate case. For HAC the results are more inconclusive.

4.4 RQ3: Which Algorithm Yields Better Results?

To determine which algorithm yields superior results, we evaluate the imbalance of cluster sizes generated by K-Means and HAC. Although we cannot assume a uniform distribution of anomaly types within the EDEN ISS dataset, we prefer a more even distribution across clusters to support the isolation of varied anomalous behavior. Similar to Sect. 4.3, we cluster anomalies for each sensor type with increasing cluster numbers K (ranging from $K = 2$ to $K = 20$) and aggregate the sensor-type-specific results. Figure 4 depicts the Gini-Index for increasing K by feature set and clustering algorithm. It is evident from the results, that K-Means produces more evenly distributed clusters compared to HAC for all four feature sets.

Univariate Anomalies. K-Means shows moderate imbalance, with average Gini Indices up to 0.51 for `Denoised` features, while HAC yields higher Gini Indices, averaging up to 0.77 for `Rocket`. In K-Means clustering, `Rocket` features generate the most balanced clusters, whereas `Denoised` features exhibit the highest

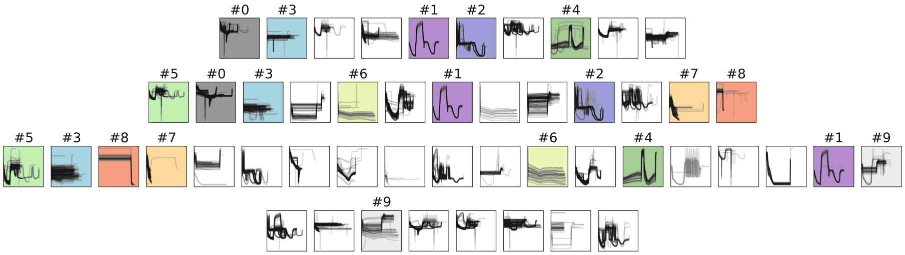


Fig. 5. Anomaly types found in ICS temperatures with K-Means clustering for *SAAI*-optimal K per feature set. Anomaly types consistent across different feature sets have the same color. T/B: `Denoised Crafted`, `Rocket`, `Catch22`

imbalance. Conversely, for HAC, this observation is inverted. The trajectories of the Gini-Index curves are largely similar within each algorithm. For K-Means clustering, Gini Indices increase until $K = 11$, while for HAC, they peak around $K = 8$ before slowly decreasing. This suggests that in HAC, large clusters are not split but gradually dissolve as K increases. `Rocket` with K-Means clustering follows a slightly different trajectory, slowly increasing until $K = 20$.

Multivariate Anomalies. In comparison to HAC, K-Means clustering shows more balanced cluster sizes, consistent with the findings in the univariate case. `Denoised` features have the highest average Gini Index of 0.56 for K-Means, while for HAC, `Rocket` features exhibit the highest mean value of 0.72. The trajectories of the Gini Index curves are largely similar for each algorithm, though for HAC, there is a steeper decline from $K = 8$, especially noticeable when compared to the univariate case.

The observation of K-Means producing more balanced cluster sizes, both in univariate and multivariate scenarios, prompts us to focus on addressing the remaining two research questions w.r.t. K-Means.

4.5 RQ4: What Anomaly Types can be Isolated?

To analyze, which anomaly types can be isolated from the clustering results, we analyze the consensus between the *SAAI*-optimal clustering solutions for the four different feature sets. Given the clustering result $C_A = \{C_{1_A}, C_{2_A}, \ldots, C_{K_A}\}$ for a feature set A and $C_B = \{C_{1_B}, C_{2_B}, \ldots, C_{K_B}\}$ a feature set B, we calculate the matrix of pairwise intersection over union $\mathbf{M}_{AB}(i,j) = iou(C_{i_A}, C_{j_B})$. The values in $\mathbf{M}_{AB}$ are normalized to the interval $[0, 1]$. We consider cluster i in feature set A to isolate the same anomaly type as cluster j in feature set B, if $\mathbf{M}_{AB}(i,j) \geq 0.5$, i.e. if both clusters share at least 50% of their samples.

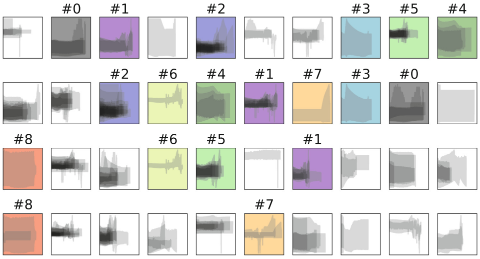


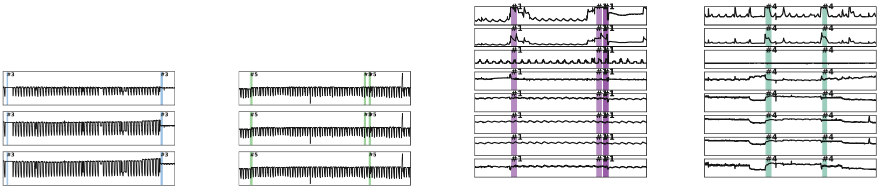
Fig. 6. Anomaly types found in the multivariate AMS-SES measurements with K-Means clustering for $K = 10$. T/B: `Denoised`, `Crafted`, `Rocket`, `Catch22`

Case Study 1: Univariate Anomalies in ICS. Figure 5 depicts the anomaly clusters obtained from K-Means clustering with $SAAI$-optimal number of clusters K[3]. The $SAAI$ results for each feature set and $2 \leq K \leq 20$ are shown in the supplemental Figure S7. The ICS measurements consist of 38 time series, representing temperature readings below the LED lamps for each growth tray, with 1303 identified anomalies for this sensor type. Optimal results for `Catch22` and `Denoised` features were observed at $K = 8$ and $K = 10$, while for `Crafted` and `Rocket` features, the highest $SAAI$ values were obtained at $K = 13$ and $K = 19$. Anomaly type candidates derived from the consensus criterion are highlighted with colored background in Fig. 5. Table S3 in the supplemental materials D provides shape descriptions for 10 isolated anomaly type candidates identified in ICS temperature readings. The "peak (long)" (#1) anomalies are isolated by `Denoised`, `Crafted`, and `Rocket` features. The remaining anomaly type candidates were isolated by two feature sets. `Rocket` and `Crafted` features yield the most anomaly type candidates, reflecting their highest $SAAI$ for the highest number of clusters. For candidate #7, we found no indication of anomalous behavior, so we suspect that this cluster contains false positives results. To investigate whether a larger number of clusters alone aids in analysis, we analyzed the isolated anomaly types at $K = 19$ for all feature sets. Annotated sequence cluster plots are presented in the supplemental Figure S10. `Rocket` features isolate the most, i.e. 8, distinct anomaly type candidates, followed by `Crafted` and `Catch22` with 6, and `Denoised` with 4 distinct anomaly type candidates, indicating superior performance for `Rocket` in forming interpretable anomaly clusters despite the increase in K. `Crafted` features however yielded more distinct clusters for the $SAAI$-optimal value $K = 13$, underlining the efficacy of that measure.

Case Study 2: Multivariate Anomalies in AMS-SES. In the multivariate case, we examine anomalies within each subsystem, here focusing on the AMS-SES subsystem. As $SAAI$ computation for multivariate anomalies is not feasible, we fix $K = 10$ and use the consensus criterion to delineate anomaly type candidates from clustering results. Figure 6 displays the outcomes, with `Crafted`

---

[3] High-Resolution versions of the images can be found in the supplemental materials D.



(a) (b) (c) (d)

Fig. 7. Recurring anomalies in the univariate ICS readings: (a) Near-flat or Flat Signal (#3), (b) Anomalous Day Phase (#5) and in the AMS-SES measurements: (c) CO^2 Peak (#1), (d) Temp. and CO^2 Peak with RH Drop (#4).

features discerning the majority (i.e. 7) of anomaly type candidates, followed by `Denoised` features with 6. `Rocket` features isolate 4 candidates and `Catch22` only 2. The "CO^2 Peak" anomaly (#1) exhibits the highest consensus, detected by `Denoised`, `Crafted`, and `Rocket` features. All other anomaly type candidates show a consensus of 2/4. Descriptions for isolated anomaly type candidates are provided in Table S4 the supplemental materials D. While semantic interpretation remains elusive, we characterize them based on their shape. For instance, anomaly types (#5, #8) in Fig. 6 denote various manifestations of "steep slope" anomalies. Yet, descriptions for candidates (#0, #2, #3) were challenging, hinting at potential false positive anomaly detection results. In summary, interpreting multivariate anomalies is more challenging due to diverse sensor readings in each subsystem's multivariate time series.

4.6 RQ5: Can We Identify Recurring Abnormal Behavior?

To identify recurring abnormal behavior, we focus on anomaly types with multiple instances. In Fig. 7, we illustrate examples of these type candidates for the univariate (7a, 7b) and multivariate case (7c, 7d). While the univariate "near flat or flat signal" anomaly type (#3) shows nearly identical behavior across both instances, the "anomalous day phase" type (#5) exhibits greater diversity. In the initial instance, the warm-up phase involves a step increase followed by a rise in daytime temperature, while in the third instance of the "anomalous day phase" anomaly, a daytime drop occurs after achieving the target temperature.

Using the same methodology in the multivariate case for AMS-SES readings as described in Sect. 4.5, we present two instances of recurring anomalous behavior, namely anomaly types "CO^2 peak" (#1) and "temperature and CO^2 peak with relative humidity drop" (#4), in Fig. 7c and 7d. In both cases, the anomaly is observable and shows similar behavior across the instances, but we lack evidence to assert consistent underlying causes across instances. Labeling these instances as the same anomaly type requires a more in-depth analysis of the anomalies and their root causes, a task we leave for future research.

5 Conclusions and Outlook

In this study, we analyzed anomalies in telemetry data from the BLSS prototype, EDEN ISS, during the mission year 2020. Using anomaly detection methods MDI and DAMP, we extracted four feature sets from identified anomalous subsequences. Employing K-Means clustering and HAC, we aimed to isolate various anomaly types. Our findings showed K-Means produced more uniform cluster sizes compared to HAC, aligning with our goal of discerning diverse anomalous behavior forms. We found `Rocket` and `Crafted` features outperformed `Denoised` subsequences and `Catch22` features in detecting univariate anomalies. However, assessing multivariate anomalies quality solely using SSC proofed challenging. Despite these challenges, our analysis identified various anomaly types, including peaks, anomalous day/night patterns, drops, and delayed events, through consensus among different feature sets. These insights are crucial for refining our risk mitigation system for future BLSS iterations. We identified instances of potentially recurring anomalous behavior in both uni- and multivariate contexts, warranting further investigation. Additionally, we will further explore Catch22 features, promising informative insights into our problem domain.

Acknowledgments. The authors would like to thank Vincent Vrakking and Daniel Schubert for providing the raw data and Tom Norman for compiling the dataset used in this work. We also thank the anonymous reviewers for their valuable suggestions.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Adkisson, M., Kimmell, J.C., Gupta, M., Abdelsalam, M.: Autoencoder-based anomaly detection in smart farming ecosystem. In: 2021 IEEE International Conference on Big Data (Big Data), pp. 3390–3399 (2021)
2. de Araujo Zanella, A.R., et al.: CEIFA: a multi-level anomaly detector for smart farming. Comput. Electron. Agric. **202**, 107279 (2022)
3. Arthur, D., Vassilvitskii, S.: k-means++: the advantages of careful seeding. In: SODA 2007, Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 1027–1035. Society for Industrial and Applied Mathematics, USA (2007)
4. Barz, B., et al.: Detecting regions of maximal divergence for spatio-temporal anomaly detection. IEEE Trans. Pattern Anal. Mach. Intell. **41**, 1088–1101 (2018)
5. Berndt, D.J., Clifford, J.: Using dynamic time warping to find patterns in time series. In: Aaaiws 1994, Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining, pp. 359–370. AAAI Press (1994)
6. Cañadas, J., et al.: Improving automatic climate control with decision support techniques to minimize disease effects in greenhouse tomatoes. Inf. Process. Agric. **4**(1), 50–63 (2017)
7. Choi, K., et al.: Classification of growth conditions in paprika leaf using deep neural network and hyperspectral images. In: 2021 Twelfth International Conference on Ubiquitous and Future Networks (ICUFN), pp. 93–95 (2021)

8. Dempster, A., et al.: Rocket: exceptionally fast and accurate time series classification using random convolutional kernels. Data Min. Knowl. Disc. **34**(5), 1454–1495 (2020)
9. Joaquim, M.M., et al.: IoT and machine learning based anomaly detection in WSN for a smart greenhouse. In: Sharma, S., Peng, S.L., Agrawal, J., Shukla, R.K., Le, D.N. (eds.) Data, Engineering and Applications, pp. 421–431. Springer Nature Singapore, Singapore (2022). https://doi.org/10.1007/978-981-19-4687-5_32
10. Laptev, N., et al.: Generic and scalable framework for automated time-series anomaly detection. In: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Association for Computing Machinery (2015)
11. Lu, Y., et al.: Matrix profile xxiv: scaling time series anomaly detection to trillions of datapoints and ultra-fast arriving data streams. In: Kdd 2022, Proceedings of the 28th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1173–1182. Association for Computing Machinery, New York, NY, USA (2022)
12. Lubba, C.H., et al.: catch22: canonical time-series characteristics: selected through highly comparative time-series analysis. Data Min. Knowl. Discov. **33**(6), 1821–1852 (2019)
13. MacGregor, J.F., Kourti, T.: Statistical process control of multivariate processes. Control Eng. Pract. **27**(2), 427–437 (1994). iFAC Symposium on Advanced Control of Chemical Processes, Kyoto, Japan, 25-27 May 1994
14. MacQueen, J., et al.: Some methods for classification and analysis of multivariate observations. In: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, vol. 1, pp. 281–297. Oakland, CA, USA (1967)
15. Nakamura, T., et al.: Merlin: parameter-free discovery of arbitrary length anomalies in massive time series archives. In: 2020 IEEE International Conference on Data Mining (ICDM), pp. 1190–1195. IEEE, Sorrento, Italy (2020)
16. Rewicki, F., et al.: Is it worth it? Comparing six deep and classical methods for unsupervised anomaly detection in time series. Appl. Sci. **13**, 1778 (2023)
17. Rewicki, F., et al.: EDEN ISS 2020 telemetry dataset. Zenodo (2024). https://doi.org/10.5281/zenodo.11485183
18. Rousseeuw, P.J.: Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. J. Comp. Appl. Math. **20**, 53–65 (1987)
19. Ruiz, A.P., et al.: The great multivariate time series classification bake off: a review and experimental evaluation of recent algorithmic advances. Data Min. Knowl. Disc. **35**(2), 401–449 (2021)
20. Sitthiyot, T., Holasut, K.: A simple method for measuring inequality. Palgrave Commun. **6**(1), 112 (2020)
21. Sohn, K., et al.: Anomaly clustering: grouping images into coherent clusters of anomaly types. In: 2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), pp. 5468–5479 (2023)
22. Tafazoli, S., et al.: Matrix profile xxix: C 22 MP, fusing catch 22 and the matrix profile to produce an efficient and interpretable anomaly detector. In: 2023 IEEE International Conference on Data Mining (ICDM), pp. 568–577. IEEE (2023)
23. Xhimitiku, I., et al.: Anomaly detection in plant growth in a controlled environment using 3D scanning techniques and deep learning. In: 2021 IEEE Int. Workshop on Metrology for Agriculture and Forestry (MetroAgriFor), pp. 86–91 (2021)

Long-Term Fairness in Ride-Hailing Platform

Yufan Kang[1], Jeffrey Chan[1(✉)], Wei Shao[2], Flora D. Salim[3],
and Christopher Leckie[4]

[1] RMIT University, Melbourne, VIC, Australia
`yufan.kang@student.rmit.edu.au, jeffrey.chan@rmit.edu.au`
[2] Data61, CSIRO, Clayton, VIC, Australia
[3] University of New South Wales, Sydney, NSW, Australia
`flora.salim@unsw.edu.au`
[4] The University of Melbourne, Carlton, VIC, Australia
`caleckie@unimelb.edu.au`

Abstract. Matching in two-sided markets such as ride-hailing has recently received significant attention. However, existing studies on ride-hailing mainly focus on optimising efficiency, and fairness issues in ride-hailing have been neglected. Fairness issues in ride-hailing, including significant earning differences between drivers and variance of passenger waiting times among different locations, have potential impacts on economic and ethical aspects. The recent studies that focus on fairness in ride-hailing exploit traditional optimisation methods and the Markov Decision Process to balance efficiency and fairness. However, there are several issues in these existing studies, such as myopic short-term decision-making from traditional optimisation and instability of fairness in a comparably longer horizon from both traditional optimisation and Markov Decision Process-based methods. To address these issues, we propose a dynamic Markov Decision Process model to alleviate fairness issues currently faced by ride-hailing, and seek a balance between efficiency and fairness, with two distinct characteristics: (i) a prediction module to predict the number of requests that will be raised in the future from different locations to allow the proposed method to consider long-term fairness based on the whole timeline instead of consider fairness only based on historical and current data patterns; (ii) a customised scalarisation function for multi-objective multi-agent Q Learning that aims to balance efficiency and fairness. Extensive experiments on a publicly available real-world dataset demonstrate that our proposed method outperforms existing state-of-the-art methods.

Keywords: Applications · Fairness · Human-aware Planning and Scheduling

1 Introduction

Ride-hailing systems have become increasingly prevalent as a mode of transportation, with platforms, such as Uber, utilising artificial intelligence (AI) algo-

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
A. Bifet et al. (Eds.): ECML PKDD 2024, LNAI 14949, pp. 223–239, 2024.
https://doi.org/10.1007/978-3-031-70378-2_14

rithms to match drivers to passengers efficiently [11]. However, while these algorithms succeed in optimising earnings for drivers and reducing passenger waiting time, they often result in inequities, such as wage disparities based on gender or race [2]. As a result, there is a growing interest in the fair allocation of jobs to drivers in ride-hailing systems.

A majority of the established research on creating equitable ride-hailing systems relies heavily on heuristic or linear programming approaches [6,7,12,18,22], augmented by mechanisms that promote fairness. These traditional algorithms have benefits such as simplicity in structure and reasonable execution times. However, they fall short of guaranteeing non-myopic solutions that can make far-sighted decisions. In addition, existing studies do not utilise longitudinal historical data to identify future patterns for raised requests. These algorithms focus on the immediate future, which lack the predictive capacity for future demand trends and fail to consider historical discrepancies. Raman *et al.* proposed an allocation system based on a Markov Decision Process (MDP) to balance efficiency and fairness, which provides non-myopic allocation plans [14]. However, the approach to addressing equity issues in the ride-hailing system only considers historical patterns for raised requests, without any consideration for future conditions.

Figure 1 illustrates a potential scenario where an emphasis on short-term equity results in a long-term inequitable allocation plan. For an allocation system that considers short-term fairness, the system tries to improve fairness by only considering the historical allocations. Thus, at the end of the second timestep, the allocation system achieves fairness among the three drivers, as all the drivers get the same utility, but the final allocation is comparably unfair due to the upcoming requests from the third timestep. As for the allocation system considering long-term fairness, the allocation system is able to consider the future requests patterns and improve fairness across the entire time duration to output a fair allocation plan. Given the existence of multiple possible scenarios where riders raising requests following different distribution, without considering future patterns, long-term equity becomes increasingly unstable, thus underscoring the necessity of a more comprehensive approach.

Importantly, in practical contexts, drivers in ride-hailing systems are generally more concerned with their weekly earnings, which can be viewed as long-term earnings as opposed to daily earnings [4,21]. The challenge of maintaining long-term fairness can be attributed to three key factors: i) **Short Sight**: optimisation algorithms without look-ahead time fall short in harnessing long-term historical data and forecasting future patterns. The example shown in Fig. 1 illustrates the challenge. ii) **Concept Drift**: similar to concept drift in time-series forecasting, existing methods that consider fairness presuppose that future ride requests will adhere to previously observed patterns. However, this assumption is fundamentally flawed as the patterns of ride requests fluctuate continuously due to different factors, including weekday peak hours and public holidays. By its nature, the ride-hailing system necessitates real-time assignments, rendering it unfeasible to accumulate a sufficient volume of ride requests to explore dif-

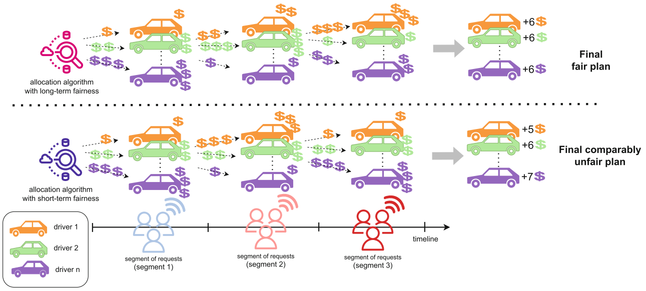


Fig. 1. The figure showcases the disparity in fairness concerns between short-term and long-term allocations by displaying two allocation systems generating allocation plans at each timestep. The arrows pointing to vehicles represents allocated requests with different utility to different drivers, and the dollar signs next to drivers indicate the total utility accumulated by the end of each timestep. In this example, the algorithm that prioritizes short-term fairness manages to achieve absolute fairness by the end of the second timestep, with no variance in utility among drivers. However, it becomes unfair by the third timestep. On the other hand, the algorithm that focuses on long-term fairness appears relatively unfair at the end of the second timestep but ultimately achieves absolute fairness by the third timestep.

ferent patterns of raised requests before optimising assignments. iii) **Disparity between Utility and Fairness Increase across Time**. Multiple fairness definitions have been proposed to evaluate fairness among drivers in ride-hailing systems. An earlier study addressed this by attempting to maximise the minimum utility amongst drivers, treating the problem as an instance of matching issues [7]. Nonetheless, the focus of the study on improving fairness is mainly on enhancing the utility for the driver with the lowest utility, thus presenting a rather restricted view of fairness. The study did not consider fairness among all drivers but to improve fairness for the driver with the lowest utility instead. Another approach is to evaluate fairness among drivers based on the variance in travel distance [14], which can be considered a comparably more global evaluation of fairness. Research findings indicate a conflict between the optimisation of utility and the promotion of fairness within these systems [6,7,12,14,18,19,22]. Therefore, the principal challenge of fairness in ride-hailing lies in reconciling these two objectives. The objective function is normally defined to balance total utility and fairness (e.g., $utility + \lambda fairness$) [7,14,19]. Whilst we adopt the concept of fairness defined based on variance, we want to highlight the challenge in maintaining this balance over a long time horizon, as the variance inevitably increases, which thus intensifies the preference for fairness. The question of balancing utility and fairness whilst preventing an escalation in the weight of fairness over time remains unresolved.

Therefore, to effectively mitigate these issues, we propose a long-term fairness approach for ride-hailing systems, one that leverages both historical data and forecasts of future scenarios. We propose a Markov Decision Process (MDP)-based method to deliver long-term fairness while balancing it with total utilities for drivers. Our proposed model leverages a multi-objective multi-agent Q-learning (MOMAQL) technique, allowing an allocation plan approach to non-myopic solution that considers various possible data patterns to mitigate the unfairness illustrated in Fig. 1. In the proposed MDP-based approach, we include a time-series forecasting module which is used to predict future requests, enabling the model to anticipate future patterns. In this case, assuming a driver gets the lowest wage in a short-term time period, the driver can be compensated in the subsequent time periods.

Our key contributions can be summarised as follows:

- We formally propose a concept of long-term fairness in ride-hailing systems to promote equal earnings among drivers over a broader time horizon, aligning more closely with their focus on weekly earnings.
- We present a novel MDP-based model with appropriate objective and scalarisation functions. This model aims to preserve utilities whilst minimising the disparity in earnings among drivers.
- We introduce a predictive module that forecasts the volume of future ride requests in different locations. This module is incorporated into the MDP-based model to allow a look-ahead time for the MDP-based model and thus help achieve long-term fairness in ride-hailing systems.
- We validate our proposed model through experiments using a real-world dataset, demonstrating that it outperforms existing methods.

2 Related Work

The problem of matching ride requests and drivers has been conceptualised as either a bipartite matching problem [24] or a Markov Decision Process (MDP) [14,16,19]. These studies have demonstrated substantial improvements in efficiency, enabling drivers to service more requests and increase their earnings. However, recent literature has brought equity concerns [13]. From the perspective of riders, Brown *et al.* underscored differential treatment of riders by ride-hailing services, leading to an amplified trip cancellation rate for riders with darker skin tones [1]. Similarly, on the driver side, a subset of ride-hailing drivers struggle to earn a sustainable income due to systemic income disparities [2]. To counteract these disparities, some research studies have proposed the use of bipartite matching with a min-max objective function aimed at maximising the minimum utility, thereby promoting more balanced utilities among drivers [7]. Subsequent research has demonstrated boundaries on the trade-off between efficiency and a defined concept of fairness [9,12]. Upon the aforementioned works, investigations have been carried out focusing on equalising utilities among drivers and riders [5,18]. Additionally, Li *et al.*. proposed a novel method to augment the fairness of ride-hailing demand functions [23]. Expanding on previous research pertaining to ride-hailing [7], Raman *et al.*. formulated a versatile methodology that can be adapted for ride-hailing matching to improve fairness through applying an MDP [14]. While the work provides a non-myopic solution compared to the previously mentioned studies, the study still focuses on promoting short-term fairness like other studies without an estimation of patterns of future requests.

3 Preliminaries

Studies revealed that there is a conflict between fairness and utility, and optimising fairness without considering utility will cause all the drivers not to get any assignment (no earnings for everyone is absolutely fair) [12,17,18]. Thus, to consider fairness in ride-hailing, the target is to balance between utility and fairness.

3.1 Problem Formulation

Assume there are n drivers $v \in V$ who aim to pick up different orders sent by riders, and all drivers and riders reside on a directed graph. The set of locations, L, are represented as nodes in this directed graph which can be locations for the n drivers or either pickup or drop-off locations for rider-raised requests. The set of directed edges, E, where $e_{i,j} \in E$, represents the travel distance between locations $l_i, l_j \in L$. In the real-world scenarios, it is normal that the travel distance from a location (e.g. l_i) to another location (e.g., l_j) is different from travelling in the opposite direction. Thus, we define the real-world map as a directed graph. We define the utility of a trip between the location l_i and l_j based on $e_{i,j}$ according to existing studies [14,20].

The rider requests, driver states and time length are formulated as follow:

- A rider request $r \in R$ and $r = (t_r, s_r, d_r)$ where the request r is raised at time t_r from locations s_r to d_r; $s_r, d_r \in L$.
- The state of each of the n drivers at time t can be represented by the tuple $v^t \in V^t$ and $v^t = (c_v, m_v^t, g_v^t, o_v^t)$, where c_v represents the capacity of the vehicle that driver v is driving, m_v^t is the number of riders in the vehicle at time t, g_v^t is the geographical location of v at time t ($g_v^t \in L$) and g_v^t is dynamically changing based on t with $g_v^t = -1$ if the driver is travelling on an edge between two different nodes, $o_v^t(R_v)$ is the total utilities gained by v from the requests that are assigned to v, represented by R_v, from starting time t_0 to t.
- The time length $T = \{\{t_{0-\delta},\ t_{0-\delta+1},\ ...,\ t_{0-1}\},\ t_0,\ \{t_1,\ t_2,\ ...,\ t_n\}\}$ of the problem is formulated in three parts: the time length $\{t_{0-\delta},\ t_{0-\delta+1},\ ...,\ t_{0-1}\}$ when the historical orders are already completed or being processed, the time step t_0 when the current orders have been raised but not been assigned to the drivers, and the time length $\{t_1,\ t_2,\ ...,\ t_n\}$ when the future orders will need to be assigned to the drivers.

It is important to note the utility of each request r is calculated by $Geo(d_r, s_r) - Geo(s_r, g_v^t)$ where Geo calculates the shortest geographical distance between two points. The utility reflects the balance between profit, defined as the distance from the start to end location for a request (s_r to d_r) and cost, represented by the distance from the current location of the driver (g_v^t) to the start location of the request (s_r). The utility calculation is performed at the moment a request is assigned. We assume drivers opt for the shortest possible route, meaning the utility is consistent for requests with identical start and end points. Upon assignment of a request r to a driver v, its utility is added to the cumulative utility of the driver o_v^t.

Requests Prediction. Assume a set of requests raised by different riders across the time length T is represented by $R_T = \{\{R_{t_{0-\delta}}, R_{t_{0-\delta+1}}, ..., R_{t_{0-1}}\}, R_{t_0}, \{R_{t_1}, R_{t_2}, ..., R_{t_n}\}\}$ where $\{R_{t_1}, R_{t_2}, ..., R_{t_n}\}$ represents the future orders that will be raised by different riders. $R_{t_n} = \{(s_r^{t_n}, d_r^{t_n}) | s \in L,\ d \in L\}$ represents orders that will be raised at node $s \in L$ and end at node $d \in L$ at time t_n. We aim to forecast the number of trips raised and ends at each node in the graph.

3.2 Efficiency

This study defines overall efficiency as the total utility acquired across all the drivers over a given timeframe. Given a set of requests $r \in R$, efficiency aims to find an assignment M that assigns each request to exactly one driver $v \in V$ to maximise total utility shown as:

$$\begin{gathered} \pi(M) = \sum_{v \in V} o_v^{t_n}(M(v)) \\ t_n = \max(T) \end{gathered} \tag{1}$$

where $M(v)$ represents the set of requests that are assigned to the driver v according to the assignment M.

3.3 Long-Term Fairness

We define long-term fairness as the accumulated fairness that consider fairness calculated from the historical, current, and future allocation plan which will be the output from the proposed method using testing data. In this study, we use 1 week as the time horizon with the first three-days records treated as historical data, the fourth day treated as the current data, and the last three days treated as the future data which will be considered as testing data for the proposed method. The fairness is calculated as the accumulated fairness through the whole week as drivers care more about weekly-based earnings instead of daily earnings in the real-world. The fairness is defined as the variance of utilities among the n drivers according to existing studies [14], where utilities for each driver is defined as the total utilities through the whole week. Given a set of requests $r \in R$, long-term fairness aims to find an assignment M that assign each request to exactly one driver $v \in V$ to minimise long-term fairness shown as:

$$\begin{gathered} F(M) = \mathrm{Var}\left(o_v^{t_n}(M(v))\right) \\ t_n = \max(T) \end{gathered} \tag{2}$$

where Var represents variance to calculate the variance of total utilities among different drivers.

3.4 Balance of Long-Term Fairness and Total Utility

To maintain a balance between long-term fairness and total utility, we formulate the problem as an operation research question with the formula shown as:

$$max_M \ \pi(M) - \lambda F(M) \tag{3}$$

subject to

$$\sum_{v \in V, r \in R} I_{rv} \leq 1 \tag{4}$$

where λ is the weight for fairness, and I_{rv} is defined as an indicator function:

$$I_{rv} = \begin{cases} 1 & if\ request\ r\ is\ assigned\ to\ vehicle\ v \\ 0 & else \end{cases} \tag{5}$$

4 Approach: Optimising Efficiency and Long-Term Fairness for Ride-Hailing

In this section, we introduce a novel solution to both optimise the efficiency and longer-term fairness for ride-hailing applications. The proposed model adopts a multi-objective multi-agent Reinforcement Learning (MOMARL) algorithm to develop the allocation system, driven by three core considerations:

- **Real-World Dynamics and Initial Conditions**: The varying initial locations of drivers in the real-world significantly influence their ability to serve ride requests, affecting their behaviour and accumulated utility. MOMARL allows different drivers learn different agent behaviours based on their starting locations, thereby optimising total income while promoting fairness.
- **Fairness and Equity Considerations**: To address fairness, defined as the equitable comparison of utility gained by drivers, our system employs a central controller to acknowledge status of all agents. This ensures that allocation decisions consider the collective situation of all drivers, promoting a fair distribution of utility and opportunities across the network. The central controller allows the proposed method dynamically adjusts to real-time conditions and redistributes resources to maintain fairness and efficiency.

We designed the scalarisation function in MOMARL to balance utility and fairness and to approach Pareto Optimal. Additionally, a time-series prediction module is incorporated to provide future available actions in MOMARL to allow the proposed method to consider future requests' pattern. The prediction module is implemented to adapt the dynamics of the requests raised by riders based on time in an online manner.

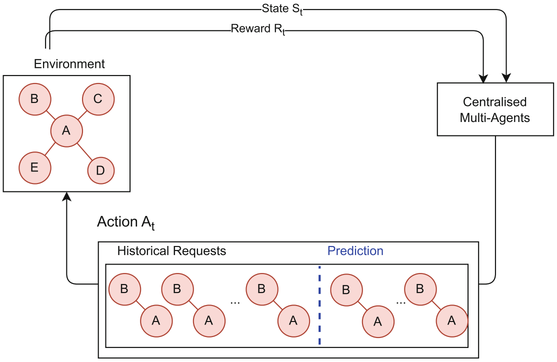


Fig. 2. Long-term Fairness for ride-hailing system. With time-series prediction, the predicted requests is part of the action space of the MDP-based model to allow the outputed allocation plan be based on the pattern of future requests.

4.1 Overview

The proposed model comprises a time-series prediction module, multi-objective Reinforcement Learning, and a scalarisation function.

- Time-series forecasting (Sect. 4.2). To encourage longer-term fairness, we utilise time-series forecasting in the proposed model to predict future requests as part of the input for MOMAQL. The time-series forecasting module exploits historical requests in different locations as input and the output is the number of requests that will be raised in the future from different locations.
- Multi-objective multi-agent Reinforcement Learning (Sect. 4.3). In this study, we exploit multi-objective multi-agent Q Learning (MOMAQL) to construct the fundamental part of the proposed model as Reinforcement Learning has been proven to be an efficient method to construct ride-hailing systems [8]. For each time step, the centralised controller assigns each request to an agent (a driver $v \in V$) if $g_v^t \neq -1$. Each objective function of MOMAQL focuses on maximising the utilities for each driver, where the utility of each request is calculated based on the geographical shortest distance. In this way, MOMAQL here output an allocation plan target on maximising the total utilities among different drivers.
- Scalarisation function (Sect. 4.4). In order to transform the multi-objective problem into a standard single-objective problem, we then propose a scalarisation function. The scalarisation function not only aims to transfer the problem, but by maximising the value of the scalarisation function, it also seeks for a balance between efficiency and fairness to approach Pareto Optimal.

The proposed model operates in four stages in each batch: predicting, evaluating, assigning and learning. We first predict the number of future requests in different locations. Then, each batch starts with an evaluation: when a request is raised by a rider, MOMAQL first finds the shortest path from the start location (the location where the request is raised) to the end location (the required destination), where the travel distance represents the utility of the request. For the assigning stage, MOMAQL assigns the request to a driver based on the value calculated by the scalarisation function, considering both efficiency and fairness. Lastly. the proposed MOMAQL based model learns from the matching results and utilises the scalarisation function to optimise both the efficiency and fairness and approach Pareto Optimal.

4.2 Time-Series Forecasting

The request prediction module is defined based on time-series prediction for which we utilise Multi-Layer Perceptron in this study (MLP) (Fig. 2). MLP consists of an input layer of source nodes, one or more hidden layers, and an output layer. As an existing study stated that a single hidden layer is sufficient to approximate different continuous functions [3], we use a three-layer MLP in our proposed method. We first utilise the pairs of locations (start and destination locations from different requests) as features, then multiple measurements at time $t, (t-1), ..., (t-n)$ are used to predict the requests that will happen in the future (the 7 d that we use to test the model), where each time step is set as 1 h. The structure of the request prediction module has *number* of neurons in the hidden layer. By using the chosen dataset, we use the previous 1 month of data for training and output the number of requests that will happen based on each pair of locations in the next 7 days.

4.3 Multi-objective Multi-agent Q Learning

To convert the ride-hailing assignment problem to a Markov Decision Process (MDP), we define its foundational elements as follows:

- **State**: The states are derived from the various locations where drivers are initially positioned and where ride requests originate and conclude. These locations constitute a finite set of states, encapsulating both driver positions and request locations at any given time point t.
- **Action**: Within the MDP framework, actions at each time point t where $t \in T = \{\{t_{0-\delta},\ t_{0-\delta+1},\ ...,\ t_{0-1}\},\ t_0,\ \{t_1,\ t_2,\ ...,\ t_n\}\}$, represent the strategic assignment of incoming ride requests to available drivers. The driver is requested to drive from the current location to the start location of the request to pick up the rider and drive to the destination. The proposed method allows an agent (driver) to accept multiple requests concurrently. The time series forecasting module is incorporated into the MDP framework to predict available actions at $t \in T$, which enables the anticipation of future requests and their strategic incorporation into current decision-making. This forward-looking capability ensures that actions not only respond to immediate demands but also adapt to predicted future conditions. Additionally, the model supports the option of no action at t, allowing periods where a driver

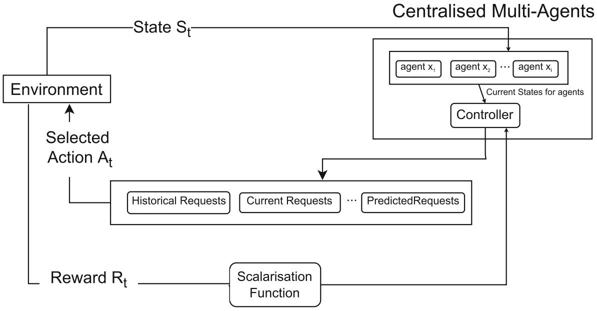


Fig. 3. Multi-objective multi-agent Q Learning. By customising the action space and scalarization function, we aim to encourage the balance between utility and fairness by utilising multi-objective multi-agent Q learning. In action space, it includes historical, current and predicted future requests to allow the proposed model trained based on the pattern of future requests. For scalarisation function, it is designed aiming to balance utility and fairness and maximise the objective.

may not receive any assignments, which provides a mechanism to prevent overburdening drivers and ensuring a fair distribution of work.

- **Reward**: Once each agent (driver) takes action, the agent receives an instant reward shown as:

$$r^t_{s,A}(v) = \sum_{a \in A^t_v} Geo(d^t_a, s^t_a) - Geo(s^t_a, g^t_v) \tag{6}$$

where A^t_v represents the set of actions taken by the agent v at time t, s^t_a represents the starting location of the assigned request, d^t_a represents the ending location of the assigned request, and g^t_v represents the location of v which means the state of the agent at time t.

The proposed model exploits centralised MOMAQL. For each time step, the centralised controller queries the raised requests and assigns requests to different agents that is currently located on a node. To simplify the problem, we set the drivers always to accept requests without cancellation.

4.4 Scalarisation Function

One of the approaches for multi-objective problems relies on single-policy algorithms [10] to learn Pareto Optimal solutions. Single-policy multi-objective Reinforcement Learning algorithms exploit scalarisation functions over the vector-based reward functions, thereby reducing the multi-objective environment's dimensionality to a single, scalar dimension (Fig. 3). Maintaining a balance between optimising utility and improving fairness over a long time horizon is challenging as the two objectives increase at different speeds. Thus we define the scalarisation function with a weight for fairness to adjust its range. The scalarisation function is designed based on Eq. 3 as:

Table 1. Long-term fairness performance compared with baselines on the dataset.

Methods	Total Utility	Fairness	Normalised Fairness	Min	Mean	Max
Greedy	-1514736.24	1696.95	-0.0005	-75803.61	-75736.81	-75653.78
REASSIGN	76536.23	493637.57	0.18	2218.72	3826.81	4760.17
LAF	80606.49	107789.96	0.0814	3001.74	4030.3245	4491.26
Balance Ride-Pooling	85923.68	100254.73	0.074	3451.47	4296.18	4674.44
Proposed Method	95823.79	85194.48	0.061	4565.56	4791.19	5931.93

$$\begin{aligned} \mathrm{SR}(M) &= \textstyle\sum_{v\in V} r_{s,A}(v) \; - \; \lambda\omega \,\mathrm{Var}(r_{s,A}(v)) \\ r_{s,A}(v) &= \textstyle\sum_{t\in T} r^{t}_{s,A}(v) \;, \end{aligned} \tag{7}$$

where $0 \leq \lambda \leq 1$ represents the weight for fairness among different drivers and $0 < \omega \leq 1$ represents the scale to adjust fairness into the same range as utility. By adjusting ω, the weight assigned to fairness is adjusted to avoid fairness getting a larger weight due to the unavoidable increase of variance while the time horizon gradually increases.

5 Experiments

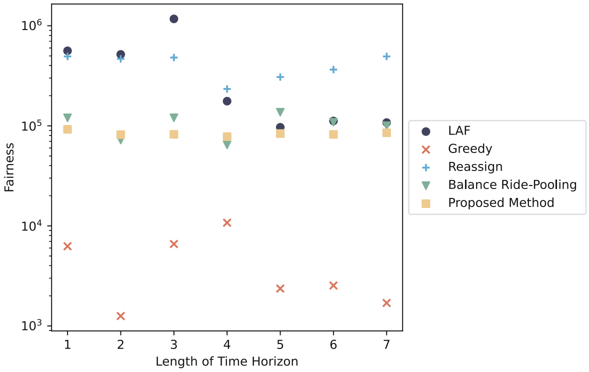


Fig. 4. Performance of baselines and proposed model in terms of fairness based on gradually increased time horizon

5.1 Datasets

We exploit New York City Taxi dataset[1], a publicly available taxi trip dataset collected in New York City, which contains essential information on all requests

---

[1] https://www.nyc.gov/site/tlc/about/tlc-trip-record-data.page.

executed by active taxis including the day and time when a request is raised, the location where the request is raised, and the dropoff location it required. Each request's pickup and drop-off locations are recorded in longitude and latitude coordinates. We extract all requests starting and ending within Manhattan, happening on the dates ranging from 01/03/2016 to 01/04/2016. For simplicity, the shortest travel time from a certain pickup to a certain drop-off location is re-calculated as the mean travel time across the time period, and multiple locations are merged together as a node in constructed graph. This study assumes that the driver will always choose the shortest path to complete a request.

5.2 Experimental Details

Baselines. We select three existing fair ride-hailing methods and Greedy with the objective to balance efficiency and fairness as baselines:

- Greedy. We implement Greedy with the objective of balancing efficiency and fairness according to Eq. 3.
- REASSIGN. REASSIGN exploits traditional optimisation with the objective of balancing efficiency and fairness. To compare with our proposed method, the fairness definition is modified in REASSIGN according to Eq. 2. In the study, Lesmana *et al.* state that their proposed method can be applied with various fairness definitions [7].
- LAF. The study conducted by Shi *et al.* exploits a Markov Decision Process as a re-weighting module to refine the weight for each edge to promote fairness. LAF then utilise Hungarian algorithm to optimise total utility and output the final allocation plan. LAF is used as one of the baseline with the fairness definition modified according to Eq. 2 [17].
- Balance Ride-Pooling[2]. The study conducted by Raman *et al.* exploits a Markov Decision process to optimise the number of rider requests serviced while maintaining fair earning among drivers. We use the method proposed by Raman *et al.* as one of our baselines [14]. The fairness definition they used in their study is similar to the fairness definition in our study. Hence, the implementation remains unchanged.

Experimental Settings. We selected data before 26/03/2016 as the training data and predicted the requests from 26/03/2016 to 01/04/2016. To reduce the training time, we extracted peak 2-hour data ranging from 19/03/2016 to 01/04/2016 and for the request prediction output. We then use the first seven days data and the extracted output from time-series forecasting to train our proposed model and test the remaining data. During the training process, we used a stratified sampling method with a sampling rate of 0.05 for the training data. We set $\lambda = 1$ (a parameter shown in Eq. 7) to indicate no preference on utility or fairness, $\omega = 0.6$ (a parameter shown in Eq. 7) to scale utility and fairness into the same range, $\gamma = 0.9$ as the value for discount factor for

---

[2] The study has not provided a formal name.

MOMAQL. All experiments are trained and tested on a Linux system (CPU: Intel(R) Xeon(R) Gold 6240 CPU @2.60GHz, GPU: NVIDIA GeForce RTX 8000).

5.3 Results and Analysis

As the range of variance varies significantly based on the attained total utility, we further utilise normalised fairness as another measurement to show experimental results. In this study, we define normalised fairness as normalised standard deviation among the utilities for different drivers shown as:

$$\hat{F}(M) = \frac{\sigma(U)}{\overline{U}}, \tag{8}$$

where U represents vector records accumulated utility by each driver and $\overline{U}$ represents the mean utility across all drivers.

Table 1 and Fig. 4 summarise the results of the proposed method compared to the baselines on the real-world dataset. We further tested the performance of each method by gradually increasing the prediction horizon to test the stability of fairness in terms of the time horizon. All the experiments are conducted under the same experimental settings.

Long-Term Fairness Performance Comparison. Under this setting, each method attains an optimised allocation by using the whole seven days testing data. For Table 1, we can observe that: (1) The two objectives, fairness and efficiency, are contradicted to a certain level. For Greedy, with the optimised allocation focusing more on fairness, the total utility can even reach negative, which cannot be a solution in the real-world. In order to obtain a fair result, Greedy tends to sacrifice the utility of all the other drivers to achieve a fair result based on the driver with lowest utility instead of increasing the utility for the driver. Essentially, all the drivers are allocated to the requests with the lowest utility which leads to the final result with the negative value for efficiency. (2) Comparing REASSIGN (based on traditional optimisation) with the Proposed Method (based on Reinforcement Learning), we can see that the proposed method perform better, which further supports the statement made by Shah *et al.* [15] that traditional optimisation makes comparably more myopic decisions compared to Reinforcement Learning. (3) Balance Ride-Pooling proposed by [14] is also based on Reinforcement Learning, Compared with our proposed model, the method does not consider the patterns of requests in the future, which indicates the dependency of future patterns can improve the total utility for the drivers. (4) The proposed method achieves more balanced results in terms of efficiency and fairness compared to Greedy and outperforms other baselines.

Table 2. Ablation study. Long-term fairness performance compared with the proposed method excluding different modules on the selected dataset.

Methods	Total Utility	Fairness
Our Method	95823.79	85193.62
Our Method w/o Prediction	56873.21	153697.27
Our Method w/o Fairness	2194901.19	2677473902.27

Stability of Long-Term Fairness. We aim to compare existing methods with our proposed model in terms of fairness with various time horizons. Figure 4 shows that: (1) With predicted future patterns, fairness is lower when the length of the time horizon is 1 and gradually improves with increases in the time horizon and approaches a stable value after the length of the time horizon is equal to 4. This is our target as long-term fairness focuses on the stability and value of fairness when the time horizon is longer, as drivers will care more about the equity of profits in the long term compared to the short term. (2) Comparing REASSIGN with Balance Ride-Pooling and the Proposed Method, it shows that the output from Reinforcement Learning based methods are comparably more stable when the length of the time horizon increases. (3) Compared with Balance Ride-Pooling, our proposed method includes the requests prediction, which allows the proposed model output allocation plan with dependency on future situations to achieve a more stable result for fairness.

5.4 Ablation Study

Table 2 and Fig. 5 show the ablation study to answer two following questions:

How well does the request prediction module balance total utility and fairness? In this study, Table 2 shows the performance comparison of the proposed method, proposed method without prediction, and proposed method without fairness. The proposed method without fairness does not consider fair earning among different drivers but leads to the highest total utility. By considering fairness, comparing the proposed method with the proposed method without fairness, it shows the request prediction module not only encourages fairer earning among different drivers but also increases the total utility. The Mean Squared Error of the request prediction module is 94.69, and the reason for the improvement on total utility and fairness is that the proposed method utilises the predicted future requests in the training process, which further updates the Q-Table based on future patterns of the requests.

How well does the request prediction module work on the stability of fairness with gradually increasing time horizons? Figure 5 shows the performance of the stability of fairness with gradually increasing time horizons on the proposed method, proposed method without prediction, and proposed method without fairness. For the proposed method without fairness, as the method does not consider fairness in the allocation, the fairness is comparably unstable with high unfairness. The earnings instability among different drivers can cause issues in the real world. Comparing the proposed method and the proposed method without a prediction module, the proposed method without the prediction module achieves fairer results when the length of the time horizon is 1 and 2. As the length of the time horizon increases to 3 or more, the fairness of the proposed method is better than the proposed method without prediction and more stable. It shows the prediction module helps to achieve fairer and more stable results in the long-term with sacrifice on the short-term, and we argue that long-term fair earning is what the drivers desire in reality.

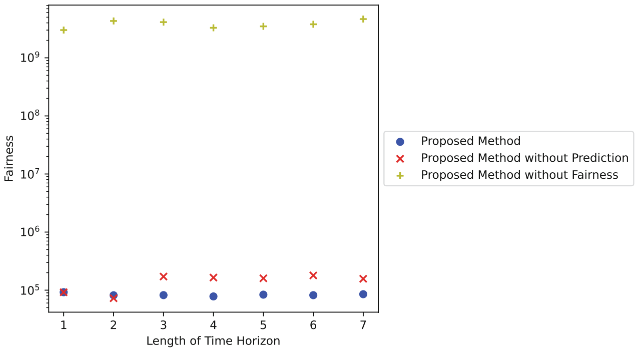


Fig. 5. Ablation study. Performance of the proposed model without different modules in terms of fairness and gradually increased time horizon, where the time horizon is increased by a number of days. For fairness, the larger value indicates the model is unfairer.

6 Conclusion

In this paper, we formally proposed long-term fairness, which focuses on achieving stable and comparably higher fairness over comparably longer time horizons. We argue that taxi drivers will care more about long-term earnings. To achieve the target, we introduce a request prediction module before allocation to allow look-ahead windows for the proposed allocation system and eliminate the assumption that requests always follow the same pattern. We exploit the output as part of the action space for the allocation model, which we designed using Multi-objective Multi-agent Q Learning. The experiments on real-world data demonstrated the effectiveness of our proposed method for maintaining overall fairness in the comparably longer time horizon and enhancing the stability of fairness when the time horizon gradually increases.

Acknowledgments. We acknowledge the support of the Australian Research Council (ARC) Centre of Excellence for Automated Decision-Making and Society (ADM+S) (CE200100005). This research was partially supported by NVIDIA Academic Hardware Grant Program. This research was partially supported by Google Travel Grant. This research was undertaken with the assistance of computing resources from RACE (RMIT AWS Cloud Supercomputing).

References

1. Brown, A.E.: Ridehail Revolution: Ridehail Travel and Equity in LA. UCLA (2018)
2. Cook, C., Diamond, R., Hall, J.V., List, J.A., Oyer, P.: The gender earnings gap in the gig economy: evidence from over a million rideshare drivers. Rev. Econ. Stud. **88**(5), 2210–2238 (2021)
3. Greenwood, P.E., Nikulin, M.S.: A Guide to Chi-squared Testing, vol. 280. Wiley (1996)
4. Huang, J., et al.: Deep reinforcement learning-based trajectory pricing on ride-hailing platforms. ACM TIST **13**(3), 1–19 (2022)
5. Kang, Y., Zhang, R., Shao, W., Salim, F.D., Chan, J.: Promoting two-sided fairness in dynamic vehicle routing problem. arXiv preprint arXiv:2405.19184 (2024)
6. Kumar, A., Vorobeychik, Y., Yeoh, W.: Using simple incentives to improve two-sided fairness in ridesharing systems. In: Proceedings of ICAPS, pp. 227–235 (2023)
7. Lesmana, N., Zhang, X., Bei, X.: Balancing efficiency and fairness in on-demand ridesourcing. In: Advances in Neural Information Processing Systems, vol. 32 (2019)
8. de Lima, O., Shah, H., Chu, T.S., Fogelson, B.: Efficient ridesharing dispatch using multi-agent reinforcement learning. arXiv preprint arXiv:2006.10897 (2020)
9. Ma, W., Xu, P., Xu, Y.: Group-level fairness maximization in online bipartite matching. arXiv preprint arXiv:2011.13908 (2020)
10. Miettinen, K., Mäkelä, M.M.: On scalarizing functions in multiobjective optimization. OR Spectrum **24**(2), 193–213 (2002)
11. Möhlmann, M., Henfridsson, O.: What people hate about being managed by algorithms, according to a study of Uber drivers. Harvard Bus. Rev. **30**, 1–7 (2019)
12. Nanda, V., Xu, P., Sankararaman, K.A., Dickerson, J., Srinivasan, A.: Balancing the tradeoff between profit and fairness in rideshare platforms. In: Proceedings of AAAI, pp. 2210–2217 (2020)
13. Rahmattalabi, A., et al.: Exploring algorithmic fairness in robust graph covering problems. In: Proceedings of NeurIPS (2019)
14. Raman, N., Shah, S., Dickerson, J.: Data-driven methods for balancing fairness and efficiency in ride-pooling. arXiv preprint arXiv:2110.03524 (2021)
15. Shah, S., Lowalekar, M., Varakantham, P.: Neural approximate dynamic programming for on-demand ride-pooling. In: Proceedings of AAAI, pp. 507–515 (2020)
16. Shao, W., Peng, Z., Kang, Y., Xiao, X., Jin, Z.: Early spatiotemporal event prediction via adaptive controller and spatiotemporal embedding. In: 2023 IEEE International Conference on Data Mining (ICDM), pp. 1307–1312 (2023)
17. Shi, D., Tong, Y., Zhou, Z., Song, B., Lv, W., Yang, Q.: Learning to assign: towards fair task assign. in large-scale ride hailing. In: Proceedings of KDD, pp. 3549–3557 (2021)
18. Sühr, T., Biega, A.J., Zehlike, M., Gummadi, K.P., Chakraborty, A.: Two-sided fairness for repeated matchings in two-sided markets. In: Proceedings of KDD, pp. 3082–3092 (2019)
19. Sun, J., Jin, H., Yang, Z., Su, L., Wang, X.: Optimizing long-term efficiency and fairness in ride-hailing via joint order dispatching and driver repositioning. In: Proceedings of KDD, pp. 3950–3960 (2022)

20. Tong, Y., Zhou, Z., Zeng, Y., Chen, L., Shahabi, C.: Spatial crowdsourcing: a survey. VLDB J. **29**, 217–250 (2020)
21. Wu, Y., Li, Q., Qin, Z.: Spatio-temporal incentives opt. for ride-hailing services with offline deep reinforcement learning. arXiv preprint arXiv:2211.03240 (2022)
22. Xu, Y., Xu, P.: Trading the system efficiency for the income equality of drivers in rideshare. arXiv preprint arXiv:2012.06850 (2020)
23. Yan, A., Howe, B.: Fairness-aware demand prediction for new mobility. In: Proceedings of AAAI, pp. 1079–1087 (2020)
24. Zhao, B., Xu, P., Shi, Y., Tong, Y., Zhou, Z., Zeng, Y.: Preference-aware task assignment in on-demand taxi dispatch. In: Proceedings of AAAI, pp. 2245–2252 (2019)

A Merge Sort Based Ranking System for the Evaluation of Large Language Models

Chenchen Li, Linfeng Shi, Chunyi Zhou, Zhaoxin Huan, Chengfu Tang, Xiaolu Zhang, Xudong Wang, Jun Zhou(✉), and Song Liu(✉)

AntGroup, HangZhou, China
`{lichenchen.lcc,john.slf,zhouchunyi.zcy,zhaoxin.hzx,tangchengfu.tcf,yueyin.zxl,xiumin.wxd,jun.zhoujun,yanda.ls}@antgroup.com`

Abstract. Efficient and accurate evaluation of Large Language Models (LLMs) is essential for progress in the field of natural language processing. To address this, our paper introduces Transitive Merge Sort (TMS), a novel method that harnesses the advantages of merge sort's efficiency, stability and parallelizability for model ranking in LLMs evaluation. This approach applies a divide-and-conquer strategy for pairwise comparisons, streamlining the evaluation process. Our experimental findings reveal that TMS not only improves the accuracy of model rankings when compared to methods like Elo rating and SuperCLUE (compared with GPT-3.5) but also significantly reduces the need for annotation resources by up to 70%. Additionally, we present an iterated version of TMS that effectively handles scenarios where initial model rankings are unknown.

Keywords: Merge sort · Pairwise comparison · Model evaluation

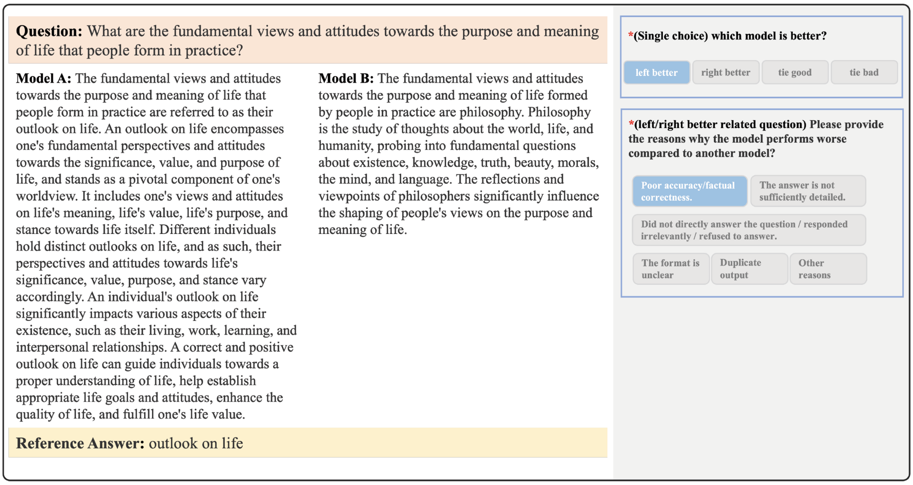
1 Introduction

Recently, the rapid advancement of large language models has led to their widespread adoption in both academic and industry domains [2,14]. This trend has underscored the need for objective evaluation [3,9] of their capabilities, to assist users in identifying the model that best meets their specific requirements. In light of this, the field has witnessed a significant growth in research [6,7,12,17,19,20] focused on the evaluation of large language models. Within the realm of evaluation, ranking systems are pivotal for ordering models according to their performance, currently, there are mainly two types of ranking methods used for evaluating LLMs:

C. Li and L. Shi—These authors contributed equally to this work.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-70378-2_15.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
A. Bifet et al. (Eds.): ECML PKDD 2024, LNAI 14949, pp. 240–255, 2024.
https://doi.org/10.1007/978-3-031-70378-2_15



Question: What are the fundamental views and attitudes towards the purpose and meaning of life that people form in practice?

Model A: The fundamental views and attitudes towards the purpose and meaning of life that people form in practice are referred to as their outlook on life. An outlook on life encompasses one's fundamental perspectives and attitudes towards the significance, value, and purpose of life, and stands as a pivotal component of one's worldview. It includes one's views and attitudes on life's meaning, life's value, life's purpose, and stance towards life itself. Different individuals hold distinct outlooks on life, and as such, their perspectives and attitudes towards life's significance, value, purpose, and stance vary accordingly. An individual's outlook on life significantly impacts various aspects of their existence, such as their living, work, learning, and interpersonal relationships. A correct and positive outlook on life can guide individuals towards a proper understanding of life, help establish appropriate life goals and attitudes, enhance the quality of life, and fulfill one's life value.

Model B: The fundamental views and attitudes towards the purpose and meaning of life formed by people in practice are philosophy. Philosophy is the study of thoughts about the world, life, and humanity, probing into fundamental questions about existence, knowledge, truth, beauty, morals, the mind, and language. The reflections and viewpoints of philosophers significantly influence the shaping of people's views on the purpose and meaning of life.

Reference Answer: outlook on life

***(Single choice) which model is better?**

***(left/right better related question) Please provide the reasons why the model performs worse compared to another model?**

Poor accuracy/factual correctness.
The answer is not sufficiently detailed.
Did not directly answer the question / responded irrelevantly / refused to answer.
The format is unclear
Duplicate output
Other reasons

Fig. 1. Pairwise Comparison

- **Single answer grading** quantifies a model's answer with a definitive score ranging from 0 to 1. This quantification is accomplished either by comparing whether the model's answer matches golden label [6,7] or by leveraging a strong LLM to assign a score to the answer [13,19]. Then we can average the scores to rank the models.
- **Pairwise comparison** determines model's relative performance by comparing their answers, instead of directly scoring. Figure 1 demonstrate an example of pairwise comparison. At the top of the figure is the question, with answers from two anonymous models presented below. On the right, the comparison results are shown along with an explanation of why one model's answer is worse than the other's. This approach is suitable for the assessment of open-ended questions, capturing the models' ability from the perspective of users. The comparison process can be conducted either through human annotation or by the judgment of LLMs [17,19].

In this paper, we adopt the pairwise comparison for model ranking, and refer to evaluation based on pairwise comparison as adversarial evaluation. Adversarial evaluation determines the quality of models by comparing their answers, which aligns more closely with the human experience of using models than single answer grading. Despite its advantages, this approach encounters two significant challenges:

- **Selection of comparative models** During each round of adversarial process, we require a strategy for choosing pairs of models to compare that takes into account the historical adversarial performance of the models.
- **Quantification of comparison results** Once acquiring comparison results, we need to map the relative rankings to absolute values for a quantitative evaluation of the models' abilities.

To best of our knowledge, there is limited research on this topic. Chatbot arena [19] uses the Elo rating [5] system for model ranking, where models of similar ability levels are matched against each other for competition. However, due to the sensitivity of the Elo rating system [18] to the order of matches, the ranking result is relatively unstable. Besides, SuperCLUE [17] ranks models by directly comparing each one with a specified benchmark model (ChatGPT) and calculating scores based on the point-scoring system. Although it reduces the number of comparisons, it fails to accurately reflect the relative relationships among the models.

To address the aforementioned challenges in adversarial evaluation, we propose Transitive Merge Sort (TMS) which applies the merge sort principle for ranking models in LLMs evaluation. We employ the merge process of merge sort to construct the model pairs for comparison and utilize historical comparison records to further reduce the number of model comparisons. Overall, the main contributions of this paper are as follows:

- We propose Transitive Merge Sort (TMS) ranking method for LLM evaluation, which effectively reduces the number of model comparisons while maintaining the accuracy of the ranking results.
- We conducted experiments to assess TMS's performance in terms of comparison count, accuracy, and stability, confirming that it outperformed other methods.
- We develop a LLMs evaluation platform that integrates features such as dataset generation, adversarial evaluation, which allows for the efficient evaluation of LLMs.

2 Related Work

Single Answer Grading. Single answer grading assigns a score to a model answer directly, which is widely used in open LLM leadboard [4,6,7]. For example, in multiple-choice questions, the grading is based on comparing the model's answer with the standard answer to determine whether it is correct or not. Alternatively, similarity calculation methods such as ROUGE [10], BLEU [11], can be used to measure the degree of similarity between the model's answer and the standard answer. Otherwise, like [13,19] explores LLMs as judges to evaluate model's ability.

Pairwise Comparison. For the evaluation of open-ended questions, the pairwise comparison method is commonly adopted. Chatbot Arena [19], for instance, employs the Elo rating system to orchestrate and score matches between models, subsequently ranking them based on their scores. The Elo process operates as follows: initially, two closely matched models A and B, with respective scores $score_A$ and $score_B$ are paired up, then compare the answers of these two models.

$$E_A = \frac{1}{1 + 10^{(score_B - score_A)/400}} \tag{1}$$

The formula (1) calculates the probability E_A that model A will perform better than model B.

$$score'_{A/B} = score_{A/B} + K \times (S_{A/B} - E_{A/B}) \tag{2}$$

After the match, the scores for models A and B are updated in line with the formula (2) and the match outcome. Here, $S_{A/B}$ denotes the result of the match, with 1 indicating a victory, and 0 indicating a loss. The constant K, which influences score adjustments, ranges between 10 and 40. Although the Elo rating system provides an efficient method for calculating model scores based on pairwise comparison, it can struggle with model score stabilization [18].

An alternative ranking method is based on point scoring where the winner of a match receives 1 point, both models receive 0.5 points in the event of a draw, and the loser gains no points. The cumulative points garnered by each model are divided by the total number of matches to obtain the final score, which is then used for ranking.

Round-robin tournament [15] is one implementation of point scoring, wherein all models compete against each other across all samples, tallying scores to establish rankings. This method ensures accurate rankings but incurs high annotation costs due to the amount of comparison.

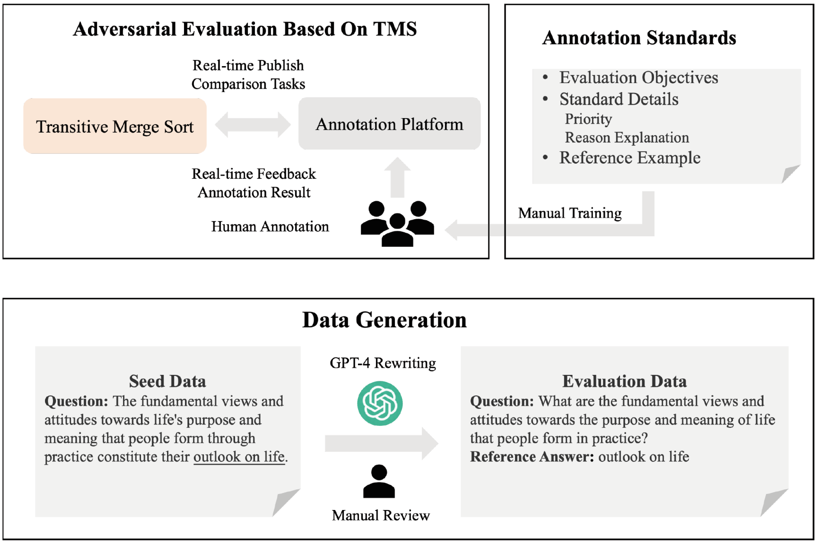
3 Methodologies

Merge sort is a stable sorting algorithm that we incorporate into the evaluation of LLMs to reduce the number of comparisons and ensure the accuracy of the assessment. The advantages of merge sort are described as follows:

- **Divide and conquer strategy**. In pairwise model comparisons, we need to select models for comparison based on a specific strategy. Merge sort breaks down a large problem into smaller ones, allowing us to effectively organize the adversarial process of model comparisons. We can first sort subsets of models and then merge them, reducing the overall number of comparisons and making it easier to choose pairs of models in each round of comparison.
- **Stability**. Merge sort guarantees that models with identical performance receive the same ranking, which is crucial for ensuring consistency in results, while the Elo rating is unstable. When mapping relative rankings of models to absolute values for quantitative evaluation, stability ensures that our assessments are precise and reliable.
- **Efficient**. The time complexity of merge sort is $O(n \times logn)$, which means that when performing numerous pairwise comparisons between models, the required number of comparisons is relatively low, making it more efficient.
- **Parallelization capability**. Compared to the sequential Elo rating system, merge sort is easily parallelizable because it divides the data into independent subsequences for sorting. In model comparisons, this allows for multiple pairwise comparisons to be conducted simultaneously, further speeding up the sorting process.

We propose the Transitive Merge Sort (TMS) rank method, which leverages the benefits of the aforementioned merge sort technique for evaluating LLMs. Building on the TMS, we have developed a platform for LLMs evaluation. Next, we will introduce the architecture of the platform and explain how the TMS works.

3.1 Evaluation Platform



Question: The fundamental views and attitudes towards life's purpose and meaning that people form through practice constitute their outlook on life.

Question: What are the fundamental views and attitudes towards the purpose and meaning of life that people form in practice?
Reference Answer: outlook on life

Fig. 2. Adversarial Evaluation Platform

Figure 2 illustrates the evaluation platform, which includes three modules: evaluation data generation, annotation standards, and adversarial evaluation.

- **Evaluation data generation** [16]. We utilize open-ended questions for evaluation, leveraging GPT-4 to revise questions from the C-Eval and CMMLU datasets, followed by manual verification. Finally, we generate evaluation data that spans four domains: humanities, social sciences, natural sciences, and encyclopedic knowledge, which includes questions of varying difficulty levels suitable for elementary, secondary, university, and professional exams.
- **Annotation standards**. Research [8,13] has shown that using GPT-4 for adversarial evaluations can lead to systematic biases, with an annotation accuracy around 80%, and even lower for more complicated questions. To ensure the accuracy of the evaluation process, expert manual annotation was adopted. To ensure high-quality annotations, we have developed a set of rigorous annotation guidelines, and the annotators were given detailed training

and conducted trial annotations. Official annotations would only start after the accuracy of trial annotations reached 95%. During the official annotation process, a three-person voting method (with more than half in agreement) was used for all evaluation samples.

- **Adversarial evaluation**. The TMS system will rank the specified models across all evaluation samples.

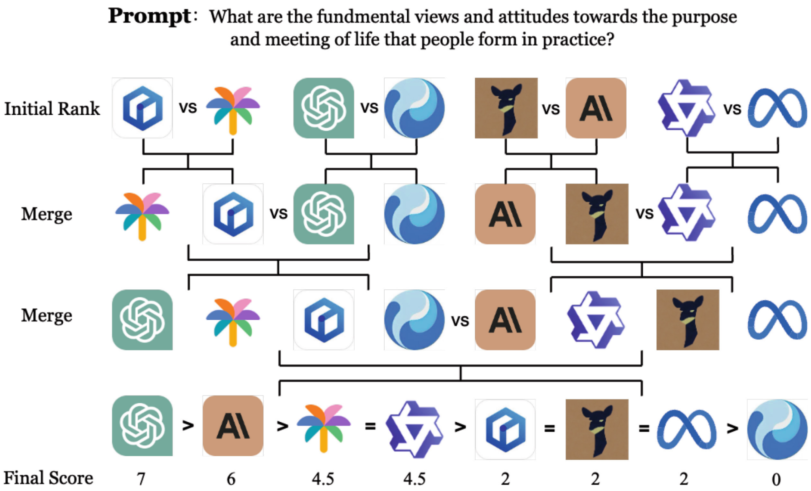

Prompt: What are the fundmental views and attitudes towards the purpose and meeting of life that people form in practice?
Initial Rank
vs
vs
vs
vs
Merge
vs
vs
Merge
vs
>
>
=
>
=
=
>
Final Score
7
6
4.5
4.5
2
2
2
0

Fig. 3. Base Merge Sort Main Process

3.2 Transitive Merge Sort

In this section, we delineate the process of the TMS in adversarial evaluation.

Generally, suppose we have N samples $\mathcal{S} = (s_1, s_2, \cdots, s_N)$ and L models $\mathcal{M} = (m_1, m_2, \cdots, m_L)$, for each sample, a sequence of model pairs will be constructed for comparison, based on a designated strategy. Subsequently, we evaluate the answers from each model within the pairings, denoting the outcomes as 'left better', 'right better', 'tie good' or 'tie bad'.

We utilize the merge sort to construct the comparison pairs, Fig. 3 demonstrates a merge sort process in detail. Firstly, we have a initial model rank $\mathcal{R} = (m_{i_1}, m_{i_2}, \cdots, m_{i_L})$, where $i_1, i_2, \cdots, i_L$ is the shuffle of $1, 2, \cdots, L$. Then, we group the initial model rank $\mathcal{R}$ according to the merge sort algorithm to construct model pairs. The annotators will proceed to label these pairs, which will be used to advance the next round of the merge sort process. The comparison results are marked as $Rel(m_i, m_j)$, which is one of the elements of the set $(>, <, =_{good}, =_{bad})$, '$>$' is left better, '$<$' is right better, '$=_{good}$' denotes that both answers are equally good, while '$=_{bad}$' indicates that they are equally bad.

Algorithm 1 Single Sample Base Merge Sort

Input: Initial model rank $\mathcal{R}$;

Output: Sorted model rank $\mathcal{R}_{final}$;

```
procedure MergeSort($\mathcal{R}$)
    $mid \leftarrow \lfloor length(\mathcal{R})/2 \rfloor$;
    $left$ = MergeSort($\mathcal{R}$[: $mid$]);
    $right$ = MergeSort($\mathcal{R}$[$mid$ :]);
    return Merge($left, right$);
end procedure

procedure Merge($left, right$)
    $merged \leftarrow [\,],\ l \leftarrow 0,\ r \leftarrow 0$;
    repeat
        $Rel \leftarrow Annotate(left_l, right_r)$;
        Add $left_l$ or $right_r$ to $merged$ according to $Rel$;
        Move the $l$ or $r$ by one according to the $Rel$;
    until $left$ or $right$ move to end;
    Append the rest models of $left$ or $right$;
    return $merged$;
end procedure
```

The algorithm described above is the Base Merge Sort (BMS). The whole BMS algorithm is illustrate in Algorithm 1.

During the merge process of the BMS, we do not consider the transitivity of sorting, which could be leveraged to further reduce the number of comparisons required. Provide an example that illustrates the application of transitivity in sorting, starting with an initial model rank $\mathcal{R} = (A, B, C, D)$, suppose we have $Rel(A, B)$ is '$=_{good}$' and $Rel(C, D)$ is '$>$', and in the merge process we get $Rel(A, C)$ is '$>$', then according to the transitivity of sorting, we can infer that $Rel(A, D)$ and $Rel(B, D)$ are both '$>$'. We refer to these method as Transitive Merge Sort (TMS). The detail merge process of TMS is demonstrated in Algorithm 2.

Data Noise in Annotation. During the TMS process, biases in annotators' understanding of annotation standards may lead to occasional mislabeling in the dataset, thereby generating noise in the annotation data. This situation presents a challenge as it disrupts the transitivity between models, leading to inconsistencies. For example, we have the model rank $\mathcal{R} = (A, B, C, D)$, and the comparison results $Rel(A, B)$ is '$=_{good}$', $Rel(C, D)$ is '$=_{good}$'. In the merge process, if we find that $Rel(A, C)$ is '$=_{bad}$', then using transitivity to determine the order between model B and model C becomes uncertain. Annotation noise occurs if and only if the same model is labeled as '$=_{good}$' or '$=_{bad}$' in different annotation tasks, we use $\mathcal{T}$ to denote the group with noisy data:

$$\mathcal{T} = \{(m_i, Rel(m_i, m_j), m_j) \mid m_i, m_j \in \mathcal{M} \text{ and } Rel \in (=_{good}, =_{bad})\} \quad (3)$$

Algorithm 2 Merge process of Transitive Merge Sort

```
Input: Two ranked model list $left$ and $right$;
Output: Merged model list $merged$;
procedure TransMerge($left, right$)
    $merged \leftarrow [\ ],\ l \leftarrow 0,\ r \leftarrow 0$;
    repeat
        $Rel \leftarrow Annotate(left_l, right_r)$;
        // Consider the transitivity of $=_{good}\ \mid\ =_{bad}$
        Find the first model $left_{l'} \mid right_{r'}$ that weak than $left_l \mid right_r$;
        if $Rel$ == '>' then
            $merged.append(left[l:l'])$;
            Update index $l \leftarrow l'$;
        else if $Rel$ == '<' then
            $merged.append(right[r:r'])$;
            Update index $r \leftarrow r'$;
        else
            $merged.append(left[l:l'])$;
            $merged.append(right[r:r'])$;
            Update index $l \leftarrow l',\ r \leftarrow r'$;
        end if
    until $left$ or $right$ move to end;
    Append the rest models of $left$ or $right$;
    return $merged$
end procedure
```

To deal with this problem, we propose a **consistency revision** approach which is based on half-majority voting mechanism. Suppose $\mathcal{P}_{=_{good}}$ and $\mathcal{P}_{=_{bad}}$ represent the proportions of '$=_{good}$' and '$=_{bad}$' in $\mathcal{T}$ respectively. We revise the annotation results by adhering to the principle of the majority rule:

$$Rel(m_i, m_j) = \begin{cases} =_{good} & \mathcal{P}_{=_{good}} >= \mathcal{P}_{=_{bad}} \\ =_{bad} & \mathcal{P}_{=_{good}} < \mathcal{P}_{=_{bad}} \end{cases} \tag{4}$$

Model Scoring. We use point scoring to map model rankings into numerical scores according to the comparison results. In each match, the score of model A is represented as follow:

$$score_A = \begin{cases} score_A + 1 & Rel(A,B) \text{ is } > \\ score_A + 0.5 & Rel(A,B) \text{ is } =_{good} \\ score_A + 0 & Rel(A,B) \text{ is } =_{bad}\ or < \end{cases} \tag{5}$$

From the Eq. 5, we know that model A's score increments by one point when it outperforms model B. In instances where model A and model B perform equally well, model A's score receives a 0.5 point boost. Conversely, when model A performs equally poorly as model B or even worse, its score remains unchanged. This logic also applies to model B.

Algorithm 3 Iterated Transitive Merge Sort

```
Input: Samples $\mathcal{S} = \{s_1, s_2, \cdots, s_N\}$, initial model rank $\mathcal{R}$, batch size $B$;
Output: Final model rank $\mathcal{R}_{final}$;
$score \leftarrow [0, \cdots, 0]$;
for $id \leftarrow 1$ to $N$ do
    $score_{id} = \textsc{TransMergeSort}(s_{id}, \mathcal{R})$;
    $score = score + score_{id}$;
    if $id \ \% \ B == 0$ then
        Sorting $score$ in descending order, obtain the new rank $\mathcal{R}'$;
        Update the rank $\mathcal{R} \leftarrow \mathcal{R}'$;
    end if
end for
```

Since not all models have competed against each other, we determine the ranking order between unmatched models directly based on the transitivity of the ranking results. Specifically, for two models m_i and m_j where $i < j$ that have not competed against each other, we calculate $Rel(m_i, m_j)$ using the following formula:

$$Rel(m_i, m_j) = \begin{cases} > & \exists k \ \text{ s.t. } m_i > m_k > m_j \\ =_{good|bad} & m_i =_{good|bad} m_{i+1} =_{good|bad} \cdots =_{good|bad} m_j \end{cases} \tag{6}$$

The overall score for each model is the sum of points accumulated across all matchups. This scoring system allows us to create a ranking that considers the different weights of model wins and ties. The annotation results for any two models can be determined through either actual annotation or transitive inference.

Stability Improvement. In the aforementioned text, it is mentioned that specifying an initial model order is required at the beginning of the merge sort process. We have discovered that altering the initial order of models can significantly affect the ranking outcomes. This variation is primarily attributed to inconsistencies in the annotation process, which are unavoidable. To tackle this issue, we introduce the iterated Transitive Merge Sort (iter-TMS) method.

In the TMS algorithm, all samples are sorted using the same initial rank of models. As the number of samples increases, the accuracy of the models' ranking becomes more precise. Therefore, the sorting results obtained can be utilized as the new initial rank for the following ranking process. Assume we have a total number of N samples, which are randomly shuffled and then evenly divided into N/B batches based on a batch size of B. We update the TMS algorithm as follows:

$$\begin{aligned} r_1 &= initial\ model\ rank \\ r_i &= TransMergeSort(r_{i-1}, \mathcal{S}_{N\times(i-1)/B \cdots N\times i/B}) \end{aligned} \tag{7}$$

For each batch of samples, TMS is performed sequentially, with the sorting results from the current iteration serving as the initial model rank for the next batch of merge sort. The whole algorithm of iter-TMS is demonstrated in Algorithm 3.

4 Experiments

In this section, we conduct experiments specifically addressing the correctness, efficiency, and stability of TMS, and compare our method with related approaches to validate its effectiveness. Additionally, we perform experimental analyses for dynamic scenarios such as the addition of new datasets.

Table 1. Datasets of adversarial evaluation

Category	Sample Questions
Humanities	Why did the government during the late Qing dynasty and early Republic of China promote Western-style attire among the military, police, and students?
Social Sciences	Why is the principle of equivalent exchange considered to be a principle of fairness?
Natural Sciences	What is the relationship between the amplitude of sound and its pitch?
Encyclopedic Knowledge	What problems can arise from eating sprouted potatoes, undercooked broad beans, and raw leeks?

Table 2. Models of adversarial evaluation

Models	Developer	Size	Access
GPT-4-Turbo	OpenAI	undisclosed	API
Skylark-Chat	ByteDance	undisclosed	API
Wenxin-3.5	Baidu	undisclosed	API
Qwen-72B-Chat	Alibaba	72B	Weights
GPT-3.5-Turbo	OpenAI	undisclosed	API
Baichuan2-13B-Chat	Baichuan	13B	Weights
Qwen-14B-Chat	Alibaba	14B	Weights
Internal-V1	Internal	private	API
Internal-V2	Internal	private	API
Internal-V3	Internal	private	API
Internal-V4	Internal	private	API

4.1 Experiment Settings

Evaluation Datasets. In our research, we use C-Eval and CMMLU as our seed datasets and utilize the rewriting capabilities of GPT-4 to generate 500 adversarial evaluation samples points spanning four domains: humanities, social sciences, natural sciences, and encyclopedic knowledge. Table 1 presents the datasets used in this experiment.

Evaluation Models. We selected 7 external open-source models and 4 internal anonymous models for a total of 11 models to undergo adversarial evaluation. For the 500 samples, we formed pairs from the 11 models, using these model pairs as the experimental data for this research. Among models, the 4 internal anonymous models uniformly start with the prefix 'Internal-', and their rankings are not in any particular order. Table 2 present the models used in this experiment.

Merge Sort Methods. In this paper, we employed merge sort techniques in adversarial evaluation: BMS and TMS, where TMS method accounts for the transitive relationships within the group during the merge process. In sorting process, the selection of the initial model rank $\mathcal{R}$ is crucial. In our experiments, we sorted models according to the size of the model parameters. In cases where the parameter sizes were unknown, we roughly ordered the models based on prior knowledge of their strengths. For the 11 models, we set 10 initial model ranks for comparison between different methods, denoted as $\{\mathcal{R}_1, ..., \mathcal{R}_{10}\}$.

When faced with a large number of models to evaluate, or when evaluators are reluctant to manually determine the initial rank of models, we can employ the iter-TMS method to address these scenarios. In the stability improvement section, we have verified the stability of the iter-BMS and iter-TMS methods when the initial model rank is random.

Compared Methods. To the best of our knowledge, there is limited research on ranking methods of LLMs evaluation to date. This paper primarily selects the following three methods for comparison:

- **Round-robin tournament** involves conducting head-to-head comparisons among all models across the entire set of samples, with the results of these adversarial matchups accumulating points. This method yields the most accurate ranking results and therefore serves as the benchmark for correctness comparison with other methods. However, it also has the poorest performance, with the number of adversarial comparisons amounting to $N * L^2$, where L represents the number of models and N the number of samples.
- **Elo Rating** is utilized on the Chatbot Arena platform. During the competitive process, the Elo system pairs models of similar strength to engage in adversarial battles and updates their ratings based on the outcomes. Similar to Chatbot Arena, we get a more stable model ranking by performing bootstrap 1,000 times on 27,500 model pairs.

- **SuperCLUE** selects a single model as the benchmark model and uses GPT-4 as the rater to evaluate the quality of answers from the benchmark model and other competing models. In our experiments, we designated GPT-3.5 as the benchmark model. To ensure consistency in the evaluation, we replaced GPT-4 with human annotators to compare the outcomes of the models.

Evaluation Metrics. In the context of adversarial evaluation, our primary concern is with the models' final rankings as well as the number of adversarial matchups required. The former reflects the correctness of the results, while the latter indicates the performance of the method. For the assessment of correctness, we use the Spearman's Footrule Distance [1] assess the consistency of rankings across all models and the Root Mean Squared Error (RMSE) to gauge the score discrepancies for each model. It should be highlighted that our attention is centered on the ranking outcomes for all models, as opposed to merely the ordering of the leading models.

Table 3. Comparison of methods based on metrics

Method	Round-robin tournament (Baseline)	Elo Rating	SuperCLUE	BMS	TMS
Spearman's Footrule	0	6.0	8.0	3.4±1.8	**1.8±1.4**
RMSE	0	/	0.115	0.055	**0.049**
#**Matchups**	27500	27500	5000	11249	**7642**

The Spearman's Footrule Distance measures the discrepancy between two rankings, the formula to calculate the Spearman's Footrule Distance is as follows:

$$Dis = \sum_{i=1}^{L} |r_i - r'_i| \tag{8}$$

where Dis is the Spearman's Footrule Distance, L is the number of models being ranked, r_i, r'_i are the rank of model i in the baseline ranking and the predicted ranking.

The Root Mean Squared Error (RMSE) is used to quantify the differences between scores calculated by various methods and is defined as follows:

$$RMSE = \sqrt{\frac{1}{L}\sum_{i=1}^{L}(score_i - score'_i)^2} \tag{9}$$

where $score_i$ and $score'_i$ are the score of model i from baseline method and another method.

For performance assessment, we directly count the number of comparisons required by each method. This straightforward approach allows us to quantify the efficiency of each method in terms of the computational resources and time needed to achieve the final model rankings. This metric is essential for understanding the practicality of the adversarial evaluation methods, particularly in scenarios where resources may be constrained or when rapid evaluations are necessary.

4.2 Experiment Results

Ranking Consistency. We use the model ranking and score calculated by round-robin tournament as a baseline to compare the ranking and score differences of four other methods. From Table 3, we can observe the differences between the four methods with round-robin tournament.

For the Elo rating method, we calculate the Elo ratings using all 27,500 model pairs. It is worth noting that the ranking from the Elo rating method shows a considerable deviation from the baseline. This can be attributed to the Elo rating system's particular sensitivity to the sequence of matches: altering the order of games can result in different rankings. Additionally, the presence of annotation errors can lead to significant score fluctuations for the models, which in turn can cause considerable instability in the Elo-based rankings.

For the SuperCLUE method, we use GPT-3.5 as a benchmark for comparison and collected data from 5,000 samples (10 models × 500 samples). Compared to the baseline, this method exhibits the highest error, resulting in low accuracy of the model rankings.

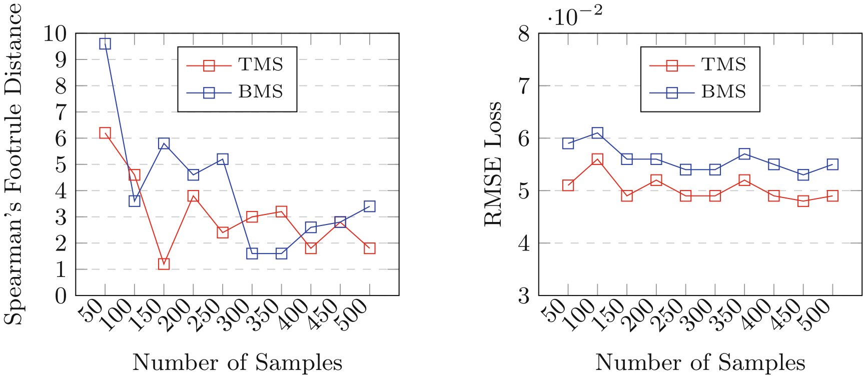


Fig. 4. Ranking results of increasing samples

In our experiments on BMS and TMS method, we use the 10 sets of initial model ranks $\{\mathcal{R}_1, ..., \mathcal{R}_{10}\}$ mentioned in the merge sort method. The Spearman's Footrule Distance and the RMSE are based on the average rankings and scores from 10 experiments. From the result metrics, the TMS method exhibits a more significant advantage. TMS method only uses 7,642 model pairs, which is about 27.8% of the baseline, with an average Spearman's Footrule Distance of only 1.8

Table 4. The Number of Annotations(The Proportion of Baseline)

Samples	Method	
	BMS	TMS
50	1132(41.2%)	775 (28.2%)
100	2271(41.3%)	1528(27.8%)
150	3431(41.6%)	2322(28.1%)
200	4558(41.4%)	3094(28.1%)
250	5701(41.5%)	3854(28.1%)
300	6799(41.2%)	4594(27.9%)
350	7897(41.0%)	5357(27.8%)
400	8998(40.9%)	6088(27.7%)
450	10142(41.0%)	6858(27.7%)
500	11249(40.9%)	7642(27.8%)

and a standard deviation of 1.4. In contrast, BMS method uses 11,249 model pairs, about 40.9% of the baseline, with an average Spearman's Footrule Distance of 3.4 and a standard deviation of 1.8. A primary reason for the improved performance of TMS is its ability to revise some incorrectly labeled data, thus reducing the proportion of mislabeled data. Overall, TMS performs better than BMS in terms of efficiency and accuracy.

Dynamic Evaluation. In the practical evaluation, the datasets often grow dynamically. In this section, we mainly investigate and compare the performance of two methods, BMS and TMS, in the context of dynamically increasing samples.

Figure 4 displays the average data from the experiments $\{\mathcal{R}_1, ..., \mathcal{R}_{10}\}$. With the dynamic growth of samples, both the BMS and TMS methods show a significant decrease in Spearman's Footrule Distance. When exceeding 400 samples, the TMS method gradually gains an advantage, narrowing the Spearman's Footrule Distance to within 2 compared to the round-robin tournament. Moreover, from the right diagram of Fig. 4, it can be seen that the RMSE of TMS is lower than that of BMS, and it maintains a stable value as the number of samples increases.

Table 4 demonstrates the average number of annotations for BMS and TMS along with their proportions relative to the round-robin tournament across ten sets of experiments with initial model ranks from $\{\mathcal{R}_1, ..., \mathcal{R}_{10}\}$. The annotation count for BMS approximates its time complexity $N \times O(L \times logL)$, where N is the number of samples and L is the number of models. As for TMS, incorporating transitivity, results in a lower number of annotations. As sample sizes increase, the BMS method's annotation count constitutes approximately 41% of $N \times L^2$, compared to only 28% for the TMS method, indicating that TMS achieves a more effective balance between efficiency and accuracy in ranking.

Stability Improvement. In stability experiments, we conduct tests on iter-BMS and iter-TMS methods across various batch sizes and perform 100 repeated experiments for each batch size. In each experiment, the initial rank of the models is randomly assigned. The results are summarized in Fig. 5. Analysis of the data in the figure, we can infer that as the batch size increases, the stability of the merge sort decreases progressively, reaching the maximum Spearman's Footrule Distance when the batch size is 512 (which corresponds to the original BMS or TMS method). These experimental findings suggest that the iter-BMS and iter-TMS can effectively enhance the stability of the sorting process when the initial rankings of the models are given randomly.

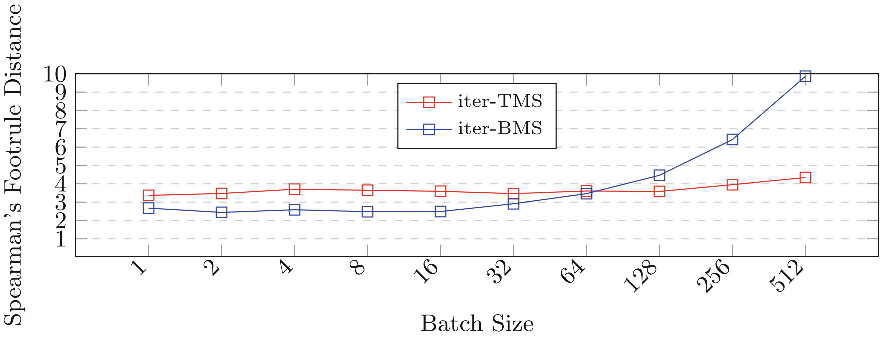


Fig. 5. Iterated merge sort's result with different batch size

5 Conclusion

In this paper, we study the ranking method in LLMs evaluation which aims to reduce the annotation costs and obtain an approximately ranking of models compared to the round-robin tournament. We propose the TMS method which utilizes the merge process in merge sort to construct the model pairs to get the relative order between them, meanwhile taking advantage of the transitivity of compare results. This approach proves to be more stable and efficient than other methods, such as Elo rating and SuperCLUE. The experiments show that when the initial rank of models is reasonable, TMS achieves the closest sorting results to round-robin tournament and requires only 27.8% model pairs of the round-robin tournament. In situations where the initial rank of models is unknown, the ranking result of merge sort vary significantly. To tackle this problem, we propose the iter-TMS that updates the model ranking after each batch of samples which improves the stability of the merge sort. When the initial model ranking is random and using a smaller batch size (less than 64), the iter-BMS and iter-TMS methods are capable of maintaining a high level of accuracy in ranking results.

Our method has demonstrated promising results in ranking LLMs, yet it faces challenges in handling situations with subpar annotation quality, which could significantly undermines the accuracy of rankings. In the future, we plan to explore this issue more thoroughly. Additionally, we will focus on examining our approach's scalability and developing strategies to adaptively adjust the batch size within iter-TMS, aiming to increase the overall flexibility of our method.

References

1. Berry, K.J., Mielke, P.W., Jr.: Spearman's footrule as a measure of agreement. Psychol. Rep. **80**(3), 839–846 (1997)
2. Bommasani, R., et al.: On the opportunities and risks of foundation models. arXiv preprint arXiv:2108.07258 (2021)
3. Chang, Y., et al.: A survey on evaluation of large language models. ACM Trans. Intell. Syst. Technol. **15**, 1–45 (2023)
4. Opencompass Contributors.: Opencompass: a universal evaluation platform for foundation models (2023). https://github.com/open-compass/opencompass
5. Elo, A.E.: The proposed USCF rating system, its development, theory, and applications. Chess Life **22**(8), 242–247 (1967)
6. Hendrycks, D., et al.: Measuring massive multitask language understanding. In: Proceedings of the International Conference on Learning Representations (ICLR) (2021)
7. Huang, Y., et al.: C-eval: a multi-level multi-discipline Chinese evaluation suite for foundation models. In: Advances in Neural Information Processing Systems, vol. 36 (2024)
8. Li, R., Patel, T., Du, X.: PRD: peer rank and discussion improve large language model based evaluations. arXiv preprint arXiv:2307.02762 (2023)
9. Liang, P., et al.: Holistic evaluation of language models. arXiv:2211.09110 (2023)
10. Lin, C.Y.: Rouge: a package for automatic evaluation of summaries. In: Text Summarization Branches Out, pp. 74–81 (2004)
11. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: Bleu: a method for automatic evaluation of machine translation. In: Proceedings of the 40th annual meeting of the Association for Computational Linguistics, pp. 311–318 (2002)
12. Big-Bench authors.: Beyond the imitation game: quantifying and extrapolating the capabilities of language models. Trans. Mach. Learn. Res. (2023)
13. Wang, P., et al.: Large language models are not fair evaluators. arXiv preprint arXiv:2305.17926 (2023)
14. Wei, J., et al.: Emergent abilities of large language models. arXiv preprint arXiv:2206.07682 (2022)
15. Wikipedia: Round-robin tournament (2023). https://en.wikipedia.org/wiki/Round-robin_tournament
16. Xu, C., et al.: WizardLM: empowering large language models to follow complex instructions. arXiv preprint arXiv:2304.12244 (2023)
17. Xu, L., et al.: Superclue: a comprehensive Chinese large language model benchmark. arXiv preprint arXiv:2307.15020 (2023)
18. Zhang, Y., et al.: Llmeval: a preliminary study on how to evaluate large language models. arXiv preprint arXiv:2312.07398 (2023)
19. Zheng, L., et al.: Judging LLM-as-a-judge with MT-bench and chatbot arena. In: Advances in Neural Information Processing Systems, vol. 36 (2024)
20. Zhong, W., et al.: AGIEval: a human-centric benchmark for evaluating foundation models. arXiv preprint arXiv:2304.06364 (2023)

Enhancing HVAC Control Efficiency: A Hybrid Approach Using Imitation and Reinforcement Learning

Kevlyn Kadamala(✉), Des Chambers, and Enda Barrett

University of Galway, Galway, Ireland
`{k.kadamala1,des.chambers,enda.barrett}@universityofgalway.ie`

Abstract. This paper explores the application of imitation learning (IL) and reinforcement learning (RL) in HVAC control. IL learns to perform tasks by imitating a demonstrator, utilising a dataset of demonstrations. However, the performance of IL is highly dependent on the quality of the expert demonstration data. On the other hand, RL can adapt control policies based on different objectives, but for larger problems, it can be sample inefficient, requiring significant time and resources for training. To overcome the limitations of both RL and IL, we propose a combined methodology where IL is used for pre-training and RL for fine-tuning. We introduce a fine-tuning methodology to HVAC control inspired by a robot navigation task. Using the 5-Zone residential building environment provided by Sinergym, we collect state-action pairs from interactions with the environment using a rule-based policy to create a dataset of expert demonstrations. Our experiments show that this combined methodology improves the efficiency and performance of the RL agent by 1% to 11.35% compared to existing literature. This study contributes to the ongoing discourse on how imitation learning can enhance the performance of reinforcement learning in building control systems.

Keywords: Imitation learning · Reinforcement learning · Continuous HVAC control

1 Introduction

Building operations account for 30% of global final energy consumption[1], with HVAC systems typically accounting for 40% of total building energy consumption[2]. Occupant comfort level satisfaction relies heavily on the effective functioning of heating, ventilation, and air conditioning (HVAC) systems. Incorrect implementation, however, may result in excessive energy consumption, escalating costs, and reduced occupant satisfaction. Traditional building controls primarily

---

[1] https://www.iea.org/energy-system/buildings.

[2] https://www.environment.gov.au/system/files/energy/files/hvac-factsheet-energy-breakdown.pdf.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
A. Bifet et al. (Eds.): ECML PKDD 2024, LNAI 14949, pp. 256–270, 2024.
https://doi.org/10.1007/978-3-031-70378-2_16

rely on rules and heuristics derived from expert experience. These rule-based controllers (RBCs) usually rely on pre-determined set points, which may not be optimal as they are not customised to the building specifics and local weather conditions. Recently, there has been an increase in solutions that leverage smart thermostats, replacing manual control configurations. New methodologies and algorithms like model predictive control (MPC) [1] and reinforcement learning (RL) [15] can adapt their control policies based on different objectives or cost functions. However, both systems have certain drawbacks. MPC systems find it challenging to deal with non-linearity in building dynamics caused by the complex nature of building systems, long-horizon predictions for accurately forecasting building system behaviour over extended periods, uncertainties in occupancy patterns and external factors like weather conditions. On the other hand, the limitation of RL lies in its sample inefficiency, requiring a significant investment in time and training resources to reach a desirable level of performance. In this work, we intend to address the challenges RL faces in HVAC control. More specifically, we aim to see whether an imitation learning approach can enhance the training speed and overall performance of the algorithm.

Imitation learning (IL) involves learning to perform a task by observing and imitating the behaviour of a demonstrator. Instead of relying on explicit programming seen in MPC or reward signals used in RL, this method utilises a dataset of demonstrations that consist of input-output pairs representing actions taken in different states by the demonstrator. These demonstrations are usually collected from interactions between humans or expert systems with the environment. IL has, thus, seen applications in domains like autonomous vehicle driving [12], robotics [18], navigation tasks [19], etc. However, a drawback of IL is that the performance of the trained agent is highly dependent on the quality of the expert demonstration data. If the expert demonstrations are sub-optimal or incomplete, the learned policy may inherit these limitations and fail to generalise well in novel situations. Thus, we propose utilising IL and RL as pre-training and fine-tuning methodologies, respectively. Policies obtained from pre-training with IL will provide a foundation for RL fine-tuning, making optimisation easier than learning from scratch.

In this study, we ask the question - how can imitation learning help improve the performance of reinforcement learning in the application of building control systems? Imitation learning techniques, specifically behavioural cloning (BC), have been used in HVAC control before [7]. However, after pre-training with BC, we obtain a trained actor and an untrained critic for RL fine-tuning. Having such a combination interact during fine-tuning can lead to a drop in performance. Thus, we introduce a fine-tuning methodology to HVAC control that was inspired by a robot navigation task [19]. For our experiments, we use the 5-Zone residential building environment[3] provided by Sinergym [10]. To create a dataset of expert demonstrations, we utilise a rule-based policy and collect the state-action pairs from interactions with the environment. Figure 1 illustrates an overview of our training methodology. Our experiments show that the combined

---

[3] https://ugr-sail.github.io/sinergym/compilation/main/pages/buildings.html#zone.

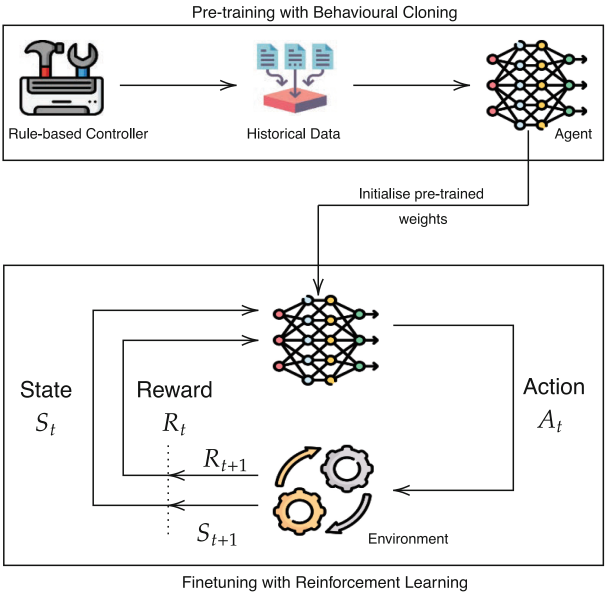

Pre-training with Behavioural Cloning
Rule-based Controller
Historical Data
Agent
Initialise pre-trained weights
State S_t
Reward R_t
R_{t+1}
S_{t+1}
Action A_t
Environment
Finetuning with Reinforcement Learning

Fig. 1. Overview of the pre-training and fine-tuning process.

methodology improves the efficiency and performance of the RL agent by 1% to 11.35% when compared to prior work [7], naive fine-tuning and training from scratch.

2 Previous Work

Reinforcement learning (RL) is gaining traction as a valuable method for Heating, Ventilation, and Air Conditioning (HVAC) systems due to its ability to learn optimal control policies for improving the management of complex and dynamic environments, such as buildings.

Previous research in HVAC systems has often utilised tabular methods in RL [2,14,24]. However, progress in the fields of deep learning and RL, has led to methods that blend both, using deep neural networks to improve reinforcement learning (DRL) algorithms. With the help of the Deep Q-Networks (DQN) algorithm, Wei et al. [23] were able to reduce costs by 20% to 70% when compared to scheduling methods. Similarly, when using DQN to control space heating and domestic hot water temperature, Lissa et al. [15] saw up to 16% energy savings. Arroyo et al. [1] combine model predictive control (MPC) with reinforcement learning (RL) to create RL-MPC, which aims to find the best policy while meeting all constraints. They show that RL-MPC outperforms basic RL in constraint satisfaction but can achieve similar results to pure MPC with a state estimator and optimiser. Further analysis of deep learning algorithms has been conducted by Biemann et al. [4], where they evaluated different model-free RL algorithms for continuous HVAC control. However, due to the considerable time and data required for RL to learn effective policies, researchers have incorporated transfer learning techniques to address this challenge. Transfer learning involves taking an established or trained policy from the source domain and adapting it to a target domain by leveraging past knowledge. In the context of building control systems, Lissa et al. [16] transfer HVAC agents with different spatial and geographical characteristics using tabular Q-learning, reporting that with transfer learning, temperature comfort violations were brought down to only 3% of the day, compared to 7% to 36% without transfer learning. Extending this work into a deep learning setup, Kadamala et al. [11] show that their heterogeneous transfer learning methodology adapts to buildings that differ in climate and/or characteristics, showcasing improvements from 1% to 4% compared to agents trained from scratch. Chen et al. [5] created Gnu-RL, incorporating a differentiable MPC. Initially pre-trained with imitation learning on historical data, it refines its policy using the PPO algorithm. Their research demonstrated a 6.6% energy reduction in simulations and a 16.7% decrease in cooling demand in a real-world conference room over three weeks, surpassing the existing controller while effectively managing temperature settings. Liu et al. [17] integrate RL with a rule-based control policy by adding a behavioural cloning loss (Eq. 3.1) to the policy update step to penalise the policy that differs too much from the behavioural policy. The proposed approach demonstrates significant performance improvements in building HVAC control tasks, notably where rule-based control methods are prevalent and robust. Coraci et al. [6] performed online transfer learning (OTL) with the help of imitation learning. Here, the pre-trained agent is transferred to the target controller, but it does not operate during the imitation learning phase; instead, the memory buffer of the OTL agent is initialized with transitions from the rule-based controller. This proved to be effective for enhancing the OTL agent during the initial days of development.

In our work, we perform imitation learning for pre-training, after which we fine-tune the agent with PPO. Similar work has been done by Dey et al. [7], where they generated a large dataset with roughly four years' worth of artificial states and discrete action data from a rule-based controller. They pre-train using imitation learning on this data and then fine-tune with PPO. However, in our work, we take inspiration from Ramrakhya et al. [19]. We perform behavioural cloning as a pre-training strategy and adopt the fine-tune methodology proposed via the critic learning and interactive learning phases, showing that the combined strategy can outperform naive fine-tuning while only requiring a single year's worth of data for pre-training.

3 Methodology

3.1 Behavioural Cloning from Demonstrations

Behavioural cloning (BC) uses supervised learning to learn a policy π from a dataset of state-action pairs $\zeta \in D$. It attempts to minimise the difference between the learned policy and expert demonstrations with respect to a defined metric or cost function L. Thus, the optimisation problem for BC can be defined as:

$$\hat{\pi}^* = \arg\min_{\pi} \sum_{\zeta \in D} \sum_{x \in \zeta} L(\pi(x), \pi^*(x)) \tag{1}$$

where $\hat{\pi}^*$ is the approximated policy and $\pi^*(x)$ is the expert action at state x.

Historical data is required to pre-train an agent using BC. For this, we generated an artificial dataset using the actions from the RBC defined in Sect. 4.3. A dataset amounting to a single year's worth of data was generated. Using this data, we then train the actor using BC. Here, the state consists of observation data from the environment consisting of building and weather information, and actions consist of the setpoint values set by the RBC (see Table 2). In our work, we implement the negative log-likelihood loss function given in Eq. 2

$$\mathcal{L}(\theta) = -\frac{1}{N} \sum_{t=1}^{N} (\log \pi_\theta(a_t|s_t) + \lambda H(\pi_\theta)) \tag{2}$$

where $\pi_\theta(a_t|s_t)$ is the predicted probability of action a_t given state s_t, $H(\pi_\theta)$ is the entropy of the policy distribution π_θ and λ is the weight given to the entropy term for regularisation. λ is set to 0.001 based on the Imitation library [8]. We perform an 80% - 20% split of the training and testing data to evaluate the performance of the trained actor. For testing, we use mean absolute error (MAE) to measure the absolute difference between predicted and actual setpoint temperatures. Table 1 summarises the training and testing losses for the three environments. Each environment was simulated thrice with different seeds.

Table 1. Training and Testing Losses.

Environment	Training Loss	Testing Loss
Hot	–0.519 ± 0.002	8.624e–5 ± 5.36e–5
Mixed	–0.52 ± 0.003	1.065e–4 ± 6.31e–5
Cool	–0.519 ± 0.003	8.325e–5 ± 3.72e–5

However, BC has a few disadvantages. Firstly, the performance of the policy learnt heavily depends on the quality of samples provided by the expert dataset. Additionally, as the expert policy π^* determines the distribution of the sampled states x, the learnt policy π will perform poorly on unseen states. Thus, BC often learns a policy that generalises poorly. This work uses BC only as pre-training to provide good neural network weight initialisation. The pre-trained agent is then fine-tuned with RL, which helps mitigate BC issues and makes the agent more robust and generalisable.

3.2 Reinforcement Learning Fine-Tuning

Reinforcement learning (RL) involves a set of states (S) and control actions (A), where the system dynamics are defined by a probabilistic transition model, denoted as $p(s_{t+1} = s' | s_t = s, a_t = a)$, representing the likelihood of transitioning from state s to state s' by taking action a at time t. Additionally, RL incorporates a reward function $r_t = R(s_t, a_t)$ that provides a reward r_t at timestep t. The goal of an RL agent is to learn a policy π that maximizes its cumulative reward.

Our work uses the Proximal Policy Optimisation (PPO) algorithm for fine-tuning with RL [20]. PPO is a policy gradient algorithm that optimises a parameterised policy using the gradients of the expected return with gradient ascent. It effectively mitigates performance collapse by introducing a clipped surrogate objective function to control policy updates within a specified range, simplifying the optimization process compared to other more complex algorithms. The objective function of the PPO algorithm is defined as:

$$L^{CLIP}(\theta) = \hat{\mathbb{E}}_t[\min(r_t(\theta), \hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t)] \tag{3}$$

where, $r_t(\theta)$ is the probability ratio $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t, s_t)}$ and $\hat{A}_t$ is the advantage function. PPO is implemented using the Actor-Critic model [13] where the actor has to maximise $L^{CLIP}(\theta)$ and an entropy bonus given as S, while the critic has to minimise the value function error term $L_t^{VF}(\theta)$. Hence, the overall objective function to be maximised can be defined as:

$$L_t^{CLIP+VF+S}(\theta) = \hat{\mathbb{E}}[L_t^{CLIP}(\theta) - c_1 L_t^{VF}(\theta) + c_2 S[\pi_\theta](s_t)] \tag{4}$$

where, c_1, c_2 are coefficients and $L_t^{VF}(\theta)$ is the squared-error loss between the estimated value function and the target value given as $(V_\theta(s_t) - V_t^{targ})^2$.

To perform fine-tuning with PPO, we initialise the actor with the policy weights that were trained with the help of Behavioural Cloning. However, the critic is initialised with random weights. As a result, we would end up with a trained actor and an untrained critic. Thus, inspired by the methodology described by [19], we first train the critic during RL. To do this, we divide the RL training into two phases. Figure 2 describes the learning rate schedules for the best-performing agent.

Critic Learning Phase. During this phase, we collect interactions from the environment using the trained actor to train the critic. In this phase, the actor can be completely frozen (as described in [19]); however, from experiments, we find that gradually increasing the actor learning rate from 0.0 at the start to 1.5×10^{-5} at the 700,000th timestep provides the best results. During this time, the critic is initialised with a relatively high learning rate, which is decayed as training progresses. In our experiments, we find that initialising the critic with a learning rate of 1.0×10^{-3} gave the best results. We maintain this high learning rate for the first 150,000 timesteps.

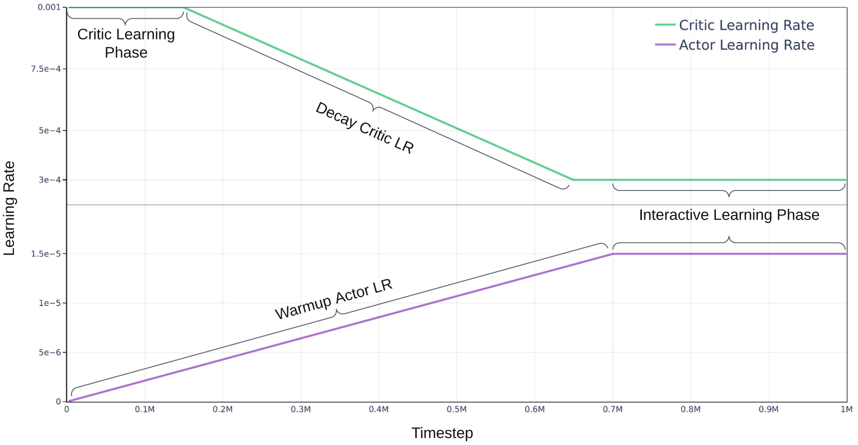


Fig. 2. Actor and Critic learning rate schedules during fine-tuning.

Interactive Learning Phase. After the actor and critic learning rates completely warm up and decay, they are stabilised for the remainder of the training process. We can keep the learning rates during this phase at the same or at different stabilisation values. Through our experiments, we find that keeping the actor and critic learning rates stable at different values yields the best results. For the actor, we stabilise the learning rate at 1.5×10^{-5}, which remains constant from the 700,000th timestep until the end of training. For the critic, we gradually decay the learning rate from 1.0×10^{-3} to 3.0×10^{-4} until the 650,000th timestep, which remains constant until completion.

4 Experimental Setup

4.1 Environment

Our experiments are simulated on the `5ZoneAutoDXVAV` environment provided by the Python library known as Sinergym [10][4] (v3.1.7). The `5ZoneAutoDXVAV` is a single-storey building divided into one indoor and four outdoor zones[5]. The state space and action space for the environment are given in Table 2. We simulate three different weather conditions in our experiments using EnergyPlus. The hot weather data is from Davis-Monthan AFB, Arizona, USA; the mixed weather data is from New York City, New York, USA, and the cool weather data is from Port Angeles, Washington, USA. All environments are initialised with stochasticity in weather. This stochasticity is introduced with the Ornstein-Uhlenbeck [3] process where σ, μ and τ are 1.0, 0.0 and 0.001, respectively. Each simulated episode spans over a duration of one year. Within each episode, 35,040 intervals exist, lasting fifteen minutes each.

---

[4] https://ugr-sail.github.io/sinergym/compilation/main/index.html.

[5] https://ugr-sail.github.io/sinergym/compilation/main/pages/buildings.html#zone.

Table 2. Environment Summary.

	Variable Names	Number
State Space	Site outdoor air dry bulb temperature, site outdoor air relative humidity, site wind speed, site wind direction, site diffuse solar radiation rate per area, site direct solar radiation rate per area, zone thermostat heating setpoint temperature, zone thermostat cooling setpoint temperature, zone air temperature, zone thermal comfort mean radiant temperature, zone air relative humidity, zone thermal comfort clothing value, zone thermal comfort Fanger model PPD, zone people occupant count, people air temperature, facility total HVAC electricity demand rate, hour, day and month	19
Action Space	Heating setpoint and Cooling setpoint	2

4.2 Rewards

The goal of the Deep RL agent is to reduce energy usage while maintaining a comfortable temperature range. This is achieved through an objective function that combines the weighted sum of energy consumption and thermal discomfort, which are normalised. Equation 5 describes the reward function.

$$R = -\omega \times \lambda_P \times P_t - (1 - \omega) \times \lambda_T \times \exp(|T_t - T_{upper}| + |T_t - T_{lower}|) \quad (5)$$

where, P_t is the power consumption and T_t is the current indoor temperature. ω represents the weight assigned to power consumption and thus $(1 - \omega)$ is the weight assigned to comfort. λ_P and λ_T are scaling constants for power consumption and comfort penalties, respectively. T_{upper} and T_{lower} define the upper and lower limits of the comfort temperature range. Discomfort is determined by calculating the absolute difference between the current temperature and the comfort range. If the temperature falls within the comfort range, the discomfort value is zero. Along with rewards, we also analyse two other Key Performance Indicators (KPIs):

- Comfort Violation Time (%): Percentage of time that the temperature has been beyond the bounds of the user comfort temperature ranges.
- Mean Power: Average power consumption per step in the episode.

4.3 Training Setup

In our work, we implement Behavioural Cloning with the help of the Imitation library [8] and the PPO algorithm with the help of CleanRL [9]. Table 3 represents the default hyperparameters for the PPO algorithm included in CleanRL. Following CleanRL, the actor and the critic networks are built using separate neural networks. Thus, there are no shared weights. The neural network architecture for both the actor and the critic consists of a single hidden layer of size 64.

Table 3. PPO Hyperparameters.

Hyperparameter Name	Value
Total Timesteps	1,000,000
Learning Rate	3.0e–4
Number of Steps per Policy Rollout	2048
Anneal Learning Rate	True
Gamma	0.99
General Advantage Estimation	0.95
Minibatch Size	32
Update Epochs	10
Advantage Normalisation	True
Surrogate Clipping Coefficient	0.2
Clip loss for Value Function	True
Entropy Coefficient	0.0
Value Function Coefficient	0.5
Maximum Norm for Gradient Clipping	0.5

The layer initialisation included in CleanRL was skipped. The Tanh activation function is used after every layer except the output layer. The performance of the learned agents is compared to the Rule-Based Controllers (RBC) provided by Sinergym[6]. The rules for the RBC are defined in Algorithm 1. During the training process, we periodically evaluate the performance of the agents in the same environment but with a different seed. We evaluate its performance for the entire year, i.e. one episode. For testing, we follow a similar procedure to the evaluation, where we test the agent in the same environment but with a different seed and monitor its performance for a total of five years, i.e. five episodes. Our

Algorithm 1 Rule for RB Controller for 5Zone

```
summer_setpoint ← $(22.5, 26.0)$
winter_setpoint ← $(20.0, 23.5)$
summer_range ← (1 June, 30 September)
for each step in environment do
    if current_date is in summer_range then
        curr_setpoint ← summer_setpoint
    else
        curr_setpoint ← winter_setpoint
    end if
end for
```

---

[6] https://github.com/ugr-sail/sinergym/blob/main/sinergym/utils/controllers.py.

implementation source code can be accessed at GitHub - https://github.com/kad99kev/HVACIRL.

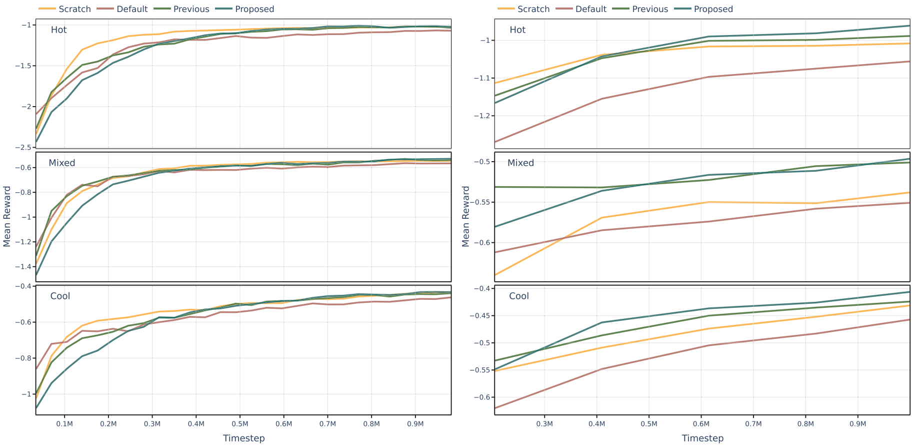
5 Results

In our experiments, we compare the proposed methodology with different agents. The RBC or **controller** agent is the rule-based controller defined in Sect. 4.3 (see Algorithm 1). The **imitation** agent is the behavioural cloning agent without fine-tuning. We compare this agent to highlight the performance of the pre-trained agent before we fine-tune it with RL. The **scratch** agent is an RL agent trained without any pre-training. It is trained with the default hyperparameters and training methodology of PPO (see Table 3). The **default** agent is a pre-trained agent that is fine-tuned naively. For this agent, we follow the same RL training methodology as the scratch agent; however, the weights of the actor are initialised with BC. Thus, fine-tuning is naive, where a pre-trained actor and an untrained critic interact with each other. Finally, we include the agent described in [7] (called **previous**), where the authors utilised a learning rate for the policy network that was one-hundredth of the learning rate of the baseline RL learning strategy for the initial ten episodes and one-tenth for subsequent training episodes. We include these agents to show how naive fine-tuning can affect the performance of an agent and, thus, the need for a different fine-tuning methodology. In the following sections, we refer to the agent trained using our proposed methodology as the **proposed** agent.

5.1 Performance Analysis

In this section, we compare the training, evaluation and testing performance of the different agents. Figure 3a shows that within 400,000 timesteps, our **proposed** agent begins outperforming all other RL agents. When we compare the performance on evaluation, Fig. 3b, shows that our **proposed** agent learns faster than the **default** agent. Additionally, our **proposed** agent consistently and significantly outperforms the **scratch** agent. This implies that pre-training the agent helps it generalise better. However, Fig. 3b also suggests that the fine-tuning methodology plays an important role. Naively fine-tuning a pre-trained agent with default learning rates can lead to poor policies being learnt, as it is evident that the **default** agent cannot outperform the **scratch** agent. Along with this, our **proposed** agent also outperforms the **previous** agent.

When comparing the rewards and KPIs (see Table 4) of the fine-tuned RL agents with an agent trained solely on the imitation learning dataset and the RBC, our **proposed** agent achieves the highest reward across all three weather environments during testing. In the 5-Zone hot weather environment, the **proposed** agent outperforms the **previous** agent by 2.83%, the **default** agent by 9.17% and the **scratch** agent by 5.04% while significantly outperforming the **controller** agent by 21.87%. While it may not be the best at saving power, the **proposed** agent has the least number of comfort violations among all agents.



(a) Mean reward during training.

(b) Mean reward during evaluation.

Fig. 3. Learning curves of the different agents during training and evaluation on hot, mixed and cool weather.

Table 4. Rewards and KPIs average performance summary during testing.

Environment	Agent	Rewards (higher better)	Power Consumption (lower better)	Comfort Violations (%) (lower better)
5-Zone Hot	Controller	–1.230	5939.19	41.82
	Imitation	–2.936	**5623.01**	50.80
	Scratch	–1.012	6834.64	31.23
	Default	–1.058	6614.23	33.48
	Previous	–0.989	6942.00	30.47
	Proposed	**–0.961**	6836.64	**28.81**
5-Zone Mixed	Controller	–0.619	**5725.47**	45.45
	Imitation	–1.793	5991.53	48.29
	Scratch	–0.538	7259.21	25.73
	Default	–0.550	7131.13	28.06
	Previous	–0.501	7027.80	23.48
	Proposed	**–0.496**	6917.05	**22.72**
5-Zone Cool	Controller	–0.455	**3751.00**	29.24
	Imitation	–1.385	4033.08	43.42
	Scratch	–0.429	5548.59	20.56
	Default	–0.458	5260.54	22.34
	Previous	–0.424	5314.49	20.09
	Proposed	**–0.406**	5021.88	**19.07**

As the table shows, the **imitation** and **controller** agents are the best at saving power; however, they have the highest percentage of comfort violations. The **proposed** agent, however, can balance this by not overly spending power yet significantly reducing comfort violations. Similar behaviour is observed in the 5-Zone mixed weather environment as well. The **proposed** agent demonstrates significant enhancements in overall rewards, with improvements of 1.00%, 9.82%, 7.81%, and 19.87% compared to **previous**, **default**, **scratch**, and **controller** agents, respectively. Again, we observe that the **imitation** and **controller** agents are the best at saving power but are the poorest at maintaining comfort. The **proposed** agent stands out for its efficient power utilisation and minimal comfort violations among all RL agents evaluated. Likewise, the **proposed** agent performs best in the 5-Zone cool weather environment. Compared to **previous**, **default**, **scratch** and **controller** agents, it sees a 4.25%, 11.35%, 5.36% and 10.77% improvement in the overall rewards, respectively. Here, the performance of the **default** agent is comparable to the **controller** agent. The **proposed** agent achieves a 5.51% decrease in power consumption compared to the **previous** agent, resulting in a 1.02% reduction in comfort violations. Similar to the mixed weather environment, the **proposed** agent performs best in both power consumption and comfort violations when compared to the other RL agents.

5.2 Policy Analysis

This section compares the policies adopted by the different agents in our experiments. We analyse their heating and cooling setpoint temperatures and their effect on the indoor temperature. Hourly observed temperatures show the average temperatures at each hour throughout the year, while monthly observed temperatures show the average temperature for each month.

From Fig. 4, we can see that the **proposed** agent and the **previous** agent follow similar policies; however, the **proposed** agent maintains a slightly higher cooling and heating setpoint on average. The **imitation** agent follows the RBC policy very closely. In all three environments, we observe that the **scratch** agent drastically changes its setpoint temperatures as the outdoor temperature becomes warmer during the day. We can also see that the RL agents consistently maintain indoor temperatures within the average hourly user comfort zone, while the **imitation** and **controller** agents periodically violate comfort.

When comparing the monthly setpoint temperatures, from Fig. 5, we see that the **imitation** agent fails to adapt its setpoint temperatures during summer

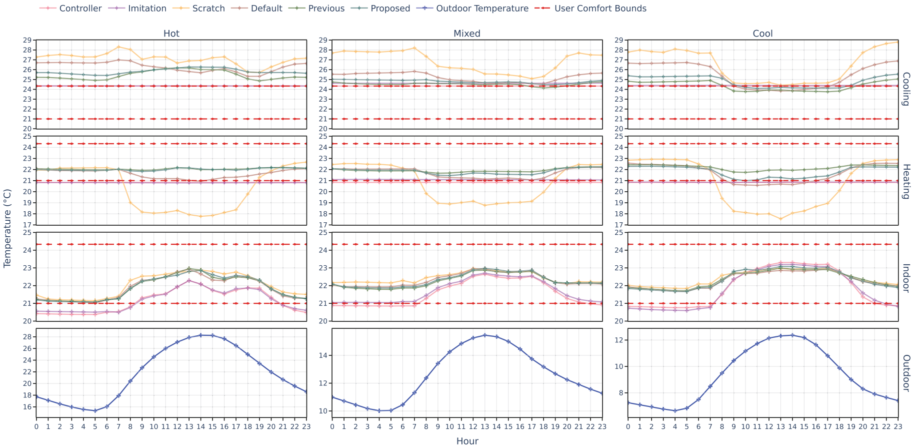


Fig. 4. Setpoint, indoor and outdoor temperatures summarised by the hour.

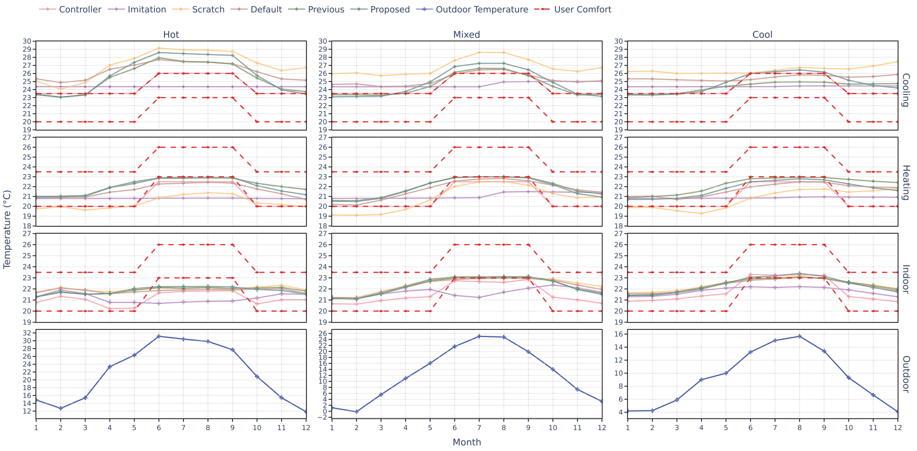


Fig. 5. Setpoint, indoor and outdoor temperatures summarised by the month.

when the temperatures increase. The graphs highlight how RL agents adapt their policies as user comfort and season change. We can see that the cooling setpoints are higher in hot weather, which decreases as we move to mixed weather, with it being the lowest in cool weather. The heating setpoints are relatively the same for hot and mixed weather but are slightly lower for cool weather. All agents struggle to maintain comfort in hot weather; however, the **proposed** agent does the best to maintain indoor temperatures. For other weather conditions, the RL agents are better able to maintain comfortable indoor temperatures. The RL agents are much better at keeping indoor temperatures closer to the user's comfort zone than the **controller** agent, which does well for mixed and cool weather. The **imitation** agent, however, does poorly throughout. This proves the need for RL fine-tuning, as the RL agents are able to adapt their setpoint temperatures well to maintain indoor comfort.

6 Conclusion and Future Work

Imitation learning helps provide a foundation for the RL agent during training. However, naively training the trained actor and untrained critic together can lead to worse policies being learnt than training from scratch. Thus, to avoid this, we can either freeze the actor or initialise it with a very low learning rate while the critic learns from interactions with the environment. From our experiments, we see that following this methodology results in agents outperforming not only the RBC but also naively fine-tuned agents and agents trained from scratch across all three weather environments. We also build upon prior work, showing that a better learning rate tuning strategy is able to outperform their agent.

The policies learnt by the proposed agent can better maintain indoor temperatures within the user's comfort bounds than all other agents. While this comes at the expense of more power, the least power-consuming agents are RBC and imitation. These two agents are the worst at respecting the user's comfort zone. Thus, the proposed agent is efficient at striking a balance between power consumption and user comfort.

For future work, a hyperparameter study could be conducted to analyse the effect of different hyperparameters on this approach. Additionally, historical data could be utilised to train an autoencoder, thus reducing the dimensionality of the observation space for effective representation learning. Finally, a multi-objective approach to this method can be considered. By separating the energy and comfort variables, further analysis can be conducted on different policies learnt to prioritise comfort and/or energy.

Acknowledgments. This work was conducted with the financial support of the Science Foundation Ireland Centre for Research Training in Artificial Intelligence under Grant No. 18/CRT/6223.

Ethical Impact. We do not foresee any ethical implications for our work. Our experiments do not involve personal data; all data and simulations used are publicly available on GitHub – https://github.com/ugr-sail/sinergym.

References

1. Arroyo, J., Manna, C., Spiessens, F., Helsen, L.: Reinforced model predictive control (rl-mpc) for building energy management. Appl. Energy **309**, 118346 (2022)
2. Barrett, E., Linder, S.: Autonomous hvac control, a reinforcement learning approach. In: Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2015, Porto, Portugal, September 7–11, 2015, Proceedings, Part III 15, pp. 3–19. Springer (2015). https://doi.org/10.1007/978-3-319-23461-8-1
3. Benth, F.E., Šaltytė-Benth, J.: Stochastic modelling of temperature variations with a view towards weather derivatives. Appl. Math. Finance **12**(1), 53–85 (2005)
4. Biemann, M., Scheller, F., Liu, X., Huang, L.: Experimental evaluation of model-free reinforcement learning algorithms for continuous hvac control. Appl. Energy **298**, 117164 (2021)
5. Chen, B., Cai, Z., Bergés, M.: Gnu-rl: a precocial reinforcement learning solution for building hvac control using a differentiable mpc policy. In: Proceedings of the 6th ACM international conference on systems for energy-efficient buildings, cities, and transportation, pp. 316–325 (2019)
6. Coraci, D., Brandi, S., Hong, T., Capozzoli, A.: Online transfer learning strategy for enhancing the scalability and deployment of deep reinforcement learning control in smart buildings. Appl. Energy **333**, 120598 (2023)
7. Dey, S., Marzullo, T., Zhang, X., Henze, G.: Reinforcement learning building control approach harnessing imitation learning. Energy AI **14**, 100255 (2023)
8. Gleave, A., et al.: Imitation: Clean imitation learning implementations. arXiv:2211.11972v1 [cs.LG] (2022). https://arxiv.org/abs/2211.11972
9. Huang, S., et al.: Cleanrl: high-quality single-file implementations of deep reinforcement learning algorithms. J. Mach. Learn. Res. **23**(274), 1–18 (2022). http://jmlr.org/papers/v23/21-1342.html

10. Jiménez-Raboso, J., Campoy-Nieves, A., Manjavacas-Lucas, A., Gómez-Romero, J., Molina-Solana, M.: Sinergym: a building simulation and control framework for training reinforcement learning agents. In: Proceedings of the 8th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation, pp. 319–323 (2021)
11. Kadamala, K., Chambers, D., Barrett, E.: Enhancing hvac control systems through transfer learning with deep reinforcement learning agents. Smart Energy, p. 100131 (2024)
12. Kebria, P.M., Khosravi, A., Salaken, S.M., Nahavandi, S.: Deep imitation learning for autonomous vehicles based on convolutional neural networks. IEEE/CAA J. Automatica Sinica **7**(1), 82–95 (2019)
13. Konda, V., Tsitsiklis, J.: Actor-critic algorithms. Advances in neural information processing systems **12** (1999)
14. Li, B., Xia, L.: A multi-grid reinforcement learning method for energy conservation and comfort of hvac in buildings. In: 2015 IEEE International Conference on Automation Science and Engineering (CASE), pp. 444–449. IEEE (2015)
15. Lissa, P., Deane, C., Schukat, M., Seri, F., Keane, M., Barrett, E.: Deep reinforcement learning for home energy management system control. Energy AI **3**, 100043 (2021)
16. Lissa, P., Schukat, M., Barrett, E.: Transfer learning applied to reinforcement learning-based hvac control. SN Comput. Sci. **1**(3), 1–12 (2020)
17. Liu, H.Y., Balaji, B., Gupta, R., Hong, D.: Rule-based policy regularization for reinforcement learning-based building control. In: Proceedings of the 14th ACM International Conference on Future Energy Systems, pp. 242–265 (2023)
18. Osa, T., Sugita, N., Mitsuishi, M.: Online trajectory planning and force control for automation of surgical tasks. IEEE Trans. Autom. Sci. Eng. **15**(2), 675–691 (2017)
19. Ramrakhya, R., Batra, D., Wijmans, E., Das, A.: Pirlnav: pretraining with imitation and rl finetuning for objectnav. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 17896–17906 (2023)
20. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347 (2017)
21. Taylor, M.E., Stone, P.: Transfer learning for reinforcement learning domains: a survey. J. Mach. Learn. Res. **10**(7) (2009)
22. Vázquez-Canteli, J.R., Nagy, Z.: Reinforcement learning for demand response: a review of algorithms and modeling techniques. Appl. Energy **235**, 1072–1089 (2019)
23. Wei, T., Wang, Y., Zhu, Q.: Deep reinforcement learning for building HVAC control. In: Proceedings of the 54th Annual Design Automation Conference 2017, pp. 1–6 (2017)
24. Yu, Z., Dexter, A.: Online tuning of a supervisory fuzzy controller for low-energy building system using reinforcement learning. Control. Eng. Pract. **18**(5), 532–539 (2010)
25. Zhu, Z., Lin, K., Jain, A.K., Zhou, J.: Transfer learning in deep reinforcement learning: a survey. IEEE Trans. Pattern Anal. Mach. Intell. (2023)

Synthesis of Standard 12-Lead ECG from Single-Lead ECG Using Shifted Diffusion Models

Jingwei Liu[1], Hongyan Li[1,2,3], and Shenda Hong[4](✉)

[1] School of Intelligence Science and Technology, Peking University, Beijing, China
leehy@pku.edu.cn
[2] Key Laboratory of Machine Perception, Peking University, Beijing, China
[3] National Key Laboratory of General Artificial Intelligence, Peking University, Beijing, China
[4] Institute of Medical Technology, Health Science Center of Peking University, Beijing, China
hongshenda@pku.edu.cn

Abstract. As the primary tool for monitoring cardiac health, a standard 12-lead ECG device is specialized medical equipment that is challenging to integrate into daily life. Meanwhile, existing portable ECG monitoring devices can only capture single-lead ECG, which is insufficient for health diagnosis. To address this issue, we propose a novel shifted diffusion model algorithm that utilizes a single-lead ECG to generate a standard 12-lead ECG. Our algorithm uses the detected single-lead ECG as the condition and employs the diffusion model to synthesize corresponding other 11-lead ECG. The extra shift is utilized in the forward process so that the model can learn better. Our approach has been tested on three datasets, yielding promising results.

Keywords: 12-lead ECG synthesis · Diffusion models · Signal process

1 Introduction

Cardiovascular diseases pose the greatest threat to human life, and long-term heart health monitoring is one of the primary ways to mitigate cardiovascular disease risks [1,2]. Currently, the most important method for monitoring heart health is the electrocardiograms (ECG) [3–5]. Depending on the number of leads used, ECG exhibits various changes, and the 12-lead ECG is the most commonly used in global healthcare centers [6,7]. The standard 12-lead ECG provides clinical information about heartbeats from multiple fixed viewpoints, and healthcare professionals assess the heart's health status by observing the waveforms of the 12 leads collectively. However, collecting 12-lead ECG data beyond clinical environments poses a formidable challenge [8]. Modern wearable devices can collect ECG using fewer leads than the standard 12 leads [10,11], but the limited lead

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
A. Bifet et al. (Eds.): ECML PKDD 2024, LNAI 14949, pp. 271–286, 2024.
https://doi.org/10.1007/978-3-031-70378-2_17

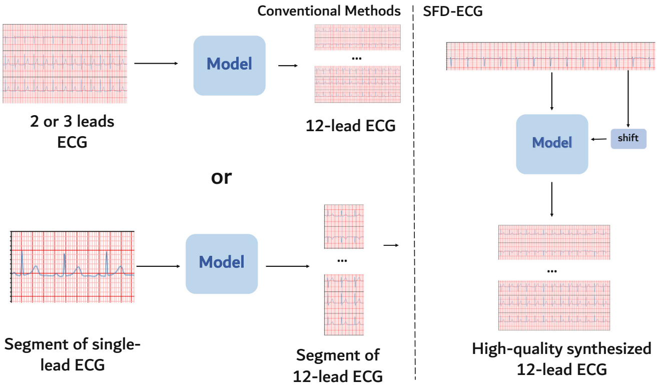


Fig. 1. The comparison between our method and existing approaches. We aim to employ an efficient and novel diffusion model framework to address the task of synthesizing standard 12-lead ECG from single-lead ECG.

data can pose additional challenges for medical professionals and traditional ECG analysis software interpretation [9].

To address this issue, some methods improve hardware design by placing enhanced single-lead devices at different positions on the body to obtain 12-lead ECG [12,13]. Their effectiveness heavily depends on the operator's experience, as inaccurate placement may introduce unwanted artifacts in the signal, leading to potential misdiagnosis [14]. Other studies aim to synthesize standard 12-lead ECG from collected 2 3 leads [15–17]. However, multiple electrodes still need to be utilized to ensure prediction accuracy, hindering the practical application. In recent years, some research has focused on the synthesis of 12-lead ECG from single-lead ECG (for clarity, we'll denote this task as '1–12 ECG synthesis') [18,19]. Due to limitations in model performance, these approaches can only synthesize precise ECG segments of 1 to 2.5 s in duration. Employing concatenation to obtain usable ECG signals results in inconsistencies in the finer details of the outcomes. In other words, viewing the 1–12 ECG synthesis task as conditional generation tasks, it's challenging for existing methods to learn the precise conditional distribution of long 12-lead ECG segments (Fig. 1).

Inspired by the recent success of diffusion models in conditional generation tasks, we have designed a novel time-frequency domain Shifted Diffusion model for 1–12 ECG synthesis (SD-ECG). However, applying conventional diffusion model frameworks yields unsatisfactory results. The sampling process of diffusion models originates from random noises, leading to high diversity in the outcomes, which conflicts with the requirements of the task. To address this issue, in our proposed diffusion model, the forward process gradually maps the target sequence to a prior distribution related to the single-lead ECG, rather than the commonly used Gaussian distribution in traditional diffusion models. Specifically, we introduce an observation-related shift in the forward process during training and consider it as the starting point for the reverse process. We introduce this point to optimize the starting point and trajectory of the diffusion process, facilitating the model to learn the accurate distribution of a complete 12-lead ECG. Furthermore, ECG signals exhibit frequency consistency across different leads, and to leverage this, our method synthesizes 12-lead ECG in the time-frequency domain.

Our contributions are as follows:

- Our method is the first to utilize diffusion models for the synthesis of 12-lead ECG from single-lead ECG, providing a novel perspective for future related research.
- We improve existing diffusion models for ECG synthesis tasks from two angles, providing meaningful references for similar tasks.
- We conduct experiments on popular ECG datasets and provide a comprehensive presentation and analysis of the results using four metrics. The experimental results demonstrate that our approach achieves comparable or better results compared to baselines.

2 Related Work

2.1 Synthesize ECG Using Diffusion Models

Diffusion models have surpassed GANs in image generation tasks [21]. Besides, they have also achieved the best results on other tasks, including generation and prediction of time series [22–24]. Recently, many studies have focused on synthesizing ECG data with diffusion models. [25] embeds single-lead ECG into a two-dimensional space and uses the Improved Denoising Diffusion Probabilistic Model (Improved-DDPM) [26] to learn the distribution of data in the two-dimensional space. [27] utilizes a diffusion model framework with Structured State Space Models (S4 model) [28] as the main structure to generate different leads of ECG. Similarly, [29] proposes a diffusion-based model coupled with a state space augmented transformer, synthesizing 12-lead ECG. These methods utilize diffusion models to generate standard 12-lead ECG from scratch, using the generated ECGs to augment existing datasets while protecting patient privacy. However, none of the existing approaches utilizes diffusion models for single-lead to 12-lead synthesis. We are the first to apply the conditional diffusion model to this task, achieving promising results.

2.2 Synthesis of 12-Lead ECG from Reduced Leads

Synthesizing standard 12-lead ECG from limited leads has attracted attention since 2004. Initial studies focus on synthesizing the remaining leads using 2–3 leads. [15] establishes relationships between three leads (D1, D2, V2) and the other six leads using MLP. Subsequent methods utilize similar network structures but with a change in lead requirements (leads I, II, V2) [16]. A model with convolutional neural network (CNN) and long short-term memory units (LSTM) is introduced for synthesizing two-lead to 12-lead ECG [17]. These methods do not fundamentally address the inconvenience caused by multi-lead ECG devices. The generative adversarial network (GAN) was utilized for producing all 12 leads of an ECG based on lead I [18]. The GAN architecture consists of a U-net as the generator and a patch discriminator as the discriminator. Using an encoder-decoder with CNN structure, ECG is synthesized from arbitrary lead perspectives [19]. Nonetheless, their approaches fail to generate sufficiently complete ECG signals, thereby constraining the practical application of these approaches.

3 Preliminary

3.1 Synthesis of 12-Lead ECG from Single-Lead ECG

Suppose we have an observed single-lead ECG signal $X = \{x_1^1, x_2^1, \cdots, x_n^1\}$, where n is the ECG length and X_i^0 is the observation at time step i. The Y is the target ECG leads $\{y_1^i, y_2^i, \cdots, y_n^i\}$ $(d^{'} \leq d)$, where the value range of i is from 2 to 12. The task of synthesizing 12-leads ECG is to learn a density $p_\theta(Y|X)$ that best approximates $p(Y|X)$, which can be written as:

$$\min_{p_\theta} D\left(p_\theta(Y|X)||p(Y|X)\right), \tag{1}$$

where θ denotes a parameters and D is some appropriate measure of distance between distributions. Given observation X the target 12-leads ECG can be obtained directly by sampling from $p_\theta(Y|X)$.

3.2 Diffusion Models

A diffusion model progressively destructs data $\boldsymbol{x}_0$ by injecting noise, then learns to reverse this process starting from $\boldsymbol{x}_T$ for sample generation. The forward process can be formulated as a Gaussian process with a Markovian structure:

$$\begin{aligned} q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) &:= \mathcal{N}(\boldsymbol{x}_t; \sqrt{1-\beta_t}\boldsymbol{x}_{t-1}, \beta_t \boldsymbol{I}), \\ q(\boldsymbol{x}_t|\boldsymbol{x}_0) &:= \mathcal{N}(\boldsymbol{x}_t; \sqrt{\overline{\alpha}_t}\boldsymbol{x}_0, (1-\overline{\alpha}_t)\boldsymbol{I}), \end{aligned} \tag{2}$$

where $\beta_1, \ldots, \beta_T$ denotes fixed variance schedule with $\alpha_t := 1 - \beta_t$ and $\overline{\alpha}_t := \prod_{s=1}^{t} \alpha_s$. This forward process progressively injects noise into data until all structures are lost, which is well-approximated by $\mathcal{N}(0, \boldsymbol{I})$. The reverse diffusion process learns a model $p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$ that approximates the true posterior:

$$p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t) := \mathcal{N}(\boldsymbol{x}_{t-1}; \mu_\theta(\boldsymbol{x}_t), \Sigma_\theta(\boldsymbol{x}_t)), \tag{3}$$

where μ_θ and Σ_θ are often computed by a UNet or a Transformer. Ho *et al.* [20] improve the diffusion training process and optimize following objective:

$$\mathcal{L}(\boldsymbol{x}_0) = \sum_{t=1}^{T} \mathop{\mathbb{E}}_{q(\boldsymbol{x}_t|\boldsymbol{x}_0)} ||\mu_\theta(\boldsymbol{x}_t, t) - \hat{\mu}(\boldsymbol{x}_t, \boldsymbol{x}_0)||^2, \tag{4}$$

where $\hat{\mu}(\boldsymbol{x}_t, \boldsymbol{x}_0)$ is the mean of the posterior $q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_0, \boldsymbol{x}_t)$ which is a closed from Gaussian, and $\mu_\theta(\boldsymbol{x}_t, t)$ is the predicted mean of $p_\theta(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t)$ computed by a neural network.

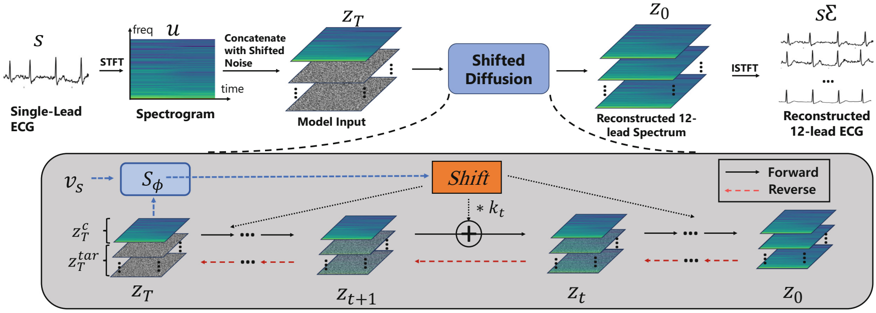


Fig. 2. Overview of SD-ECG. As depicted in the upper half of the figure, we transform the signal-lead ECG into the time-frequency domain using the Short-Time Fourier Transform (STFT), concatenate them with shifted noise, and then input them into the diffusion model for training. The lower half of the figure illustrates the diffusion process within the latent space. During the training process, we calculate the observed shifts and incorporate them into each step of the forward process.

4 Time-Frequency Shifted Diffusion for 12-Lead ECG Synthesis

In this section, we elucidate our proposed SD-ECG in detail, which adapts the temporal-spectral shifted diffusion model to synthesize 12-lead ECG as in Fig. 2. In Sect. 4.1, we introduce the motivation of our approach. The Sect. 4.2 delves into the design of the shifted diffusion model, providing a comprehensive overview of both the forward and reverse processes. Finally, we theoretically derive the optimization objective and summarize the overall pipeline of the proposed framework (Sect. 4.3).

4.1 Motivation

The diffusion model is currently the most advanced model for conditional generative tasks. Its multi-step interactions enable the model to iteratively interact with conditions, resulting in more precise generation outcomes. Conventional diffusion models involve constructing a Markov chain in the forward process, gradually transforming observed data into a predefined prior distribution, typically a Gaussian distribution. Subsequently, the sampling process involves drawing noise mappings from the prior distribution and feeding them into the reverse path of the Markov chain. While this method excels in unconditional generation tasks, it may not be optimal for conditional generation tasks. For the task of generating the remaining 11 leads in a 12-lead generation task, we aim for the prior distribution of the generated leads to be at least correlated with the observed single-lead signal, rather than a Gaussian distribution. In our proposed method, we introduce condition-related shifts in the forward process to incrementally modify it, while preserving the noise sampled from the Gaussian distribution. Through this approach, the model can learn a better prior distribution, optimizing the trajectory of the entire forward and reverse process and making learning the accurate distribution of the 12 leads more feasible.

Additionally, in our approach, the diffusion model learns the distribution of ECG in the time-frequency domain rather than the time domain. Considering the strong periodicity, ECG in the time-frequency domain exhibits more obvious features. Learning the distribution of ECG in the time-frequency domain may yield better results.

4.2 Forward Process and Reverse Process

We will provide a detailed explanation of the process involved in constructing such a Markov chain for 12-lead ECG synthesis. Let us consider the following problem: given a single-lead ECG observation (In our experiment, it is I-lead) $\boldsymbol{z}^c$, we generate the remaining target 11-lead ECG $\boldsymbol{z}^{tar}$ with extra patient information v_s. v_s serves as additional features for the ECG, and the computation process takes place within Sect. 5.2. We define a shifting sequence $\{k_t\}_{t=1}^{T}$, which exhibits a monotonically increasing trend with the timestep t and fulfills the conditions $k_1 \to 0$ and $k_T \to 1$. Subsequently, the formulation of the transition distribution at timestep t is derived from this shifting sequence as outlined below:

$$q_\phi(\boldsymbol{z}_t^{tar}|\boldsymbol{z}_{t-1}^{tar}, \boldsymbol{z}^c, v_s) = \mathcal{N}(\boldsymbol{z}_t^{tar}; \boldsymbol{z}_{t-1}^{tar} + \alpha_t s_\phi(\boldsymbol{z}^c, v_s), \alpha_t \boldsymbol{I}) \tag{5}$$

where $[z^c, z_0^{tar}] = z_q$, $\alpha_t = k_t - k_{t-1}$ for $t > 1$ and $\boldsymbol{I}$ is the identity matrix. $s_\phi(\cdot)$ is the shift network with trainable parameters ϕ, it takes the observed single-lead ECG $\boldsymbol{z}^c$ and side information v_s as inputs and produces the shift with the same dimension as $\boldsymbol{z}_0^{tar}$. Then, we demonstrate that the marginal distribution at each timestep t is amenable to analytical integration, specifically,

$$q_\phi(\boldsymbol{z}_t^{tar}|\boldsymbol{z}_0^{tar}, \boldsymbol{z}^c, v_s) = \mathcal{N}(\boldsymbol{z}_t^{tar}; \boldsymbol{z}_0^{tar} + k_t s_\phi(\boldsymbol{z}^c, v_s), k_t \boldsymbol{I}) \tag{6}$$

To ensure a smooth transition between $\boldsymbol{z}_t^{tar}$ and $\boldsymbol{z}_{t-1}^{tar}$, to the standard deviation, denoted as $\bar{\alpha}_t$, is introduced. The rationale behind this lies in bounding the expected distance between $\boldsymbol{z}_t^{tar}$ and $\boldsymbol{z}_{t-1}^{tar}$ within $\bar{\alpha}_t$, given that the ECG data falls within the range of $[0, 1]$ after pre-processing.

$$max[(\boldsymbol{z}_0^{tar} + k_t\hat{s}) - (\boldsymbol{z}_0^{tar} + k_{t-1}\hat{s})] = max[\alpha_t \hat{s}] < \alpha_t < \sqrt{\alpha_t}, \tag{7}$$

where $\hat{s} = s_\phi(\boldsymbol{z}^c, v_s)$, $max(\cdot)$ represents the point-wise maximizing operation. Besides the mean parameter, specifically $\boldsymbol{z}_0^{tar} + \alpha_t \hat{s}$, which contributes to the marginal distribution described in 6. Moreover, the marginal distributions of $\boldsymbol{z}_1^{tar}$ and $\boldsymbol{z}_T^{tar}$ converge to $\delta_{z_0^{tar}}(\cdot)$ and $\mathcal{N}(\cdot; \boldsymbol{z}^c, \boldsymbol{I})$, serving as approximations for the target and observation distributions, respectively. Through the deliberate construction of the Markov chain, it becomes feasible to address the synthesis task by reverse sampling from it given the observed single-lead ECG $\boldsymbol{z}^c$.

Following [20,30,31] the objective of the reverse process is to infer the posterior distribution $p(\boldsymbol{z}^{tar}|\boldsymbol{z}^{c})$ through the subsequent expression:

$$p(\boldsymbol{z}^{tar}|\boldsymbol{z}^{c}) = \int p(\boldsymbol{z}_T^{tar}|\boldsymbol{z}^{c}) \prod_{t=1}^{T} p^{\theta}(\boldsymbol{z}_{t-1}^{tar}|\boldsymbol{z}_t^{tar}, \boldsymbol{z}^{c}) d\boldsymbol{z}_{1:T}^{tar}, \tag{8}$$

where $p(\boldsymbol{z}_T^{tar}|\boldsymbol{z}^{c}) \approx \mathcal{N}(\boldsymbol{z}_T^{tar}|\boldsymbol{z}^{c}, \boldsymbol{I})$, $p_\theta(\boldsymbol{z}_{t-1}^{tar}|\boldsymbol{z}_t^{tar}, \boldsymbol{z}^{c})$ is the reverse transition kernel from $\boldsymbol{z}_t^{tar}$ to $\boldsymbol{z}_{t-1}^{tar}$ with a learnable parameter θ. Following most of the literature in the diffusion model, we adopt the assumption:

$$p_\theta(\boldsymbol{z}_{t-1}^{tar}|\boldsymbol{z}_t^{tar}, \boldsymbol{z}^{c}) = \mathcal{N}(\boldsymbol{z}_{t-1}^{tar}; \mu_\theta(\boldsymbol{z}_t^{tar}, \boldsymbol{z}^{c}, t), \Sigma_\theta(\boldsymbol{z}_t^{tar}, \boldsymbol{z}^{c}, t)) \tag{9}$$

Combining 5 and 6, the targeted distribution $q(\boldsymbol{z}_{t-1}^{tar}|\boldsymbol{z}_t^{tar}, \boldsymbol{z}_0^{tar}, \boldsymbol{z}^{c})$ in 9 can be rendered tractable and expressed in an explicit form given below:

$$q(\boldsymbol{z}_{t-1}^{tar}|\boldsymbol{z}_t^{tar}, \boldsymbol{z}_0^{tar}, \boldsymbol{z}^{c}) = \mathcal{N}(\boldsymbol{z}_{t-1}^{tar}|\frac{k_{t-1}}{k_t}\boldsymbol{z}_t^{tar} + \frac{\alpha_t}{k_t}\boldsymbol{z}_0^{tar}, \frac{k_{t-1}}{k_t}\alpha_t \boldsymbol{I}) \tag{10}$$

The detailed calculation of this derivation is presented in supplementary material. Considering that the variance parameter is independent of $\boldsymbol{z}_t^{tar}$ and $\boldsymbol{z}^{c}$, we thus set $\Sigma_\theta(\boldsymbol{z}_t^{tar}, \boldsymbol{z}^{c}, t) = \frac{k_{t-1}}{k_t}\alpha_t \boldsymbol{I}$. Because $\boldsymbol{z}_t^{tar}$ is known, the mean parameter is parameterized as below:

$$\mu_\theta(\boldsymbol{z}_t^{tar}, \boldsymbol{z}^{c}, t) = \frac{k_{t-1}}{k_t}\boldsymbol{z}_t^{tar} + \frac{\alpha_t}{k_t} f_\theta(\boldsymbol{z}_t^{tar}, \boldsymbol{z}^{c}, t) \tag{11}$$

where f_θ is a deep neural network with parameter θ.

4.3 Training and Sampling

We introduce the derivation of the final optimization objective for the training SD-ECG. Following the standard diffusion process, the objective function of SD-ECG can be written as:

$$\begin{aligned} \mathcal{L}_{\theta,\phi} = \mathbb{E}_{q_\phi(\boldsymbol{z}_{1:T}^{tar}|\boldsymbol{z}_0^{tar}, \boldsymbol{z}^{c})} \Big[& D_{\mathrm{KL}}(q_\phi(\boldsymbol{z}_T^{tar}|\boldsymbol{z}_0^{tar}, \boldsymbol{z}^{c}) \| p(\boldsymbol{z}_T^{tar}|\boldsymbol{z}^{c})) \\ & + \sum_{t>1} D_{\mathrm{KL}}(q_\phi(\boldsymbol{z}_{t-1}^{tar}|\boldsymbol{z}_t^{tar}, \boldsymbol{z}_0^{tar}, \boldsymbol{z}^{c}) \| p_\theta(\boldsymbol{z}_{t-1}^{tar}|\boldsymbol{z}_t^{tar}, \boldsymbol{z}^{c})) \\ & - \log p_\theta(\boldsymbol{z}_0^{tar}|\boldsymbol{z}_1^{tar}, \boldsymbol{z}^{c}) \Big], \end{aligned} \tag{12}$$

However, this training objective can be hard to stabilize [26]. Thus we simplify the objective as follows:

$$\mathcal{L}_{\theta,\phi,t} = \Big|\Big|\boldsymbol{\mu}_\theta(\boldsymbol{z}_t^{tar}, \boldsymbol{z}^{c}, t) - \frac{\alpha_t}{k_t}\boldsymbol{z}_0^{tar} - \frac{k_{t-1}}{k_t}\boldsymbol{z}_t^{tar}\Big|\Big|_2^2, \tag{13}$$

With 11, the objective can be further simplified as:

$$\min_\theta \sum_t w_t || f_\theta(\boldsymbol{z}_t^{tar}, \boldsymbol{z}^{c}, t) - \boldsymbol{z}_0^{tar} ||_2^2 \tag{14}$$

where $w_t = \frac{\alpha_t}{2k_t k_{t-1}}$. In practical experimentation, we observe empirically that excluding the weight w_t leads to a noticeable enhancement in performance, consistent with the findings in [20].

Recall that the target ECG is generated in the time-frequency domain at each step t. We provide the pseudo-code of the sampling process at the inference stage in 1.

Algorithm 1. Sampling Algorithm of SD-ECG

```
Input: Observed single-lead ECG $\boldsymbol{z}^c$, sampling steps $t_1^T$, shifting sequence $k_t$, the learned shifter model $s_\phi$, the learned reverse model $f_\theta$.
Output: The forecasting $\hat{s}^{ta}$.
Collecting side information $v_s$,
Computing shift $\hat{s} = s_\phi(\boldsymbol{z}^c, v_s)$
Sample $\hat{z}_T^{ta} \sim \mathcal{N}(0, \boldsymbol{I})$
Reverse process start point $z_T^{ta} = \hat{z}_T^{ta} + \hat{s}$
for $t = T, T-1, \cdots, 1$ do
    $\alpha_t = k_t - k_{t-1}$
    Computing mean $\mu_t = \frac{k_{t-1}}{k_t}\boldsymbol{z}_t^{tar} + \frac{\alpha_t}{k_t} f_\theta(\boldsymbol{z}_t^{tar}, \boldsymbol{z}^c, t)$
    Computing variance $\Sigma_t = \frac{k_{t-1}}{k_t}\alpha_t \boldsymbol{I}$
    Sampling $\hat{\boldsymbol{z}}_{t-1}^{tar} \sim \mathcal{N}(\boldsymbol{z}_{t-1}^{tar}; \mu_t, \sigma_t)$
end for
Decoding $\hat{s}^{ta} = \mathrm{ISTFT}(\hat{\boldsymbol{z}}_0^{tar})$
```

5 SD-ECG Architecture

In this section, as illustrated in Fig. 3, we provide the architecture of SD-ECG in detail. It is worth noting that the shift network is jointly trained with the main network. More experimental details are provided in Sect. 5.1.

5.1 Main Network

Building upon the preceding discussion in Sect. 4.2, the primary inputs to the main network encompass the collected single-lead ECG and the generated targets. We adopt bidirectional transformer architecture following previous works [22,32], given its established efficacy in learning distributions of sequence data. Here, time embeddings and leads correlation serve as supplementary information guiding the generation process. We use 128-dimensions time embedding following previous studies [33]:

$$s_{embedding}(s_l) = \left(\sin(s_l/\tau^{0/64}), \ldots, \sin(s_l/\tau^{63/64}), \cos(s_l/\tau^{0/64}), \ldots, \cos(s_l/\tau^{63/64})\right) \tag{15}$$

where $\tau = 10000$. The leads correlation matrix is a learnable 128-dimensional embedding representing the associations between different leads. Notably, considering the significance of lead correlations in the time-frequency domain for the 12-lead ECG, we introduce additional cross-attention modules at each layer to amplify the influence of lead correlations.

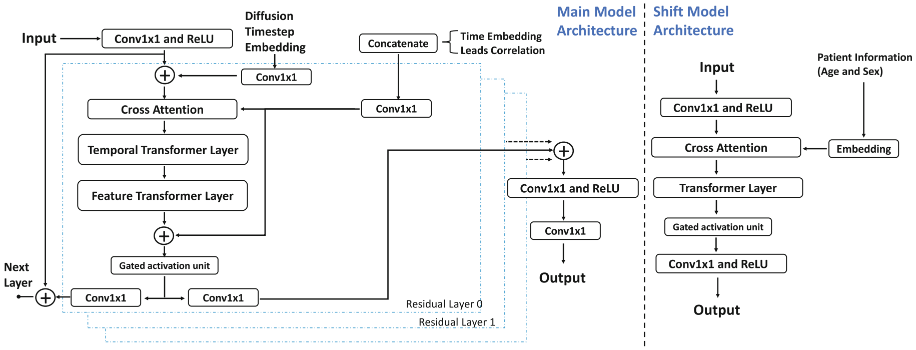


Fig. 3. Architecture of SD-ECG. The left part of the figure describes the structure of the main network f_θ. The model's inputs include the observed values (collected single-lead ECG), the targets (generated remaining leads), as well as side information. The model comprises observed values and updated targets. The right part of the figure describes the structure of the shift network $s_\phi(\cdot)$. The input of the shift network consists of observed values and basic physiological information relevant to this ECG segment and the output is the corresponding shift.

5.2 Shift Network

As for the shift network, the input consists of the observed single-lead ECG and the output is the shift. Considering that ECG features often vary with age and gender, and aiming to maximize the differentiation of the entire prior shift, we further incorporate additional physiological information embeddings (age represented numerically, gender represented by tokens 0,1 and 2). These embeddings are expanded to the same dimensions as the inputs for computation. The shift network runs only once at the beginning of the sampling process and thus does not incur excessive time costs.

6 Experiments

6.1 Experimental Setup

Datasets and Baselines. We conduct experiments using the PTB-XL dataset [34], MiT-BIH Arrhythmia Database [35], and the Tianchi ECG dataset[1]. The

---

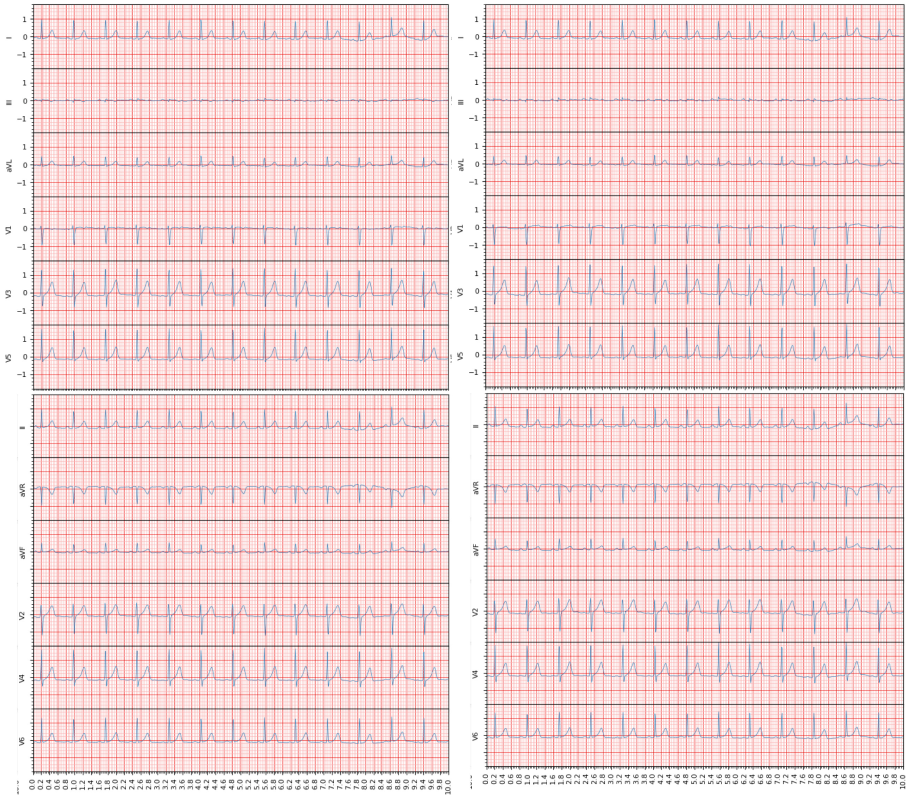
[1] https://tianchi.aliyun.com/competition/entrance/231754/information?lang=en-us.

PTB-XL dataset comprises 21837 clinical 12-lead ECGs from 18885 patients of 10-second length at a frequency of 500 Hz, while the Tianchi dataset includes 31,779 12-lead ECG signals recorded at a frequency of 500 Hz. The MIT-BIH Arrhythmia Database contains 48 half-hour excerpts of two-channel ambulatory ECG recordings. The datasets were randomly partitioned into training and testing sets, with probabilities of 0.7 and 0.3, respectively. We performed preprocessing, including down-sampling to 125 Hz, conversion to the time-frequency domain, and linear scaling normalization to the range of 0 to 1. For baselines, we choose existing single-lead to 12-lead ECG synthesis methods, including GAN [18], Nef-Net [19], VAE-CNN [36] and E-LSTM [37].

Table 1. Performance comparisons on PTB-XL in terms of MSE and FD score. The best is in bold.

	GAN		VAE-CNN		E-LSTM		Nef-Net		SD-ECG (Ours)	
	MSE	FD score	MSE	FD score	MSE	FD score	MSE	FD score	MSE	FD score
Lead II	0.0080	2.561	0.0102	3.183	0.0091	2.639	0.0062	2.159	**0.0036**	**0.783**
Lead III	0.0080	1.974	0.0085	2.312	0.0092	2.159	0.0076	2.031	**0.0039**	**1.014**
Lead aVR	0.0090	0.359	0.0097	2.010	0.0096	2.746	0.0073	1.455	**0.0010**	**0.124**
Lead aVL	0.0020	0.219	0.0081	1.081	0.0066	1.893	0.0041	1.573	**0.0012**	**0.139**
Lead aVF	0.0080	2.272	0.0121	2.481	0.0071	2.033	0.0056	1.936	**0.0049**	**0.652**
V1	0.0160	8.032	0.0337	13.125	0.0213	12.263	0.0178	8.396	**0.0132**	**3.195**
V2	0.0290	9.382	0.0379	13.312	0.0219	7.342	0.0189	5.621	**0.0140**	**3.991**
V3	**0.0310**	9.373	0.0512	14.111	0.0424	10.351	0.0397	7.215	0.0312	**6.315**
V4	0.0240	7.929	0.0531	12.081	0.0480	9.259	0.0320	6.216	**0.0239**	**5.105**
V5	0.0300	12.153	0.0537	11.315	0.0487	10.969	0.0323	9.216	**0.0231**	**4.084**
V6	0.0260	19.453	0.0591	14.180	0.0382	13.215	0.0319	10.001	**0.0105**	**3.056**
Average	0.0170	6.701	0.0306	8.108	0.0238	6.806	0.0184	5.074	**0.0117**	**2.587**

Implementation Details. For the preprocessing, we utilize the STFT with a window length of 16 points. We retain only the low-frequency components in the time-frequency spectrum, which reduces computational costs without affecting the results. For training the diffusion model, we utilized the Adam optimizer with an initial learning rate of 10^{-3}. In the training process of the shifted diffusion, the batch size is 64, and training includes early stopping for a maximum of 200 epochs. For the 10 s 12-lead ECG signal, the shape of the input is (64, 48, 313). The number of diffusion steps T is set to 100, and the linear variance schedule starts from $K_0 = 0.001$ to $K_T = 0.999$. All experiments are conducted on an Nvidia RTX 3090 GPU with 24 GB memory.



Left: Synthesized 10S 12-lead ECG

Right: Original 10s 12-lead ECG

Fig. 4. Case study. The comparison between the ECG synthesized by our method and the corresponding ground truth. Our approach is capable of synthesizing accurate 12-lead ECG with sufficient length using single-lead ECG as input.

Metrics. We use Mean Square Error (MSE) and Fréchet Distance (FD) as metrics. MSE are calculated in the formula below, $\hat{Y}$ represents the synthesized ECG, and Y represents the ground truth ECG. MSE calculates the average squared difference between predictions and true values:

$$MSE = \sqrt{mean(|\hat{Y} - Y|)} \tag{16}$$

Following [38], the FD score between two ECGs is defined as the FD of their corresponding continuous functions:

$$FD(\hat{Y}(t), Y(t)) = \inf_{\hat{Y}} \max_{t} \|\hat{Y}(t) - Y(t)\|, \tag{17}$$

A smaller MSE or FD score implies better synthesis.

Table 2. Classification performances (ROC-AUC) on the MIT-BIH dataset augmented by synthesized data. In the experiment of SimDCGAN [40], the ECG is synthesized from scratch. The best results are in **bold**.

Diseases	GAN	VAE-CNN	E-LSTM	SimDCGAN	Nef-Net	Ours
SVEB	0.758±0.05	0.697±0.02	0.795±0.07	0.724±0.05	0.782±0.05	**0.802**±0.09
VEB	0.987±0.03	0.953±0.00	0.981±0.01	0.980±0.01	**0.983**±0.00	0.981±0.02
FUSION	0.801±0.07	0.763±0.10	0.783±0.05	0.827±0.09	0.855±0.07	**0.866**±0.013

6.2 Experimental Results

Our model was trained and tested on the PTB-XL dataset and the results are shown in Table 1. Our method outperforms existing time series diffusion models. Compared to the baselines, our approach demonstrates superior performance in 10 out of 11 lead reconstruction tasks. Overall, our method significantly surpasses the current state-of-the-art methods (Fig. 4).

Besides, we directly applied the trained model to the data from the Tianchi and MIT-BIH datasets, achieving satisfactory results as well without any additional fine-tuning. As shown in Figure, the comparison on the Tianchi dataset demonstrates that our method is capable of synthesizing realistic and accurate long 12-lead ECG. This indicates that our model has learned the distribution of the 12-lead ECG rather than merely replicating the content from the train set.

To assess the ability of SD-ECG to capture important information in the single-lead ECG, we provide abnormal single-lead ECG from the MIT-BIH dataset as input and observe the performance of synthesized 12-lead ECGs under the classifier. If the synthesized leads can be classified as abnormal, we consider that the generated segments contain important information from the observation. The results are shown in Table 2, and the classifier is implemented as [39]. As shown in the table, our model demonstrates superiority in two diseases (SVEB(%2) and FUSION(%1.1)), indicating that SD-ECG is capable of generating 12-lead ECGs with important information.

Table 3. Average MSEs by different variants of the conditioning network.

In TF domain	With shift network	With Patient Information	*PTB-XL*	*Tianchi*
			0.147	0.0209
✓			0.0.145	0.0207
	✓		0.0139	0.0199
✓	✓		0.0117	0.0146
	✓	✓	0.0117	0.0146
✓	✓	✓	0.0117	0.0146

6.3 Model Analysis

Ablation Study. To evaluate the effectiveness of the mechanisms in our method, we conduct a comprehensive assessment by comparing the full version of SD-ECG with five variants on two datasets. The MSE for 11 leads is presented as the result in Table 3. In the table, "in TF domain" denotes whether the model learns the distribution in the time-frequency domain, and "Use patient information" indicates whether real patient data is fed into the shift network. The symbol ✓ indicates the mechanism is utilized in the network, while a blank denotes its absence. As observed, the shift mechanism significantly influences the model's performance, while the introduction of patient information has minimal impact on performance improvement. This suggests that the model can be used while preserving patient privacy without compromising performance.

Impact of Diffusion Steps. To mitigate the impact of uncertainty while retaining informative temporal patterns, configuring the diffusion steps is crucial. Inadequately small diffusion steps may result in a process lacking meaningful outcomes, while substantial steps may lead to uncontrolled diffusion. In this study, we investigated the effect of the number of diffusion steps, denoted as T, on *Tianchi* and *PTB-XL*. We varied T within the range of 50 to 500. As shown in Fig. 5, the optimal value for T is approximately 100 for SD-ECG, and increasing T beyond this value does not significantly improve the results.

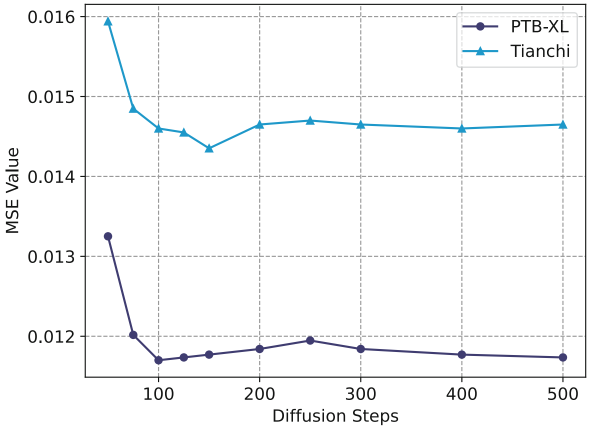


Fig. 5. Comparisons of predictions with different T on two datasets.

7 Conclusion

In this paper, we propose SD-ECG, an innovative framework that represents the first attempt to introduce the diffusion model into the task of synthesis of

12-lead ECG from single-lead ECG. By incorporating additional shifts in the forward process, our model is better equipped to learn the accurate distribution of the spectrogram, enabling more precise predictions through the utilization of frequency domain information. Comprehensive experiments conducted on three real-world datasets demonstrate the outstanding performance of SD-ECG in improving the quality of probabilistic predictions, underscoring its effectiveness.

Acknowledgements. This work is supported by the National Natural Science Foundation of China (No. 62172018, No. 62102008) and Wuhan East Lake High-Tech Development Zone National Comprehensive Experimental Base for Governance of Intelligent Society.

References

1. Gaziano, T., Reddy, K.S., Paccaud, F, et al.: Cardiovascular disease. Disease Control Priorities in Developing Countries, 2nd edn. (2006)
2. Dahlöf, B.: Cardiovascular disease risk factors: epidemiology and risk assessment. Am. J. Cardiol. **105**(1), 3A-9A (2010)
3. De Bacquer, D., De Backer, G., Kornitzer, M., et al.: Prognostic value of ECG findings for total, cardiovascular disease, and coronary heart disease death in men and women[J]. Heart **80**(6), 570–577 (1998)
4. Luz, E.J.S., Schwartz, W.R., Cámara-Chávez, G., et al.: ECG-based heartbeat classification for arrhythmia detection: a survey. Comput. Methods Programs Biomed. **127**, 144–164 (2016)
5. Berkaya, S.K., Uysal, A.K., Gunal, E.S., et al.: A survey on ECG analysis. Biomed. Signal Process. Control **43**, 216–235 (2018)
6. Denes, P.: The importance of derived 12-lead electrocardiography in the interpretation of arrhythmias detected by Holter recording[J]. Am. Heart J. **124**(4), 905–911 (1992)
7. Vogiatzis, I., Koulouris, E., Ioannidis, A., et al.: The importance of the 15-lead versus 12-lead ECG recordings in the diagnosis and treatment of right ventricle and left ventricle posterior and lateral wall acute myocardial infarctions[J]. Acta Informatica Medica **27**(1), 35 (2019)
8. Mortara, J.L.: ECG acquisition and signal processing: 12-lead ECG acquisition. Cardiac Safety of Noncardiac Drugs: Practical Guidelines for Clinical Research and Drug Development, pp. 131–145. Humana Press, Totowa, NJ (2005)
9. Maheshwari, S., Acharyya, A., Rajalakshmi, P., et al.: Accurate and reliable 3-lead to 12-lead ECG reconstruction methodology for remote health monitoring applications. IRBM **35**(6), 341–350 (2014)
10. Mehta, D.D., Nazir, N.T., Trohman, R.G., et al.: Single-lead portable ECG devices: perceptions and clinical accuracy compared to conventional cardiac monitoring. J. Electrocardiol. **48**(4), 710–716 (2015)
11. Boriani, G., Palmisano, P., Malavasi, V.L., et al.: Clinical factors associated with atrial fibrillation detection on single-time point screening using a hand-held single-lead ECG device. J. Clin. Med. **10**(4), 729 (2021)
12. Gifari, M.W., Zakaria, H., Mengko, R.: Design of ECG Homecare: 12-lead ECG acquisition using single channel ECG device developed on AD8232 analog front end. In: 2015 International Conference on Electrical Engineering and Informatics (ICEEI). IEEE, pp. 371–376 (2015)

13. Lee, H.J., Lee, D.S., Kwon, H.B., et al.: Reconstruction of 12-lead ECG using a single-patch device. Methods Inf. Med. **56**(04), 319–327 (2017)
14. Khunti, K.: Accurate interpretation of the 12-lead ECG electrode placement: a systematic review. Health Educ. J. **73**(5), 610–623 (2014)
15. Atoui, H., Fayn, J., Rubel, P.: A neural network approach for patient-specific 12-lead ECG synthesis in patient monitoring environments. IEEE Comput. Cardiology **2004**, 161–164 (2004)
16. Kachenoura, A., Porée, F., Carrault, G., et al.: Non-linear 12-lead ECG synthesis from two intracardiac recordings. In: 2009 36th Annual Computers in Cardiology Conference (CinC), pp. 577–580. IEEE (2009)
17. Gundlapalle, V., Acharyya, A.: A Novel Single Lead to 12-Lead ECG reconstruction methodology using convolutional neural networks and LSTM. In: 2022 IEEE 13th Latin America Symposium on Circuits and System (LASCAS), pp. 01–04. IEEE (2022)
18. Seo, H.C., Yoon, G.W., Joo, S., et al.: Multiple electrocardiogram generator with single-lead electrocardiogram. Comput. Methods Programs Biomed. **221**, 106858 (2022)
19. Chen, J., Zheng, X., Yu, H., et al.: Electrocardio panorama: synthesizing new ECG views with self-supervision. arXiv preprint arXiv:2105.06293, 2021
20. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. Adv. Neural. Inf. Process. Syst. **33**, 6840–6851 (2020)
21. Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models. arXiv preprint arXiv:2010.02502 (2020)
22. Tashiro, Y., Song, J., Song, Y., et al.: Csdi: conditional score-based diffusion models for probabilistic time series imputation. Adv. Neural. Inf. Process. Syst. **34**, 24804–24816 (2021)
23. Alcaraz, J.M.L., Strodthoff, N.: Diffusion-based time series imputation and forecasting with structured state space models. arXiv preprint arXiv:2208.09399 (2022)
24. Shen, L., Kwok, J.: Non-autoregressive conditional diffusion models for time series prediction. In: International Conference on Machine Learning. PMLR, pp. 31016–31029 (2023)
25. Adib, E., Fernandez, A.S., Afghah, F., et al.: Synthetic ecg signal generation using probabilistic diffusion models. IEEE Access (2023)
26. Nichol, A.Q., Dhariwal, P.: Improved denoising diffusion probabilistic models. In: International Conference on Machine Learning. PMLR, pp. 8162–8171 (2021)
27. Alcaraz, J.M.L., Strodthoff, N.: Diffusion-based conditional ECG generation with structured state space models. Comput. Biol. Med. **163**, 107115 (2023)
28. Gu, A., Goel, K., Ré, C.: Efficiently modeling long sequences with structured state spaces. arXiv preprint arXiv:2111.00396 (2021)
29. Zama, M.H., Schwenker, F.: ECG synthesis via diffusion-based state space augmented transformer. Sensors **23**(19), 8328 (2023)
30. Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., et al.: Deep unsupervised learning using nonequilibrium thermodynamics. In: International Conference on Machine Learning. PMLR, pp. 2256–2265 (2015)
31. Song, Y., Sohl-Dickstein, J., Kingma, D.P., et al.: Score-based generative modeling through stochastic differential equations. arXiv preprint arXiv:2011.13456 (2020)
32. Kong, Z., Ping, W., Huang, J., et al.: Diffwave: a versatile diffusion model for audio synthesis. arXiv preprint arXiv:2009.09761 (2020)
33. Vaswani, A., Shazeer, N., Parmar, N., et al.: Attention is all you need. Advances in neural information processing systems, 30 (2017)

34. Wagner, P., Strodthoff, N., Bousseljot, R.D., et al.: PTB-XL, a large publicly available electrocardiography dataset. Sci. Data **7**(1), 1–15 (2020)
35. Moody, G.B., Mark, R.G.: The impact of the MIT-BIH arrhythmia database. IEEE Eng. Med. Biol. Mag. **20**(3), 45–50 (2001)
36. Matyschik, M., Mauranen, H., Bonizzi, P., et al.: Feasibility of ECG reconstruction from minimal lead sets using convolutional neural networks. In: 2020 Computing in Cardiology. IEEE, pp. 1–4 (2020)
37. Sohn, J., Yang, S., Lee, J., et al.: Reconstruction of 12-lead electrocardiogram from a three-lead patch-type device using a LSTM network. Sensors **20**(11), 3278 (2020)
38. Driemel, A., Krivošija, A., Sohler, C.: Clustering time series under the Fréchet distance. In: Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms. Society for Industrial and Applied Mathematics, pp. 766–785 (2016)
39. Kachuee, M., Fazeli, S., Sarrafzadeh, M.: Ecg heartbeat classification: a deep transferable representation. In: 2018 IEEE International Conference on Healthcare Informatics (ICHI), pp. 443–444. IEEE (2018)
40. SimGANs: Simulator-based generative adversarial networks for ECG synthesis to improve deep ECG classification

SAGS-DynamicBio: Integrating Semantic-Aware and Graph Structure-Aware Embedding for Dynamic Biological Data with Knowledge Graphs

Yao Liu[1], Yongfei Zhang[1,2](✉), and Xin Wang[3]

[1] Beijing Key Laboratory of Digital Media, School of Computer Science and Engineering, Beihang University, Beijing 100191, China
`{ly0708,yfzhang}@buaa.edu.cn`
[2] State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, Beijing 100191, China
[3] Department of Computer Science and Technology, Tsinghua University, Beijing, China
`xin-wang18@mails.tsinghua.edu.cn`

Abstract. Accurate prediction of drug-target interactions (DTIs) is critical for drug design and optimization in pharmacology. Existing models face challenges such as data sparsity and lack of contextual information, resulting in poor accuracy. Knowledge graphs (KGs) provide a solution by representing relationships in biological data. However, current KG-based DTI methods are limited to static graphs that require time-consuming retraining when knowledge is updated. In this paper, we propose SAGS-DynamicBio, an efficient dynamic embedding model for biological data that integrates semantics and graph structure information. We first generate KGs for the biological knowledge base, representing drugs and targets as entities and interactions as relations. Using KG embedding techniques, we convert each entity and relation into a vector representation. To effectively handle dynamic data, we introduce a semantic perception module based on the attention mechanism, which uses information from neighboring nodes to generate initial representation vectors for new data. Furthermore, we apply graph structure-based representation learning to these initial vectors to satisfy KG's structural constraints and improve prediction accuracy. To evaluate the effectiveness of our method, we conduct experiments comparing SAGS-DynamicBio with existing KG-based DTI prediction models and generic KGE models. The experimental results show that our method significantly improves the embedding efficiency, reducing the embedding time by 41.5% on average, while maintaining a high prediction accuracy, which proves the effectiveness of our method. SAGS-DynamicBio is able to efficiently adapt to the dynamic data updates without retraining the whole graph, thus providing a promising solution for DTI prediction in real-time scenarios.

Keywords: Dynamic Biological Data · Knowledge Graphs · Graph Network

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
A. Bifet et al. (Eds.): ECML PKDD 2024, LNAI 14949, pp. 287–302, 2024.
https://doi.org/10.1007/978-3-031-70378-2_18

1 Introduction

Drug-Target Interactions (DTI) [1,2] plays a critical role in pharmacology and healthcare. It involves the study of the potential effects that occur when two or more drugs are administered simultaneously and helps identify previously unknown interactions between drugs and protein targets in the human body. Understanding and predicting DTIs is of paramount importance in ensuring patient safety, optimizing drug therapy and avoiding adverse drug reactions. With the increasing complexity of drug combinations and the ever-growing number of available drugs, accurate DTIs prediction has become a challenging task.

In recent years, Knowledge Graph Embedding (KGE) techniques have emerged as a promising approach for predicting drug-target interactions (DTIs) [3–6]. Knowledge graphs (KGs) [7] provide a structured representation of drug interactions by capturing the relationships between drugs and their associated properties. These relationships are modeled as (subject, predicate, object) (SPO) triples, such as (*aspirin, drug-target, COX-1*). By exploiting these graph structures, KGE enables the learning of low-dimensional representations for drugs and their interactions. These learned embeddings serve as condensed features for various downstream machine learning tasks, including node classification [8], clustering [9], link prediction [10], and visualization [11], providing significant opportunities in the field of biomedical data science.

Previous studies on predicting DTIs using KGE techniques have proposed different models. Typical models include TriModel [12], NeoDTI [13] and (EEG)-DTI [14]. However, a major drawback of these models is their limited ability to effectively handle dynamic graphs. In real-world scenarios, the DTI network is constantly updated and changed due to the discovery of new drugs, new protein targets, or evolving biological knowledge. As a result, the knowledge graph representing the DTI network must be frequently updated to reflect these changes. Current models require re-training each time the graph is updated, which can be a time-consuming process, especially for large graphs.

To address this challenge, we propose a novel approach that efficiently manages dynamic graphs. Our approach leverages semantic and structural awareness derived from neighborhood information to improve the process of learning embeddings for new data. By incorporating these insights, our model can effectively update the knowledge graph embeddings without requiring complete re-training. This capability enables real-time updates and predictions, resulting in significant savings in computational time and resources.

In our approach, we first construct a KG using the available drug target knowledge base. We then use KGE techniques to address the task of identifying drug-target associations as a link prediction problem. To generate semantic representations for new data, we use the embeddings of neighboring nodes in the KG. To prioritize the acquisition of valid information and prevent the incorporation of invalid data, we incorporate an attention mechanism. We then fine-tune the resulting semantic representations to ensure compliance with the structural constraints within the KG. These structural constraints are defined by KGEs such as $\mathbf{e_s} \odot \boldsymbol{r} \approx \mathbf{e_t}$ in RotatE [15] or $\mathbf{e_s} + \mathbf{r} \approx \mathbf{e_t}$ in TransE [16], where $\mathbf{e_s}$

and $\mathbf{e_t}$ denote the source and target entity embedding, respectively, and $\mathbf{r}$ represents the relation embedding. It is expected that all triples in the KG follow this structural form. The resulting embedding of the new data obtained at the end serves a dual purpose: it captures semantic information while conforming to the structural constraints within the KG. Our contribution can be summarized as follows:

- We present the SAGS-DynamicBio embedding models, designed specifically for dynamic biological data, which provide an efficient solution for integrating semantic information from neighboring data and structural information from KGs to meet the real-time needs of downstream tasks.
- We perform a thorough evaluation of our proposed model on four biological benchmark datasets, providing a comprehensive assessment of its performance. The experimental results demonstrate the embedding efficiency of the model, which significantly outperforms existing approaches.
- We conduct experiments comparing our method with traditional KGE models to verify the generality and scalability of our method. The results demonstrate the superior performance of our method in terms of both accuracy and embedding time.

The rest of this paper is organized as follows: Sect. 2 provides an overview of related work in the area of DTIs prediction and KGE. Section 3 describes our proposed methodology in detail. Section 4 presents experimental results and evaluations of our approach. Finally, Sect. 5 concludes the paper.

2 Related Work

This section provides a comprehensive summary of recent advances in DTIs techniques using KGE. It includes KGE methods applicable to generic models as well as models specifically designed for biological data.

2.1 Knowledge Graph Embedding

In the field of knowledge graph embedding (KGE) models, various approaches have been developed, falling into three main categories: translation-based models, semantic matching models, and rotation-based models.

Translation-based models, such as TransE [16], consider relations as translations in vector space, aiming to bring head and tail entity vectors closer together. While effective for simple relations, they struggle with complex relationships. TransH [17] addresses this limitation by projecting entity embeddings onto relation-specific hyperplanes. TransD [18] takes a different approach by using entity projection vectors instead of dense matrices, resulting in parameter efficiency.

Semantic matching models focus on comparing the semantic similarity between entities or relations in text and those in the knowledge graph. RESCAL [19] utilizes a bilinear scoring function to solve a three-way rank-r matrix factorization problem. DistMult [20] reduces the number of free parameters by enforcing the relation embedding matrix to be diagonal. ComplEx [21] extends the bilinear product score to the complex vector space, enabling more effective modeling of antisymmetric relations.

Rotation-based embedding models map entities and relations to a complex vector space, enhancing representational capabilities. RotatE [15] is an example of such a model, leveraging rotations in the complex plane. HAKE [22] takes a unique approach by rewriting the rotation formula using polar coordinates and separating the scoring function into phase and modulus components.

2.2 Knowledge Graph Embedding for DTIs

The integration of knowledge graphs and Drug-Target Interactions (DTIs) has emerged as a promising approach for studying DTIs, offering new avenues for exploring drug-target relationships and advancing drug discovery and development.

Mohamed et al. [12] introduced the TriModel, a novel approach that leverages embedded learning of DTI knowledge graphs. By considering structural information in the knowledge graphs (KGs), this model overcomes previous limitations and treats the problem as link prediction in KGs. This approach incorporates prior information about drugs and targets, enhancing the predictive power of the model. DTiGEMS+ [23] improves drug-target interaction graphs by incorporating drug-drug and target-target similarity graphs. By considering the similarity between drugs and targets, this model enhances the understanding of complex interactions and provides a more comprehensive view of DTIs. The NeoDTI model [13] presents a novel nonlinear end-to-end learning approach that integrates diverse information from heterogeneous graph data. By leveraging the power of heterogeneous graphs, this model captures the complex relationships between drugs and targets, enabling more accurate predictions and a deeper understanding of DTIs. The EEG-DTI model [14] utilizes graph convolutional networks to predict DTIs based on low-dimensional feature representations of drugs and targets within heterogeneous graphs. This approach effectively captures the structural information of the graph and leverages it for accurate prediction of DTIs. GraphDTA [24] employs graph neural networks to predict drug-target affinities. By exploiting the expressive power of graph neural networks, this model captures the intricate relationships between drugs and targets, enabling precise predictions of drug-target affinities. Zhao et al. [25] propose an approach that utilizes graph convolutional networks (GCNs) to learn potential representations. By leveraging the capabilities of GCNs, this model effectively identifies potential drug-target interactions, facilitating the exploration of novel drug-target pairs.

However, both generic knowledge graph embedding models and knowledge graph embedding models tailored for biological databases are unable to handle dynamic graphs. Each update requires the model to be retrained, resulting in a time-consuming process that limits the ability to meet the real-time requirements of downstream tasks.

3 Methodology

In this section, we describe our proposed methods in detail. Formally, a KG $\mathcal{G}$ is represented as a directed graph: $G = \{(e_s, r, e_t), e_s, e_t \in E, r \in R\}$, where E is the set of entities and R is the set of relations. Each directed link in the knowledge graph $l = (e_s, r, e_t) \in G$ corresponds to a fact triple, e.g., (*aspirin, drug-target, COX-1*). We use bold to denote embedding vectors, for example, $\mathbf{e_s}$ represents the embedding representation of e_s.

3.1 Knowledge Graph Embedding

We construct a KG based on the biomedical knowledge base. Specifically, drugs and targets that have interactions are connected by relationships or links within the knowledge graph. On the other hand, drugs and targets that do not have interactions are not directly connected.

To learn embeddings of the KG, we use existing KGE methods. We take RotatE as an example. The training process is as follows.

Firstly, we map entities, including drugs and targets, and relations (whether there is a connection or not) to a complex space. Relations are defined as rotations from e_s to the e_t, allowing modeling of directional relationships. For each triple $l = (e_s, r, e_t)$, RotatE uses a distance function to measure the compatibility between the entities and the relation, as follows:

$$d_r(\mathbf{e_s}, \mathbf{e_t}) = \|\mathbf{e_s} \circ \mathbf{r} - \mathbf{e_s}\| \tag{1}$$

where $\circ$ is the Hadamard product. We then utilize the self-adversarial training negative sampling loss function to compute the scores of the triples in the KG:

$$p\left({e_s}'_j, r, {e_t}'_j \mid \{({e_s}_i, r_i, {e_t}_i)\}\right) = \frac{\exp \alpha f_r\left({\mathbf{e_s}}'_j, {\mathbf{e_t}}'_j\right)}{\sum_i \exp \alpha f_r\left({\mathbf{e_s}}'_i, {\mathbf{e_t}}'_i\right)} \tag{2}$$

$$L = -\log \sigma\left(\gamma - d_r(\mathbf{e_s}, \mathbf{e_t})\right) - \sum_{i=1}^{n} p\left({e_s}'_i, r_i, t'_i\right) \log \sigma\left(d_r\left({\mathbf{e_s}}'_i, {\mathbf{e_t}}'_i\right) - \gamma\right) \tag{3}$$

where α is the sampling temperature, γ is a fixed margin, σ is sigmoid function, $({e_s}'_i, r, {e_t}'_i)$ is the ith negative sample.

Based on the loss score, we use gradient updating to optimize the embedding vector for a number of iterations. The goal is to minimize the loss score and improve the quality of the embeddings.

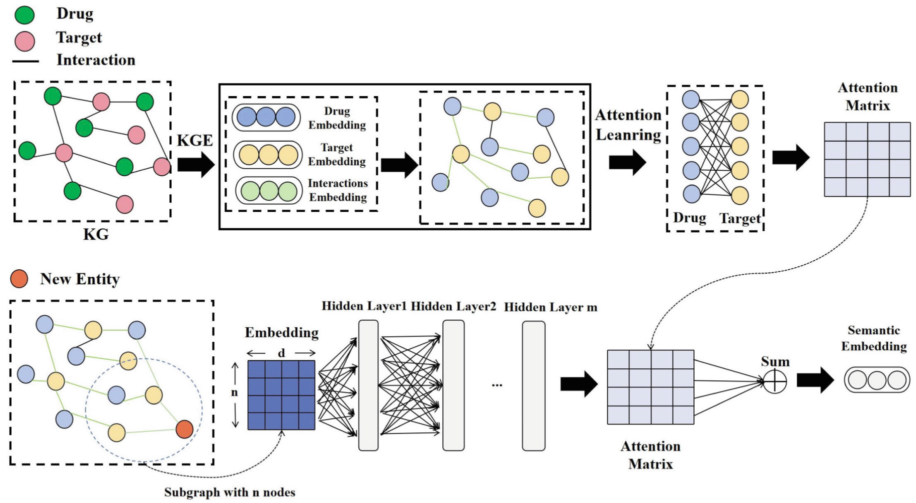


Fig. 1. The overivew of semantic-aware learning based on attention mechanism. First, the KG is constructed. The KGE method is used to generate embedding for the constructed KG. Then, the association strength between entities is learned and the attention matrix is generated. When a new node is added, the neighborhood information is merged by a convolution operation in the hidden layer. The strength of association with each neighboring node is calculated using the obtained attention matrix and finally the weighted sum is output.

3.2 Semantic-Aware Learning Based on Attention Mechanism

After applying the KGE method to obtain embedded representations of drugs, targets, and relations in the KG, we use the information from neighboring nodes to efficiently generate representations for new data. The close connection between neighboring nodes and target nodes allows us to capture contextual information and semantic features about the target nodes, which can represent the meaning and role of the target nodes in the KG. Furthermore, considering that neighboring nodes make different contributions to the target node, with some being more strongly associated while others have weaker relationships, we use the attention mechanism to learn the degree of association between neighboring nodes and the target node. Figure 1 illustrates the structure of our method.

Specifically, given an embedding of an entity $\mathbf{e_i}$, we calculate the weights in relation to the other entities to get the weight vector α_i. The calculation of α_i is as follows:

$$\text{score}\left(\mathbf{e_j}, \mathbf{e_i}\right) = \mathbf{u}^T \, \text{ReLU}\left(\mathbf{e_j} \odot \mathbf{e_i}\right) \tag{4}$$

$$\alpha_i = \frac{\exp\left(\text{score}\left(\mathbf{e_j}, \mathbf{e_i}\right)\right)}{\sum_{i=1}^{n} \exp\left(\text{score}\left(\mathbf{e_j}, \mathbf{e_i}\right)\right)} \tag{5}$$

where $\mathbf{e_j}$ represents the vector representation of e_j, $\mathbf{e_i}$ represents the vector representation of the entity e_i, and $i \neq j$. $\mathbf{u}$ represents the parameter vector of the attention network, $\odot$ represents the element-wise multiplication, and score $(\mathbf{e_j}, \mathbf{e_i})$ represents the correlation between e_j and e_i.

After calculating the correlation between all entities, an attention matrix $\boldsymbol{A}$ is obtained, where $\boldsymbol{A}(i, j)$ represents the correlation value between $e_i$and e_j.

Next, we employ Graph Convolutional Networks (GCN) to capture relevant information from neighboring nodes and combine the weight information derived from the attention model to generate an initial embedding representation of the new node. This initial embedding representation contains contextual information and semantic features provided by neighboring nodes. We define the directly connected nodes of the new node e_n as (e_i, e_j). The initial semantic representation formula is as follows:

$$\mathrm{Sem}_{\mathbf{e_n}} = \boldsymbol{A}(n, i) \odot \mathbf{e_i} + \boldsymbol{A}(n, j) \odot \mathbf{e_j} \tag{6}$$

where $\mathrm{Sem}_{\mathbf{e_n}}$ denotes the initial embedded representation obtained, which is defined as $\mathrm{Sem}_{\mathbf{e_n}}$ since it contains only semantic information about the context.

3.3 Graph Structure-Aware Learning

The KG contains structural information, such as the requirement in TransE that $\mathbf{e_s} + \mathbf{r} \approx \mathbf{e_t}$, in RotatE that $\mathbf{e_s} \odot \mathbf{r} \approx \mathbf{e_t}$, in TransR that $\|\mathbf{e_s} + \mathbf{r} - \mathbf{e_t}\|_{\mathbf{2}}^{\mathbf{2}} \approx \mathbf{0}$.

Thus, in different embedding methods, entities and relations are linked in different structures. We train the obtained vectors, which contain semantic information, to satisfy the inherent structural constraints through an iterative embedding training. During this process, the vectors are refined and adapted to capture both the semantic and structural aspects of the KG. Additionally, we adopt the neighbor 3-hop triples for embedding training. This choice allows us to avoid retraining the entire graph and to exploit the maximally correlated data within the neighbor 3-hop region. By fine-tuning the already available semantic information, we ensure that our goal is achieved.

Specifically, using RotatE as an example, a semantic embedding of a new node e_n denoted as $Sem_{\mathbf{e_n}}$, which is directly connected to the entity e_i via the relation r_i, and to the entity e_j via the relation r_j. The required structure is:

$$Sem_{\mathbf{e_n}} \odot \mathbf{r_i} \approx \mathbf{e_i} \tag{7}$$

$$Sem_{\mathbf{e_n}} \odot \mathbf{r_j} \approx \mathbf{e_j} \tag{8}$$

We iteratively train the embedding, using e_n as the central node, and include all triples within 3 hops to satisfy the structural constraints.

3.4 Training for Joint Semantic-Aware and Graph Structure-Aware

The overall training process of our model is depicted in Fig. 2. Firstly, we construct the KG based on the biological database and generate the embedded representation of this graph using the existing KGE method. Then, an attention model is trained using the obtained embedding representation to capture important information. Next, the GCN is used to capture the information of neighboring nodes and combined with the attention model to generate the initial representation vector (semantic representation) of the new node. Finally, this initial representation vector is iteratively trained using a specific optimization algorithm, such as gradient descent, to learn structural information until the best result is obtained.

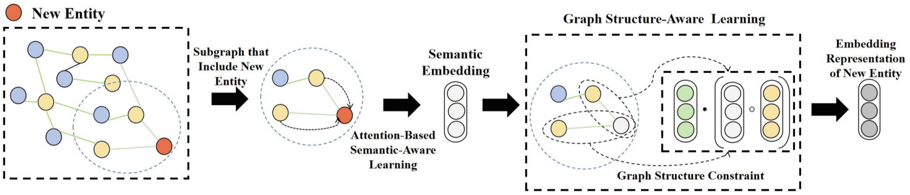

New Entity
Subgraph that Include New Entity
Attention-Based Semantic-Aware Learning
Semantic Embedding
Graph Structure-Aware Learning
Graph Structure Constraint
Embedding Representation of New Entity

Fig. 2. The training process of our method. When a new node is added, we first obtain the embedding learning with semantic information through semantic-aware learning based on the attention mechanism. Then, we iteratively train this embedding representation by incorporating the structural constraints of the graph through graph structure-aware learning.

4 Experiments

4.1 Experiment Setup

We present a comprehensive performance analysis of our approach by comparing its performance before and after integration on four benchmark datasets specifically designed for the DTIs task. Additionally, we test the efficiency of our method on standard KGE datasets. Finally, we perform ablation experiments to analyze the impact of individual components in our approach. Our experiments conduct on an Ubuntu 18.04.4 environment with 62GB of RAM and a 60G GPU. The software versions used include Pytorch.

Datasets. We employ four benchmark datasets, each representing a different class of target proteins, namely enzymes (Es), ion channels (ICs), G protein-coupled receptors (GPCRs), and nuclear receptors (NRs). These datasets have been curated and simulated using information extracted from publicly available databases, including KEGG BRITE [26], BRENDA [27], SuperTarget [28], and DrugBank [29]. The use of these datasets allows us to evaluate the performance and generalizability of our proposed model across different protein classes. Table 1 summarizes the statistical information for the four datasets.

To ensure effective learning and dynamic updates, we divide the dataset into two parts. Specifically, we allocated two thirds of the dataset for learning embeddings, while reserving one third for dynamic updating.

Table 1. Dataset description.

Datasets	Nuclear Receptor	GPCR	Ion Channel	Enzyme
Drugs	54	223	201	445
Targets	26	95	204	664
Interactions	90	635	1476	2926

Evaluation Protocols. We employ two widely recognized evaluation metrics for DTIs task: AUC (Area Under the Curve) and AUPR (Area Under the Precision-Recall Curve). Both AUC and AUPR metrics range from 0 to 1, with higher values indicating better model performance.

The AUC metric measures the area under the ROC (Receiver Operating Characteristic) curve. The ROC curve plots the true positive rate (sensitivity) against the false positive rate (1-specificity) at various classification thresholds. A higher AUC value indicates better discrimination ability and a more accurate model.

The AUPR metric calculates the area under the Precision-Recall curve. The Precision-Recall curve plots the precision (positive predictive value) against the recall (sensitivity) at various classification thresholds. A higher precision signifies a lower false-positive rate, while a higher recall indicates a lower false-negative rate.

Baselines. We evaluate and compare the performance of our proposed model with several state-of-the-art KGE-based models specifically designed for DTIs tasks.

TriModel [12] uses a triple scoring mechanism to learn entity and relation embeddings, enabling accurate representation of drug-target interactions.

TBSelfNet-DTI [30] uses a self-attention mechanism to capture the importance of neighboring entities and relations, enabling a more comprehensive understanding of the underlying biological context.

Zhongyu He's [31] introduces a novel embedding method that uses graph convolutional networks (GCNs) to capture both local and global dependencies in knowledge graphs.

The experimental data for each baseline model are extracted from the original papers, ensuring consistency and reproducibility in our evaluation.

Hyper-parameters Settings. We employ a grid search technique to identify the optimal hyperparameters, performing an average of 10 iterations. The experiments are conducted on GPUs using the PyTorch. The optimal settings for hyper-parameters are as follows:

Enzyme dataset: Batch_size - 128, Epoch - 90000, Dimension - 200, Number of Attention Layers - 3 layers, Number of Nodes in the Hidden Layer of the Attention Network - 128, Learning rate - 0.001, Number of Hops of Structure Learning Nodes - 3 hops.

Table 2. The accuracy results for the DTI task. The best results are shown in bold.

Model	Enzyme		Ion Channel	
	AUC	AUPR	AUC	AUPR
TriModel [12]	0.952	**0.780**	0.940	0.760
TriModel (+SAGS-DynamicBio)	**0.971**	0.760	**0.941**	0.729
TBSelfNet-DTI [30]	0.9731	**0.7496**	0.9785	**0.8387**
TBSelfNet-DTI (+SAGS-DynamicBio)	**0.9740**	0.7489	**0.9870**	0.8350
Zhongyu He. [31]	0.9670	**0.810**	0.9710	**0.821**
Zhongyu He. (+SAGS-DynamicBio)	**0.9778**	0.808	**0.9778**	0.809

Table 3. The accuracy results for the DTI task. The best results are shown in bold.

Model	GPCR		Nuclear Receptor	
	AUC	AUPR	AUC	AUPR
TriModel [12]	0.920	**0.810**	0.900	0.870
TriModel (+SAGS-DynamicBio)	**0.931**	0.798	**0.911**	0.865
TBSelfNet-DTI [30]	0.9786	0.7804	0.9271	**0.7697**
TBSelfNet-DTI (+SAGS-DynamicBio)	**0.9862**	**0.7811**	**0.9308**	0.7590
Zhongyu He. [31]	0.9567	**0.742**	0.9307	0.711
Zhongyu He. (+SAGS-DynamicBio)	**0.9609**	0.729	**0.9398**	**0.712**

Ion Channel dataset: Batch_size - 128, Epoch - 90000, Dimension - 300, Number of Attention Layers - 3 layers, Number of Attention Network Hidden Layer Nodes - 128, Learning rate - 0.001, Number of Hops of Structure Learning Nodes - 3 hops.

GPCR dataset: Batch_size - 256, Epoch - 100000, Dimension - 500, Number of Attention Layers - 3 layers, Number of Attention Network Hidden Layer Nodes - 128, Learning rate - 0.001, Number of Attention Network Hidden Layer Nodes - 256, learning rate - 0.001, Number of Hops of Structure Learning Nodes- 3 hops.

Nuclear Receptor dataset: Batch_size - 512, Epoch - 150000, Dimension - 1000, Number of Attention Layers - 3 layers, Number of Attention Network Hidden Layer Nodes - 256, Learning rate - 0.001, Number of Hops of Structure Learning Nodes- 3 hops.

4.2 Drug-Target Interactions Task

Accuracy. Table 2 and Table 3 present the results of the embedding quality evaluation. The results demonstrate that our dynamic embedding method achieves comparable performance to the baseline model, as observed in the AUC. Specifically, the models incorporating our approach show an average improvement of 0.0071, with a maximum improvement of 0.019. This improvement corresponds

Table 4. The results of the embedding times for the DTIs task. The best results are shown in bold.

Model	Enzyme	Ion Channel	GPCR	Nuclear Receptor
TriModel [12]	>1.5h	>4.5h	>6.2h	>7.2h
TriModel(+SAGS-DynamicBio)	≈**51.2** min	≈**1.7**h	≈**4.2**h	≈**5.3**h
TBSelfNet-DTI [30]	>1.7h	>4.7h	>5.9h	>8.5h
TBSelfNet-DTI(+SAGS-DynamicBio)	≈**50.2** min	≈**1.7**h	≈**4.5**h	≈**5.7**h
Zhongyu He. [31]	>1.2h	>3.5h	>5.5h	>8.5h
Zhongyu He.(+SAGS-DynamicBio)	≈**46.2** min	≈**1.9**h	≈**4.7**h	≈**5.9**h

Table 5. The results of link prediction task on FB15k and WN18.

Model	FB15k		WN18	
	MRR	Hits@10	MRR	Hits@10
TransE [16]	0.380	0.641	0.454	0.934
TransE + SAGS-DynamicBio	**0.411**	**0.662**	**0.481**	**0.953**
TransD [18]	-	0.773	-	0.925
TransD + SAGS-DynamicBio	**0.632**	**0.801**	**0.934**	**0.944**
TransH [17]	-	0.644	-	0.867
TransH + SAGS-DynamicBio	**0.578**	**0.684**	**0.911**	**0.912**
RotatE [15]	0.797	0.884	0.949	0.959
RotatE + SAGS-DynamicBio	**0.823**	**0.899**	**0.961**	**0.971**
ComplEx [21]	0.692	0.840	0.941	0.947
ComplEx + SAGS-DynamicBio	**0.734**	**0.873**	**0.957**	**0.962**
DistMult [20]	0.654	0.824	0.822	0.936
DistMult + SAGS-DynamicBio	**0.684**	**0.856**	**0.845**	**0.951**

to a 2% increase in accuracy, highlighting the improved classification ability of our models. Here, a decrease in AUPR indicates a potential decrease in the ability to accurately identify certain critical interactions, such as the differential effects of certain drugs and targets. This suggests that the embedding approach may weaken the ability to effectively differentiate such interactions. In future work, we aim to enhance the ability to accurately differentiate between different types of interactions related to the same target.

Embedding Time. The embedding time results are summarized in Table 4. Our model outperforms other baseline models in terms of embedding efficiency by eliminating the need to retrain the added data using the KGE approach. This key feature significantly improves the embedding efficiency, as shown by an average improvement of 41.5%.

Based on the results presented, our model demonstrates significant improvements in both the accuracy of DTIs classification and the efficiency of updating. The results show an average improvement of 0.0071 in accuracy compared to the baseline model, indicating the enhanced performance of our approach. Additionally, our model exhibits notable improvements in updating efficiency, as evidenced by embedding time. These findings collectively highlight the effectiveness and efficiency of our model in addressing dynamic biological data.

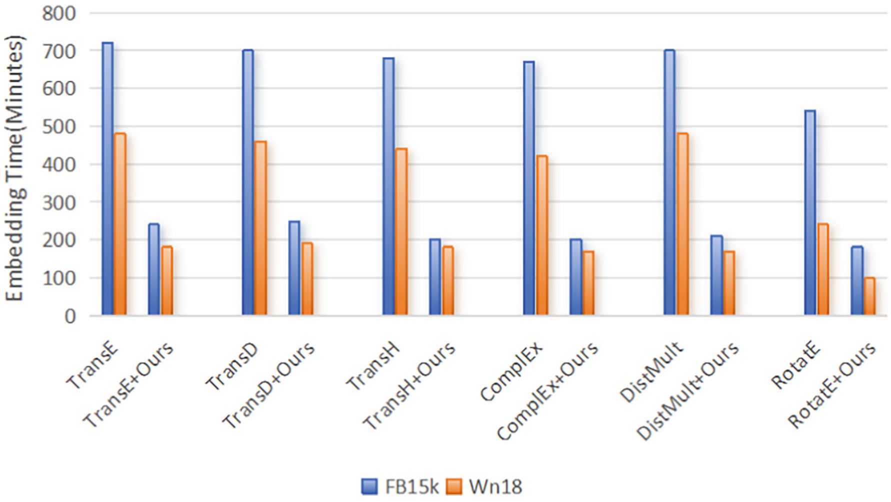


Fig. 3. The results for embedding time for link prediction task on the FB15k and WN18 datasets.

4.3 Link Prediction Task

To ensure the validity and generalizability of our approach, we utilize two widely used standardized datasets, FB15k [16] and WN18 [16]. These datasets have been specifically designed for KGE and are well-established benchmarks in the field. By using these datasets, we mitigate potential errors arising from sparse or unevenly distributed biological data.

To evaluate the performance of our approach, we compare its accuracy and embedding time with a traditional KGE model on the link prediction task. The goal of link prediction is to anticipate potential relationships or connections between two entities based on existing graphical information. To measure performance, we use two standard evaluation metrics: Mean Reciprocal Rank (MRR) and Hits@10. MRR and Hits@10 both return values in the range [0, 1], with higher values indicating better performance in predicting links.

Table 6. The results of Ablation studies for link prediction task on FB15k.

Model	MRR	Hits@10		MRR	Hits@10
TransE [16]	0.380	0.641	TransD [18]	-	0.773
TransE + Sem-A	0.390	0.649	TransD + Sem-A	0.562	0.780
TransE + GraphS-A	0.401	0.654	TransD + GraphS-A	0.612	0.786
TransE + Ours	**0.411**	**0.662**	TransD + Ours	**0.632**	**0.801**
TransH [17]	-	0.644	RotatE [15]	0.797	0.884
TransH + Sem-A	0.542	0.651	RotatE + Sem-A	0.801	0.890
TransH + GraphS-A	0.556	0.660	RotatE + GraphS-A	0.812	0.894
TransH + Ours	**0.578**	**0.684**	RotatE + Ours	**0.823**	**0.899**
ComplEx [21]	0.692	0.840	DistMult [20]	0.654	0.824
ComplEx + Sem-A	0.703	0.851	DistMult + Sem-A	0.660	0.831
ComplEx + GraphS-A	0.721	0.863	DistMult + GraphS-A	0.672	0.840
ComplEx + Ours	**0.734**	**0.873**	DistMult + Ours	**0.684**	**0.856**

Accuracy. The results of the prediction accuracy comparison are presented in Table 5. Specifically, the MRR metric shows an average improvement of 0.03. This improvement reflects the improved ability of our model to accurately rank the most relevant predictions. Additionally, the Hits@10 metric shows an average improvement of 0.02, indicating an increased accuracy in identifying the correct predictions within the top 10 results. This improvement highlights the effectiveness of our model in accurately identifying the most relevant predictions among the top candidates.

Embedding Time. The embedding time results for each model on the FB15k and WN18 datasets are depicted in Fig. 3. Notably, our dynamic embedding model shows a significant time reduction of about 50% compared to other models. Furthermore, as discussed in the previous section, our accuracy results demonstrate that we have successfully reduced the update time while maintaining a high level of accuracy. This performance characteristic is particularly well suited to meet the real-time requirements of downstream tasks.

4.4 Ablation Studies

We evaluate and analyze the contribution of the semantic-aware and structural-aware modules in our study. To test the effectiveness of these modules, we fuse existing traditional KGE models and perform ablation studies. The results of these studies are presented in Table 6 and Table 7, which illustrate the impact of the semantic-aware (Sem-A) and graph structure-aware (GraphS-A) modules on the quality of embeddings. Based on the results presented in the tables, it is clear that both the semantic-aware and structure-aware modules play a crucial role in improving the quality of embeddings. This finding highlights the importance of

Table 7. The results of Ablation studies for link prediction task on WN18.

Model	MRR	Hits@10		MRR	Hits@10
TransE [16]	0.454	0.934	TransD [18]	-	0.925
TransE + Sem-A	0.464	0.944	TransD + Sem-A	0.918	0.929
TransE + GraphS-A	0.470	0.949	TransD + GraphS-A	0.924	0.934
TransE + Ours	**0.481**	**0.953**	TransD + Ours	**0.934**	**0.944**
TransH [17]	-	0.867	RotatE [15]	0.949	0.959
TransH + Sem-A	0.893	0.872	RotatE + Sem-A	0.951	0.962
TransH + GraphS-A	0.901	0.892	RotatE + GraphS-A	0.957	0.969
TransH + Ours	**0.911**	**0.912**	RotatE + Ours	**0.961**	**0.971**
ComplEx [21]	0.941	0.947	DistMult [20]	0.822	0.936
ComplEx + Sem-A	0.949	0.951	DistMult + Sem-A	0.829	0.941
ComplEx + GraphS-A	0.952	0.958	DistMult + GraphS-A	0.832	0.949
ComplEx + Ours	**0.957**	**0.962**	DistMult + Ours	**0.845**	**0.951**

incorporating both semantic and structural learning aspects to achieve optimal performance.

5 Conclusion

This paper presents a novel embedding model, SAGS-DynamicBio, specifically designed for dynamic biomedical data. Our proposed model effectively utilizes both semantic and structural information extracted from KGs to represent new data. By incorporating semantic perception modules based on attention mechanisms, our SAGS-DynamicBio model captures the contextual information and dependencies between entities and relations in the KGs. Our experimental results validate the effectiveness of the proposed model. Future work can focus on incorporating spatial structure and exploring the applicability of our model in other domains.

Acknowledgments. This work was partially supported by the National Natural Science Foundation of China (No. 62072022) and the Fundamental Research Funds for the Central Universities.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Alvarez-Mamani, E., et al.: Graph embedding on mass spectrometry-and sequencing-based biomedical data. BMC Bioinform. **25**(1), 1 (2024)
2. Veleiro, U., et al.: GENNIUS: an ultrafast drug-target interaction inference method based on graph neural networks. Bioinformatics **40**(1), p.btad774 (2024)
3. Ezzat, A., Zhao, P., Wu, M., Li, X.-L., Kwoh, C.-K.: Drug-target interaction prediction with graph regularized matrix factorization. In: IEEE/ACM Transactions on Computational Biology and Bioinformatics, vol. 14, no. 3, pp. 646–656, 1 May-June 2017. https://doi.org/10.1109/TCBB.2016.2530062.
4. Iliadis, D., De Baets, B., Pahikkala, T., Waegeman, W.: A comparison of embedding aggregation strategies in drug-target interaction prediction. BMC Bioinformatics **25**(1), 59 (2024)
5. Chen, M., et al.: Drug-target interactions prediction based on signed heterogeneous graph neural networks. Chinese J. Electron. **33**(1), 231–244 (2024)
6. Li, N., et al.: Drug-target interaction prediction using knowledge graph embedding. iScience (2024)
7. Ge, X., Wang, Y.C., Wang, B., Kuo, C.C.: Knowledge graph embedding: an overview. APSIPA Trans. Signal Inform. Process. **13**(1) (2024)
8. Liu, Y., Wang, P., Yang, D., Qiu, N.: A knowledge graph embedding model based attention mechanism for enhanced node information integration. PeerJ Computer Science. **22**(10), e1808 (2024)
9. Sanjak, J., Binder, J., Yadaw, A.S., Zhu, Q., Mathé, E.A.: Clustering rare diseases within an ontology-enriched knowledge graph. J. Am. Med. Inform. Assoc. **31**(1), 154–64 (2024)
10. Wang, J., Huang, H., Wu, Y., Zhang, F., Zhang, S., Guo, K.: Open knowledge graph link prediction with semantic-aware embedding. Expert Syst. Appl. **20**, 123542 (2024)
11. Guo, S., Wang, Q., Wang, B., et al.: Semantically smooth knowledge graph embedding. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), 84–94 (2015)
12. Mohamed, S.K., Nováček, V., Nounu, A.: Discovering protein drug targets using knowledge graph embeddings. Bioinformatics **36**(2), 603–610 (2020)
13. Wan, F., Hong, L., Xiao, A., et al.: NeoDTI: neural integration of neighbor information from a heterogeneous network for discovering new drug-target interactions. Bioinformatics **35**(1), 104–111 (2019)
14. Peng, J., Wang, Y., Guan, J., et al.: An end-to-end heterogeneous graph representation learning-based framework for drug-target interaction prediction. Briefings Bioinfor. **22**(5), bbaa430 (2021)
15. Sun, Z., Deng, Z.H., Nie, J.Y., et al.: Rotate: knowledge graph embedding by relational rotation in complex space[J]. ar**v preprint ar**v:1902.10197 (2019)
16. Bordes, A., Usunier, N., Garcia-Duran, A., et al.: Translating embeddings for modeling multi-relational data. Advances in neural information processing systems, 26 (2013)
17. Wang, Z., Zhang, J., Feng, J., et al.: Knowledge graph embedding by translating on hyperplanes. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 28(1) (2014)

18. Ji, G., He, S., Xu, L., et al.: Knowledge graph embedding via dynamic map** matrix. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, pp. 687–696 (2015)
19. Nickel, M., Tresp, V., Kriegel, H.P.: A three-way model for collective learning on multi-relational data. ICML **11**(10.5555), 3104482–3104584 (2011)
20. Yang, B., Yih, W., He, X., et al.: Embedding entities and relations for learning and inference in knowledge bases. ar**v preprint ar**v:1412.6575 (2014)
21. Trouillon, T., Welbl, J., Riedel, S., et al.: Complex embeddings for simple link prediction. In: International Conference on Machine Learning. PMLR, pp. 2071–2080 (2016)
22. Zhang, Z., Cai, J., Zhang, Y., et al.: Learning hierarchy-aware knowledge graph embeddings for link prediction. In: Proceedings of the AAAI conference on artificial intelligence, vol. 34(03), pp. 3065–3072 (2020)
23. Thafar, M.A., Olayan, R.S., Ashoor, H., et al.: DTiGEMS+: drug-target interaction prediction using graph embedding, graph mining, and similarity-based techniques[J]. J. Cheminformatics **12**, 1–17 (2020)
24. Nguyen, T., Le, H., Quinn, T.P., et al.: GraphDTA: predicting drug-target binding affinity with graph neural networks. Bioinformatics **37**(8), 1140–1147 (2021)
25. Zhao, T., Hu, Y., Valsdottir, L.R., et al.: Identifying drug-target interactions based on graph convolutional network and deep neural network. Brief. Bioinform. **22**(2), 2141–2150 (2021)
26. Kanehisa, M., Goto, S., Hattori, M., et al.: From genomics to chemical genomics: new developments in KEGG. Nucleic Acids Res. **34**(suppl_1), D354-D357 (2006)
27. Schomburg, I., Chang, A., Ebeling, C., et al.: BRENDA, the enzyme database: updates and major new developments. Nucleic Acids Res. **32**(suppl_1), D431-D433 (2004)
28. Günther, S., Kuhn, M., Dunkel, M., et al.: SuperTarget and Matador: resources for exploring drug-target relationships. Nucleic Acids Res. **36**(suppl_1), D919–D922 (2007)
29. Wishart, D.S., Knox, C., Guo, A.C., et al.: DrugBank: a knowledgebase for drugs, drug actions and drug targets. Nucleic Acids Res. **36**(suppl_1), D901–D906 (2008)
30. Zhang, Y., Cheng, M.: Drug-target interaction prediction based on knowledge graph embedding and BiLSTM networks. In: International Conference on Intelligent Computing. Singapore: Springer Nature Singapore, pp. 803–813 (2023)
31. He, Z.: Drug-target interaction prediction based on knowledge graph and convolutional neural network integrated with CBAM module. In: International Conference on Intelligent Computing. Singapore: Springer Nature Singapore, pp. 653–665 (2023)

Graph Machine Learning for Fast Product Development from Formulation Trials

Manuel Dileo[1(✉)], Raffaele Olmeda[2], Margherita Pindaro[2], and Matteo Zignani[1]

[1] Department of Computer Science, University of Milan, Milan, Italy
{manuel.dileo,matteo.zignani}@unimi.it
[2] Intellico s.r.l, Milan, Italy
{raffaele.olmeda,margherita.pindaro}@intellico.ai

Abstract. Product development is the process of creating and bringing a new or improved product to market. Formulation trials constitute a crucial stage in product development, often involving the exploration of numerous variables and product properties. Traditional methods of formulation trials involve time-consuming experimentation, trial and error, and iterative processes. In recent years, machine learning (ML) has emerged as a promising avenue to streamline this complex journey by enhancing efficiency, innovation, and customization. One of the paramount challenges in ML for product development is the models' lack of interpretability and explainability. This challenge poses significant limitations in gaining user trust, meeting regulatory requirements, and understanding the rationale behind ML-driven decisions. Moreover, formulation trials involve the exploration of relationships and similarities among previous preparations; however, data related to formulation are typically stored in tables and not in a network-like manner. To cope with the above challenges, we propose a general methodology for fast product development leveraging graph ML models, explainability techniques, and powerful data visualization tools. Starting from tabular formulation trials, our model simultaneously learns a latent graph between items and a downstream task, i.e. predicting consumer-appealing properties of a formulation. Subsequently, explainability techniques based on graphs, perturbation, and sensitivity analysis effectively support the R&D department in identifying new recipes for reaching a desired property. We evaluate our model on two datasets derived from a case study based on food design plus a standard benchmark from the healthcare domain. Results show the effectiveness of our model in predicting the outcome of new formulations. Thanks to our solution, the company has drastically reduced the labor-intensive experiments in real laboratories and the waste of materials.

Keywords: Product Development · Structure Learning · XAI for tabular data

---

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-70378-2_19.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
A. Bifet et al. (Eds.): ECML PKDD 2024, LNAI 14949, pp. 303–318, 2024.
https://doi.org/10.1007/978-3-031-70378-2_19

1 Introduction

Product development refers to the systematic process of designing, creating, and introducing new or improved products into the market. A fundamental step of this process is represented by the formulation trials, in which the research and development (R&D) department of industrial companies experiments with various ingredients, proportions, physical properties, and other factors to determine the optimal combination that meets the desired specifications and performance criteria. Conventional approaches to formulation trials typically utilize labor-intensive experimentation, trial and error, and iterative procedures, which can take several weeks to meet a desired formulation.

Over the past few years, machine learning (ML) has emerged as a promising solution for simplifying this process and enhancing efficiency, innovation, and customization. A main challenge in ML for product development is the models' lack of interpretability and explainability. This limitation poses significant burdens in gaining user trust, fulfilling regulatory standards, and understanding the logic behind ML-driven decisions. Moreover, formulation trials involve the exploration of relationships and similarities among previous preparations or solutions; but, data related to these formulations are typically stored in tables without explicit relationships between trials.

To cope with the above challenges, we propose a general methodology for fast product development leveraging graph machine learning models, explainability techniques, and powerful data visualization tools. Starting from tabular formulation trials, our model simultaneously learns a latent graph between items and a downstream task, i.e. predicting consumer-appealing properties of a formulation. This choice enables the model to explore latent correlations among formulations and feature values and allows transferable knowledge between similar product lines or iterative design processes. As a further step, explainability techniques based on graphs, perturbation, and sensitivity analysis effectively support the R&D department in their formulation process. Specifically, global-level explanations - related to the overall predictions of the model - allow them to identify the most influential characteristics for obtaining a certain property, while graph and single-level explanations - related to the prediction of single formulations - effectively support the users in identifying new recipes, the impact of changes in existing recipes, and the restocking of ingredients. The current solution is in the deployment phase and is offered to industrial companies through customized web applications.

As case study, we present the application of the above solution in the context of food design; presenting two real-world datasets collected in collaboration with a renowned company in the sector. In this scenario, industrial researchers are interested in finding recipes that satisfy specific sensorial properties of the product over time, given the composition of its ingredients and its physical properties. We model the problem of predicting sensorial properties as a multi-regression task. We evaluate our model on the two datasets, exhibiting the effectiveness of our solution compared to common tabular, graph, and structure learning approaches. Moreover, an ablation study highlights that the graph machine learning approach plays a crucial role in obtaining the best performance against the baselines. Finally, we also evaluate our method on a standard benchmark dataset for structure learning in the healthcare domain, showing an increase in performance compared to state-of-the-art models.

From the client side, our solutions are positively impacting different business and production metrics: the company is expected to reduce the labor-intensive experiments from 150 tests before approval to 30, the waste of materials dropped up to 30%, and the time to market has passed from seven to two months. We believe our work highlights a promising avenue for graph machine learning on general formulation trials in product development.

2 Background

In this section, we briefly provide background on graph neural networks. Then, we describe related works on the use of graph neural networks for product development and graph structure learning for tabular data, highlighting some works that provide also explainability techniques for the task.

Graph Neural Networks. Graph Neural Networks (GNNs) [18] are neural networks specifically designed to handle graph-structured data. Thanks to their ability to propagate and aggregate information across nodes and edges within a graph, they excel at capturing relationships and dependencies in graph-structured information to generate a vector-based representation for nodes, leveraging also node and/or edge-level attributes. This versatility has led to their application in diverse domains such as social network analysis [3], recommendation systems [7], or bioinformatics [5]. Modern GNNs rely on the 1-hop message-passing framework [6] for processing graph data. Specifically, given a graph $\mathcal{G} = (V, E)$, the representation of a node $v \in V$ at the l-layer of a GNN is obtained as a combination of the representation of the node v at layer $l - 1$ and aggregation of the representation of the nodes in the 1-hop neighborhood of v, where layer zero is represented by the initial node features. Different kinds of aggregation and combination functions can be considered to build different GNN layers.

GNNs for Product Development. Recently, Graph Neural Networks have been successfully applied in product development. In particular, GNNs are used for product design of items that can be naturally represented as graphs, meaningfully processed as 3D data, or whose interactions with other elements are explicitly defined. For instance, Bian *et al.* [1] formulate the material selection problem as a binary node-level classification task over the assembly of Computer-aided design (CAD) projects, modeled as a graph of material components, and leverage GNNs to obtain representation for materials. CAD models can be also represented as 3D structures, e.g. using Point Clouds [12], whose representation can be learned using GNNs [22]. In the context of drug design, the interaction between co-prescribed drugs and drug-target protein can be leveraged by GNNs for drug repurposing and identification of side effects [5]. In the last few years, GNNs were also employed for scaling deep learning for materials discovery and improving the modeling of material properties given their molecular structure or composition [17]. Although all these methods have shown successful applications of GNNs in product development, most of them assume a fixed graph structure between elements or explicit interactions between products. But, in most cases, formulation trials in product development are typically stored as tables and their complex relationships (e.g. recipe similarities, sharing knowledge between experiments) are not explicitly modeled. Moreover, in the context of food design, where industrial researchers are interested in consumer-appealing taste properties, the use of 3D data structures would not be as meaningful as in the context of manufacturer design.

Graph Learning for Tabular Data. In recent years, the community has underlined a critical gap in deep learning for tabular data: the lack of representation of latent correlations among data instances and feature values [14]. GNNs, due to their ability to model relationships between diverse elements of tabular data, have attracted considerable interest and have been applied across various steps of tabular data processing [14]. In particular, graph structure learning methods [24] aim to jointly learn an optimized latent graph structure among elements and an element-level downstream task. Among these approaches, only a few provide also explanations for the obtained results. For instance, Verdone *et al.* [21] utilizes GNNExplainer [23] for providing explainable spatio-temporal predictions for multi-site photovoltaic energy production. In the healthcare domain, Kazi *et al.* [9] provides an attention-based mechanism for interpretable Alzhaimer's disease predictions, while Li *et al.* [15] utilizes interpretable feature learning for Parkinson's disease diagnosis. To the best of our knowledge, no explainable graph machine-learning techniques have been applied in the context of product development and food design.

3 Dataset and Case-Study

We present a case study in the context of product development for the food industry introducing two real-world datasets collected in collaboration with a renowned company in the sector, Perfetti Van Melle (PVM). In particular, the PVM Lainate Labs is the entity in charge of running the analysis and trials necessary to create new recipes and formulations for their products. Given formulations of previously tested products and their corresponding consumer-appealing properties, the primary goal of our solution is to speed up product development by predicting the property values of new formulations. This allows R&D departments and labs to fast discover new potential formulations for a desired property. The next subsections will detail the case study and datasets used to test our solution.

Case-Study. Product development in the food industry involves traditionally several steps, which can be summarized as follows: (i) *Product Design*: defining in detail the consumer-appealing characteristics of the new product, based on several factors such as customer feedback, new ingredients availability, new strategies, and other market needs; (ii) *Recipe formulation*: creating the recipe to meet the desired design. This step is based mostly on experience matured over the years by the members of the research and development (R&D) department; (iii) *Laboratory Analysis*: measuring the physical properties of the product utilizing highly specialized technical equipment (e.g. rheometers) and monitoring the various aspects of the product through dashboards; (iv) *Process iteration*: at the end of the described steps, the overall quality of the product is evaluated. If the metrics are not aligned with the desired properties, the whole process is repeated iteratively. This process can take several weeks to obtain a desired recipe formulation. The goal of our solution is to allow the company to abandon an iteration-based process in favor of a data-driven approach, in which AI techniques can suggest a new product design, given the past trials and the desired consumer-appealing characteristics. Specifically, a machine learning model is trained to predict the properties of new formulations. Afterwards, explainability and advanced data analysis techniques are leveraged to support the R&D department in designing new recipes.

Dataset. We collected two datasets derived from the formulation trials tested by the R&D department of PVM. Each formulation trial is described by the following groups of features: (i) *Raw materials*: the ingredients of the recipe. They are values between zero and one, each value represents the percentage of a certain ingredient in the recipe. As a consequence, the sum of these features for each row must be one. Some recipes require specific ingredients, while others may not utilize all of them; (ii) *Physical properties*: features derived from the laboratory analysis (e.g. using rheometers). An example of physical property is the viscosity of the product; (iii) *Sensorial properties*: the consumer-appealing properties to predict. They are obtained thanks to a panel of several people who taste the recipe and measure a particular sensation multiple times over a defined time interval. In this work, we focus on trials for "malleability" and "toughness" sensorial properties. Hence, overall, we obtained two datasets, each of them consisting of one hundred formulation trials described by forty raw materials and twenty-three physical properties. More information about the dataset and its preprocessing can be found in the supplementary material.

4 Methodology

In this section, we describe the proposed methodology for predicting the outcome of new formulation trials for specific desired product properties, which allows fast product development for industrial labs. The methodology leverages the characteristics of previously tested formulations and an inferred underlying graph structure that captures the similarities between previous laboratory trials.

Figure 1 shows the pipeline of our solution. Starting from the characteristics of previously tested products, modeled as tabular data, the *Differentiable Graph Module* (DGM)-based model [8] simultaneously learns, in an end-to-end fashion, the downstream task, i.e. the properties of a product, and an optimal underlying latent graph. The *Explainer* shows the most important product characteristics for a certain expected property and the most important nodes, i.e. previously tested recipes, for reaching the desired characteristic. The *Explorer* allows the R&D department to visualize the obtained graph for investigating new possible trials. Finally, the *Simulation module* leverages the DGM module at inference time to simulate the outcome of new formulation trials before getting in real laboratories.

The current solution is in the deployment phase and is offered to industrial companies through customized web applications. The next subsections describe the pipeline modules and the main features of the web application in detail.

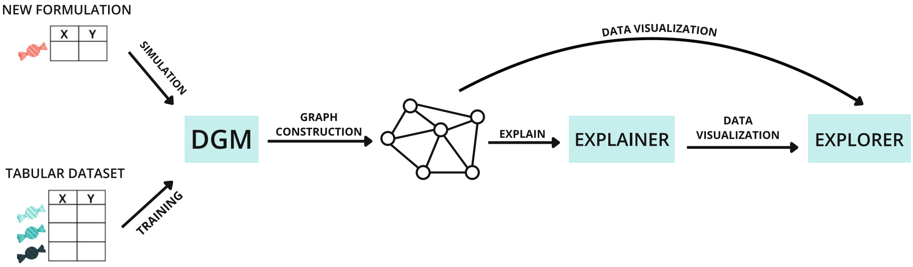


Fig. 1. Pipeline of our methodology for explainable prediction outcome of formulation trials.

4.1 DGM-Based Model

We model formulation trials as a $\mathbf{X} \in \mathbb{R}^{N \times d}$ matrix, where N is the number of formulations and d is the number of features (raw materials and physical properties). Each row of $\mathbf{X}$ has an associated vector $\mathbf{y} \in \mathbb{R}^z$ of consumer-appealing characteristics, where z is the number of considered characteristics. In our scenario, $\mathbf{y}$ represents a sensorial property with $z = 8$ sampled values over time.

We model the problem of predicting the outcomes of a formulation as a multi-regression task on the function $F : \mathbb{R}^d \mapsto \mathbb{R}^z$ that maps formulations to their consumer-appealing properties. Hence, the objective of the DGM-based model is to solve a multi-regression task and learn an optimal underlying latent graph for solving the task.

Initially, the model takes the input feature matrix $\mathbf{X} \in \mathbb{R}^{N \times d}$ and generates a graph $\mathcal{G}$ as its output. The process involves transforming input features $\mathbf{X} \in \mathbb{R}^{N \times d}$ into auxiliary features $\hat{\mathbf{X}} = f_{\Theta}(\mathbf{X}) \in \mathbb{R}^{N \times \hat{d}}$ using a parametric function f_{Θ}. Then, the auxiliary features $\hat{\mathbf{X}}$ are utilized for graph construction. In particular, the auxiliary features are used to construct a matrix $\mathbf{P} \in \mathbb{R}^{N \times N}$, where each element p_{ij} represents the probability of an edge between formulations i and j. Afterward, the probability matrix is leveraged to construct the adjacency matrix $\mathbf{A}$ of the graph $\mathcal{G}$. The edge probabilities are defined as follows:

$$p_{ij}(\mathbf{X}; \boldsymbol{\Theta}, t) = e^{-t\varDelta(\hat{\mathbf{x}}_i, \hat{\mathbf{x}}_j)^2} = e^{-t\varDelta(f_{\Theta}(\mathbf{x}_i), f_{\Theta}(\mathbf{x}_j))^2}, \tag{1}$$

Here, t is a learnable parameter, and $\varDelta(\cdot, \cdot)$ represents the Euclidean distance between two nodes in the graph embedding space.

Once the probability matrix $\mathbf{P}(\mathbf{X}; \boldsymbol{\Theta}, t)$ is obtained, a graph $\mathcal{G}$ is derived by constructing a sparse k-degree graph using the k-NN rule, as detailed in [8], obtaining the unweighted adjacency matrix $\mathbf{A}(\mathbf{X}; \boldsymbol{\Theta}, t)$.

Given the adjacency matrix $\mathbf{A}$, the second component of the model takes in input $\mathbf{A}$ and the initial features $\mathbf{X}^{(\mathbf{0})}$, yielding a new set of features $\mathbf{X}^{(1)} = g_{\boldsymbol{\Phi}}(\mathbf{X}^{(0)})$ as output, where $g_{\boldsymbol{\Phi}}$ represents a graph neural network function on $\mathbf{X}$ and $\mathbf{A}$. In our model, $g_{\boldsymbol{\Phi}}$ is a one-layer GCN [11].

The final node features $\mathbf{X}^{(1)}$ is then given as input to an MLP to obtain the final node predictions $\tilde{y}_i = \mathrm{MLP}(X^{(1)})$.

The entire DGM-based model is optimized in an end-to-end fashion constructing a compound loss function that provides incentives for edges involved in accurate predictions while imposing penalties for edges with large prediction errors [8]:

$$\mathcal{L}(y_i, \tilde{y}_i) = \sum_{\substack{i \in V \\ j:(i,j) \in E}} \delta(y_i, \tilde{y}_i) \log p_{ij} \tag{2}$$

where V is the set of nodes, E is the set of obtained edges, y_i and $\tilde{y}_i$ the correct and predicted values for node i, p_{ij} is the probability score for edge (i, j), and $\delta(y_i, \tilde{y}_i)$ is a reward function:

$$\delta(y_i, \tilde{y}_i) = \sum_{m=1}^{z} |y_i(m) - \tilde{y}_i(m)| \tag{3}$$

where z is the number of regression task, and the notation $y_i(m)$ indicates the correct value for the product i on property m.

4.2 Explainer Model

The *Explainer* takes the trained DGM-based model and its prediction(s), and it returns an explanation in the form of a small subgraph of the input graph together with a rank of the node features most influential for the prediction(s). Specifically, we adopt the GNNExplainer [23] method as it is the most well-known and consolidated explainability technique for graph neural networks.

GNNExplainer specifies an explanation as a rich subgraph of the entire graph the GNN was trained on, such that the subgraph maximizes the mutual information with GNN's prediction(s). This is achieved by formulating a mean field variational approximation and learning a real-valued graph mask that selects the important subgraph of the GNN's computation graph. Simultaneously, GNNExplainer also learns a feature mask that masks out unimportant node features.

In our application scenario, it is important to provide both local, i.e. related to single nodes, and global-level explanations. Global-level explanations are useful for identifying qualitatively the most important characteristics for obtaining a desired product property. Hence, we provide global-level explanations by computing global feature importance scores that can be obtained by simply averaging the feature mask learned by GNNExplainer. Alternatively, global feature importance scores can be provided by computing the permutation importance [2] on node features.

On the other hand, local-level explanations are useful for identifying potential starting recipes - nodes - for new formulation trials to test in our platform or real labs, as well as product characteristics - features - that impact the value of the target property. While potential starting recipes can be identified by analyzing the subgraphs learned by GNNExplainer, the feature mask provided by the model can only highlight the subset of the most influential features and it is not able to quantify the impact of the features on the value of the target property. Hence, we compute the local feature importance of the feature i for the node v as the percentage change in the target property of v in response to a 5% change in the value of i. The percentage of change is chosen following the R&D department's recommendations.

4.3 Explore and Simulate Product Development

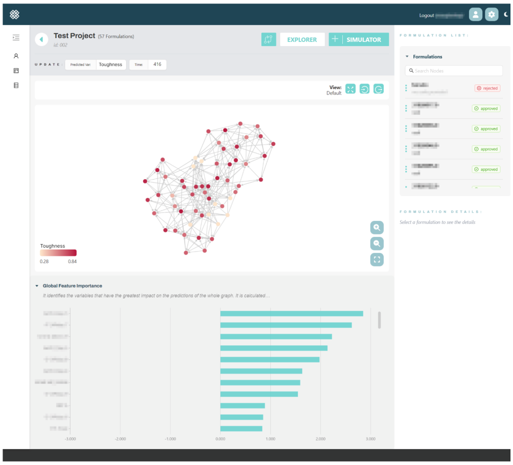
We visualize the results of the trained machine learning models and test them on new data in a web application. Specifically, once the user has imported tabular data representing the history of its laboratory trials, the application allows them to explore the learned graph representation, obtain local and global-level node explanations, and initiate simulations to predict outcomes of new products. Based on the results of the simulations, users can decide which products to test in real laboratories, and eventually approve their trials to add them as new training instances, updating the trained models in an online learning fashion. The next two paragraphs will detail the *Explorer* and *Simulation* modules.

Explorer. The *Explorer* is the data visualization module that allows users to visualize the complex relationships among their recipes, and the most influential characteristics/previous trials, to determine good starting points for potential new recipes. Figure 2 shows the graphical interface of the *Explorer* component. The graph constructed through the DGM-based model is displayed as an interactive graph on which users can interact to zoom in on details, select nodes, and view explanations and ancillary information. The interactive graph is built using the Cytoscape.js framework [4] and displayed using the Cola layout[1]. The platform offers filtered views for large graphs to ensure that users are directly

---

[1] https://marvl.infotech.monash.edu/webcola/, March 2024.

presented with the most interesting nodes. Specifically, the application utilizes the PageRank scores [13] to prioritize and suggest the subgraph around the most important nodes first. Below the interactive graph, a horizontal bar chart displays the global feature importance, which identifies the variables that most significantly impact the predictions across the entire network. Users can select a target node u to see its features, view its local explanations, and obtain the subgraph of the top N nodes starting from u using a dept- or a breadth-first search. The graphical interface for the *Explainer* is very similar to the *Explorer*, with the interactive graph representing the subgraph for a node-level explanation, and the bar chart showing its local feature importance.

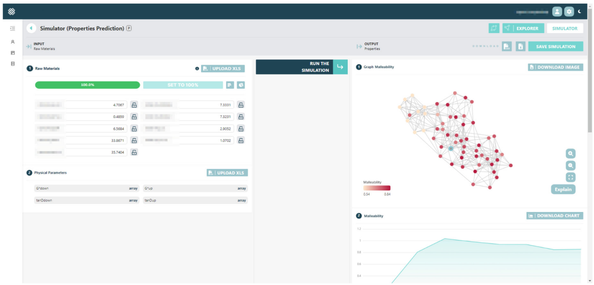


It identifies the variables that have the greatest impact on the predictions of the whole graph. It is calculated…

Select a formulation to see the details

Fig. 2. Example of the GUI for the *Explorer* component in the web application. The graph constructed through the DGM-based model is displayed as an interactive graph. Nodes are colored based on the predicted "toughness" value, the darker the higher. Below the interactive graph, a horizontal bar chart displays the global feature importance for predicting the "toughness".

Simulation. Users can initiate simulations for predicting product outcomes for specific properties from trained machine learning models. First, they can select an existing formulation, chosen with the *Explorer* component, to use as a good basis for a new trial and edit it to create a new recipe. Subsequently, they can run a simulation to obtain the predicted outcome and the relative explanation of the new formulation. Based on the results, users can now edit the recipe multiple times to quickly investigate slight changes in their formulation. At the end of this process, users have the option to save the simulation as a new node within the system. This way, the new node will be part of the training set and the model parameters will be updated. Figure 3 shows an example of the simulation interface for the product property "malleability".



[redacted] 4.7067
[redacted] 0.4859
[redacted] 6.5684
[redacted] 33.8671
[redacted] 35.7404
[redacted] 7.3331
[redacted] 7.3231
[redacted] 2.9052
[redacted] 1.0702

G*down array
G*up array
tanDdown array
tanDup array

Fig. 3. Example of the GUI for simulations. On the left side, the user can insert and edit a new formulation. On the right side, it can visualize the newly obtained graph and the predicted values for the desired property. The newly inserted formulation is represented by the node circled in light blue. (Color figure online)

5 Experiments

In this section, we compare our DGM-based model against five common baselines for structure learning tasks. Subsequently, we provide insights into the explanation generated on our two datasets. Finally, we validate our method on a benchmark dataset for structure learning.

Experimental Setting. We developed the DGM-based models and the *Explainer* using PyTorch Geometric (PyG)[2]. We evaluated the DGM-based model over the multi-regression task. We used the mean absolute error (MAE) and the root mean squared error (RMSE) to evaluate the prediction performance. We split the datasets into training, validation, and test sets adopting a 60/20/20 split. Consistently, we apply identical dataset divisions and training procedures across all the experiments. In all our experiments, we use the Adam [10] optimizer on the L1 loss on the training set. Hyperparameters are tuned by optimizing the loss function on the validation set, and the model parameters are randomly initialized. More information can be found in the supplementary material.

Baselines. We compare our model against five different baselines. Specifically, following previous works on structure learning [8,9], we choose an MLP that processes the tabular data as is, GCN [11] and GAT [20] using the k-NN graph construction on tabular data, and DGCNN [22], where the graph is dynamically constructed using nearest neighbors in the feature space and learned end-to-end with the supervised task. In addition, we consider a GraphTransformer [19] (GT) baseline, where the graph structure is learned using an attention mechanism over the complete graph. For a fair comparison, we use nearest neighbors with $k = 7$ for all the experiments and we adopt the same embedding dimension for all the models.

---

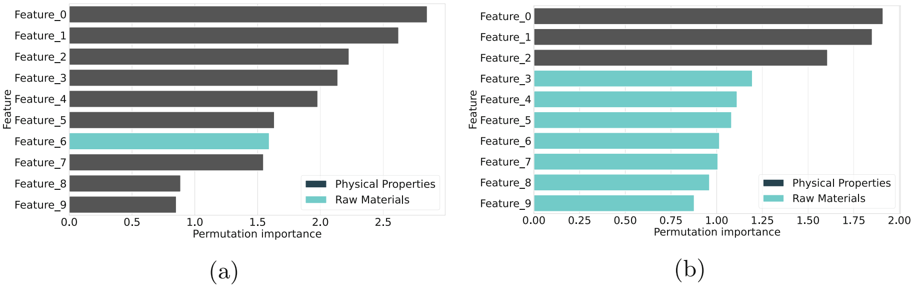
[2] https://www.pyg.org/, June 2024.

Results. We report the RMSE and MAE for "malleability" and "toughness" property prediction in Table 1. For each model, we ran experiments with five different random seeds, reporting the average result and standard deviation for each method. Our model shows better prediction performance and lower variation compared to all the other baselines. Moreover, the GraphTransformer reaches the second-best performance, highlighting that it is a strong baseline to consider in structure learning evaluation. Overall, the results show that models able to learn a latent graph structure perform better than models with a fixed k-NN graph topology. Finally, it is worth noting that an MLP, which does not take into account relationships among items, reaches comparable performances with structure learning models; however, it exhibits large variations in its results.

Table 1. Results for "malleability" and "toughness" prediction on test set using RMSE and MAE. The lower, the better. For each model, we ran experiments with five different random seeds, reporting the average result and standard deviation for each method.

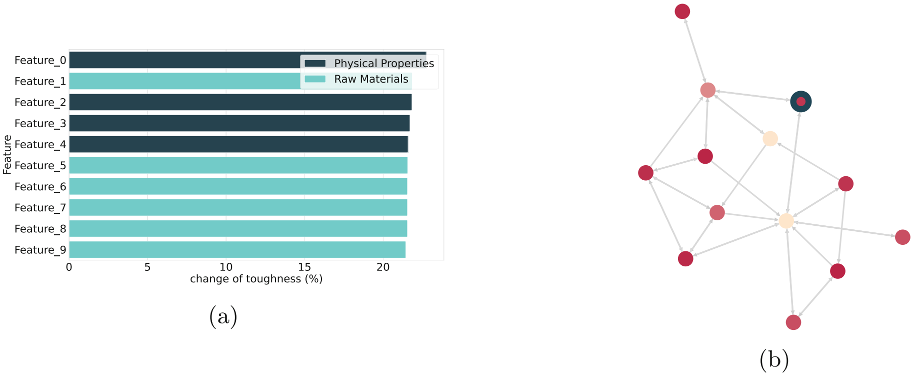
Method	Malleability		Toughness	
	RMSE	MAE	RMSE	MAE
MLP	23.40 ± 18.30	21.60 ± 18.90	28.90 ± 16.20	23.80 ± 17.80
GCN	67.60 ± 03.30	66.50 ± 03.30	70.70 ± 03.00	69.30 ± 03.20
GAT	60.50 ± 09.20	59.10 ± 09.40	62.70 ± 09.40	60.80 ± 09.50
GT	22.30 ± 07.30	20.10 ± 07.00	27.50 ± 03.40	24.80 ± 03.40
DGCNN	30.60 ± 04.70	28.90 ± 04.90	32.60 ± 05.10	30.80 ± 05.40
Our model	**14.22 ± 00.43**	**10.54 ± 00.46**	**12.48 ± 00.73**	**09.15 ± 00.29**

Feature and Subgraph Importance. We report the top ten globally important features for the property "malleability" and "toughness" in Fig. 4. Feature names are anonymized for trade secret protection. We use two different colors for raw materials and physical features. Results show that "toughness" is more influenced by the choice of ingredients while "malleability" is heavily affected by the physical properties of the final product. As an example, given the node with the highest PageRank score, we report its local explanations in Fig. 5. Nodes are colored based on their "toughness": the darker, the higher. Results allow the R&D department to identify new recipes and do the restock of raw materials. In fact, the local feature importance allows the users to understand which recipe changes have an impact on the outcome of the product, while the subgraph gives an idea of the effect of recipe changes by exploring the relationships and comparing the formulations. It is worth noting that local-level explanations are impacting the production process of the products since the company has reduced up to 30% the waste of material.



(a) (b)

Fig. 4. Global feature importance for the "malleability" (4a) and "toughness" (4b) properties. Features are anonymized for trade secret protection. Raw material and physical properties are colored in cyan and dark blue, respectively. "toughness" is more influenced by the choice of ingredients, while "malleability" is heavily affected by the physical properties of the final product. (Color figure online)



(a) (b)

Fig. 5. Local explanations for the "toughness" property of a target node. (a) The top ten locally important features for the node (anonymized for trade secret protection). (b) The important subgraph for its "toughness" prediction. The target node is circled in dark blue. Nodes are colored based on their "toughness": the darker, the higher. The local-level explanations reduce up to 30% the waste of material of the company. (Color figure online)

Ablation Study. We conducted an ablation study of our model by removing the following components: the preprocessing MLP layers, the structure learning (SL) module, and the GNN layer. We report the prediction performances for the property "toughness" in Table 2a. Results show that the graph machine learning approach - the use of structure learning and GNN layers - plays a crucial role in obtaining the best performance against the baselines. Moreover, the use of an MLP to preprocess the features is beneficial in improving the performance of the model and reducing its variability.

Table 2. (2a) Ablation study of our model considering the performance on the test set for the "toughness" property. (2b) Impact of the GNN architecture for the "toughness" property. L is the number of layers.

(a)

Method	RMSE
Our model	**12.48 ± 00.73**
Model w/o MLP	14.10 ± 03.05
Model w/o SL	28.07 ± 01.91
Model w/o GNN	28.90 ± 16.20

(b)

Method	RMSE
GAT (L=1)	13.04 ± 0.65
GCN (L=1)	12.48 ± 0.73
GCN (L=2)	12.01 ± 0.36
GCN (L=3)	12.30 ± 0.29

Choice of GNN Architecture. As described in Sect. 4, our model employs only one GCN layer to process the graph constructed by DGM. In this way, we leverage the relationships between formulations learned by the model avoiding aggregating features of too dissimilar recipes. To show the effectiveness of our choice, we compare the result of our model using one GAT layer and two or three GNN layers in Table 2b. Results show that there is no substantial gain in using more than one GCN layer and that an attention mechanism is not beneficial for the downstream task.

Model Validation. Besides our application scenario, to facilitate transparency and increase trust, we validate our model on a standard benchmark dataset for structure learning tasks in the field of healthcare and brain imaging. Specifically, we utilize the TadPole dataset [16], which contains multimodal data related to 564 patients. The task is the classification of the patients into three classes: "Normal Control", "Alzhaimer's disease" and "Mild cognitive impairment", which represent their clinical status. Each patient is represented by a 354-dimensional representation derived from imaging (MRI, fMRI, PET) and non-imaging (demographics and genotypes) features. We follow the experimental setting used in Kazi *et al.* (IA-GCN) [9] and we compare our model with the same baselines as, to the best of our knowledge, it represents the most updated and recent benchmark evaluation on the TadPole dataset. We report the results of our experiments in Table 3. Note that cDGM stands for continuous DGM, which is a model that utilizes a graph construction technique that leads to a dense network, as described in [8], and it is different from the strategy used by our solution. The results show that our model reaches the best performance against the state-of-the-art models for the task.

Table 3. Performance of our DGM-based model compared with the results presented in IA-GCN on the Tadpole dataset. Results of the baselines were taken from [9]. LC stands for Linear Classifier. cDGM stands for continuous DGM.

Method	Accuracy	AUC	F1
LC	70.22 ± 06.32	80.26 ± 04.81	68.73 ± 06.70
GCN	81.00 ± 06.40	74.70 ± 04.32	78.4 ± 06.77
GAT	81.86 ± 05.80	91.76 ± 03.71	80.90 ± 05.80
DGCNN	84.59 ± 04.33	83.56 ± 04.11	82.87 ± 04.27
cDGM	92.92 ± 02.50	97.16 ± 01.32	91.4 ± 03.32
IA-GCN	96.08 ± 02.49	98.6 ± 01.93	94.77 ± 04.05
Our model	**97.18 ± 00.72**	**99.02 ± 00.62**	**96.40 ± 00.62**

6 Conclusion

In this work, we propose a data-driven approach for fast product development, allowing companies to avoid an intensive trial-and-error iterative process for creating a new or improved product. Starting from tabular formulations of past experiments conducted by a research and development department, a machine-learning model is trained to predict the desired properties of unseen formulations. Specifically, we utilized a graph machine learning model that can simultaneously learn the prediction task and an underlying latent graph to explore similarities and complex relationships between the experiments. Subsequently, explainability techniques based on graphs, perturbation, and sensitivity analysis effectively support R&D in identifying new recipes for reaching a desired property. The constructed graph, the property predictions, their explanations, and the model at inference time are offered to the R&D department through data exploration, visualization, and editing modules in a web application. We presented a case study in the context of food design where industrial researchers are interested in finding recipes that satisfy specific sensorial properties of the product over time. We show the effectiveness of our model on two datasets derived from the case study, achieving the best performance compared to other tabular, graph, and structure learning approaches. Explainability techniques allow users to identify the most influential characteristics for obtaining a certain property, discover new recipes to test in real laboratories and do the restock of raw materials. Thanks to our solutions, the company is expected to reduce the labor-intensive experiments from 150 tests before approval to 30, the waste of materials dropped up to 30%, and the time to market has passed from seven to two months.

In future works, we plan to apply our methodology to additional case studies in several application scenarios that exhibit huge collections of past trials such as drug design or insurance stipulation.

Acknowledgments. We would like to thank the members of the PVM Lainate team for the opportunity given to us by this collaboration, which fostered the development of this novel methodology. A special thanks to Marco Violi and Maria Giovanna Esposito for their time and assistance in understanding and analyzing the data.

References

1. Material prediction for design automation using graph representation learning. In: International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, vol. Volume 3A: 48th Design Automation Conference (DAC) (08 2022). https://doi.org/10.1115/DETC2022-88049, https://doi.org/10.1115/DETC2022-88049
2. Altmann, A., Toloşi, L., Sander, O., Lengauer, T.: Permutation importance: a corrected feature importance measure. Bioinformatics **26**(10), 1340–1347 (2010). https://doi.org/10.1093/bioinformatics/btq134
3. Dileo, M., Zignani, M., Gaito, S.: Temporal graph learning for dynamic link prediction with text in online social networks. Mach. Learn. **113**(6), 1–20 (2023). https://doi.org/10.1007/s10994-023-06475-x
4. Franz, M., et al.: Cytoscape.js 2023 update: a graph theory library for visualization and analysis. Bioinformatics **39**(1), btad031 (2023)
5. Gaudelet, T., et al.: Utilizing graph machine learning within drug discovery and development. Briefings Bioinf. **22**(6), bbab159 (2021). https://doi.org/10.1093/bib/bbab159
6. Gilmer, J., Schoenholz, S.S., Riley, P.F., Vinyals, O., Dahl, G.E.: Neural message passing for quantum chemistry. In: ICML Proceedings of Machine Learning Research, vol. 70, pp. 1263–1272. PMLR (2017)
7. He, X., Deng, K., Wang, X., Li, Y., Zhang, Y., Wang, M.: LightGCN: simplifying and powering graph convolution network for recommendation. In: SIGIR, pp. 639–648. ACM (2020)
8. Kazi, A., Cosmo, L., Ahmadi, S., Navab, N., Bronstein, M.M.: Differentiable graph module (DGM) for graph convolutional networks. IEEE Trans. Pattern Anal. Mach. Intell. **45**(2), 1606–1617 (2023)
9. Kazi, A., Farghadani, S., Aganj, I., Navab, N.: IA-GCN: interpretable attention based graph convolutional network for disease prediction. In: Cao, X., Xu, X., Rekik, I., Cui, Z., Ouyang, X. (eds.) Machine Learning in Medical Imaging: 14th International Workshop, MLMI 2023, Held in Conjunction with MICCAI 2023, Vancouver, BC, Canada, October 8, 2023, Proceedings, Part I, pp. 382–392. Springer Nature Switzerland, Cham (2024). https://doi.org/10.1007/978-3-031-45673-2_38
10. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. In: Bengio, Y., LeCun, Y. (eds.) 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings (2015). http://arxiv.org/abs/1412.6980
11. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: ICLR (Poster). OpenReview.net (2017)
12. Krahe, C., Bräunche, A., Jacob, A., Stricker, N., Lanza, G.: Deep learning for automated product design. Procedia CIRP **91**, 3–8 (2020). https://doi.org/10.1016/j.procir.2020.01.135, https://www.sciencedirect.com/science/article/pii/S2212827120307769, enhancing design through the 4th Industrial Revolution Thinking

13. Langville, A.N., Meyer, C.D.: A survey of eigenvector methods for web information retrieval. SIAM Rev. **47**(1), 135–161 (2005). https://doi.org/10.1137/S0036144503424786
14. Li, C.T., Tsai, Y.C., Liao, J.C.: Graph neural networks for tabular data learning. In: 2023 IEEE 39th International Conference on Data Engineering (ICDE), pp. 3589–3592 (2023). https://doi.org/10.1109/ICDE55515.2023.00275
15. Li, F., et al.: Developing a dynamic graph network for interpretable analysis of multi-modal MRI data in parkinson's disease diagnosis. In: EMBC, pp. 1–4. IEEE (2023)
16. Marinescu, R.V., et al.: TADPOLE challenge: accurate alzheimer's disease prediction through crowdsourced forecasting of future data. Predict. Intell. Med. **11843**, 1–10 (2019)
17. Merchant, A., Batzner, S., Schoenholz, S.S., Aykol, M., Cheon, G., Cubuk, E.D.: Scaling deep learning for materials discovery. Nature **624**(7990), 80–85 (2023). https://doi.org/10.1038/s41586-023-06735-9
18. Scarselli, F., Gori, M., Tsoi, A.C., Hagenbuchner, M., Monfardini, G.: The graph neural network model. IEEE Trans. Neural Networks **20**(1), 61–80 (2009). https://doi.org/10.1109/TNN.2008.2005605
19. Shi, Y., Huang, Z., Feng, S., Zhong, H., Wang, W., Sun, Y.: Masked label prediction: Unified message passing model for semi-supervised classification. In: IJCAI, pp. 1548–1554. ijcai.org (2021)
20. Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y.: Graph attention networks. In: ICLR (Poster), OpenReview.net (2018)
21. Verdone, A., Scardapane, S., Panella, M.: Explainable spatio-temporal graph neural networks for multi-site photovoltaic energy production. Appl. Energy **353**, 122151 (2024). https://doi.org/10.1016/j.apenergy.2023.122151, https://www.sciencedirect.com/science/article/pii/S0306261923015155
22. Wang, Y., Sun, Y., Liu, Z., Sarma, S.E., Bronstein, M.M., Solomon, J.M.: Dynamic graph CNN for learning on point clouds. ACM Trans. Graph. **38**(5), 146:1–146:12 (2019)
23. Ying, Z., Bourgeois, D., You, J., Zitnik, M., Leskovec, J.: GNNExplainer: generating explanations for graph neural networks. In: NeurIPS, pp. 9240–9251 (2019)
24. Zhu, Y., et al.: A survey on graph structure learning: Progress and opportunities (2022)

HPExplorer: XAI Method to Explore the Relationship Between Hyperparameters and Model Performance

Yulia Grushetskaya(✉), Mike Sips, Reyko Schachtschneider,
Mohammadmehdi Saberioon, and Akram Mahan

GFZ German Research Centre for Geosciences, 14473 Potsdam, Germany
`{yulia.grushetskaya,sips,reyko.schachtschneider,saberioon,amahan}@gfz-potsdam.de`

Abstract. Hyperparameters (HPs) play a central role in the performance of machine learning (ML) models, governing model structure, regularization, and convergence properties. Understanding the intricate relationship between HP configurations and model performance is essential for ML practitioners, especially those with limited expertise, to develop effective models that produce satisfactory results. This paper introduces HyperParameter Explorer (HPExplorer), a semi-automated eXplainable AI (XAI) method, to support ML practitioners to explore this relationship. HPExplorer integrates an automated HP discovery algorithm with an interactive visual exploration component. The HP discovery algorithm identifies performance-consistent subspaces within the HP space, where models perform similarly despite minor variations in HP configurations. The interactive visual exploration component enables users to explore the discovered performance-consistent subspaces using an interactive 2-D projection called Star Coordinate. Users can also compare HP configurations from different subspaces to explore their impact on model performance. We developed HPExplorer in close collaboration with ML practitioners, particularly geoscientists, using ML in their research. Initial feedback from scientists using HPExplorer in real-world scenarios indicates that HPExploer enhances the transparency in configuring HPs and increases the confidence of users in their decisions.

Keywords: Explainable Artificial Intelligence (XAI) · Hyperparameter Space Exploration · Visual Analytics · Real-World Applications

1 Introduction

Explainable Artificial Intelligence (XAI) is increasingly essential in fostering widespread acceptance and utilization of artificial intelligence (AI) systems in our daily lives. A significant challenge in XAI is understanding how hyperparameter (HP) configurations affect machine learning (ML) model performance, as HPs significantly specify the model's architecture, regularization strategies,

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
A. Bifet et al. (Eds.): ECML PKDD 2024, LNAI 14949, pp. 319–334, 2024.
https://doi.org/10.1007/978-3-031-70378-2_20

and convergence properties. However, understanding the relationship between HP configurations and model performance can be challenging for humans due to its complex, multivariate nature. Human cognition is typically effective at understanding functional relationships with up to two variables. This cognitive limitation poses a significant hurdle for users of ML models, particularly those utilizing ML in their respective scenarios with limited ML expertise (referred to as ML practitioners), hindering their ability to discern how specific choices of HP configurations influence their model's performance.

As a result, the ML research community has developed several automated search techniques [1–3] to help users find appropriate HP configurations for their models. Because users are reluctant to use methods they do not fully understand there is a growing need for transparency in these techniques. Automated search techniques provide only a few "good" results without explaining how different HP configurations along the search path affect model performance. We developed HPExplorer to address the challenge of ML practitioners in understanding the relationship between HP configurations and model performance. HPExplorer leverages the fundamental Visual Analytics (VA) concept of seamlessly integrating an HP discovery algorithm with interactive visual exploration. The HP discovery algorithm uncovers subspaces within the HP space where model performance remains consistent despite minor variations in HP configurations. We call these subspaces "performance-consistent subspaces." The interactive visual exploration component enables users to explore the performance-consistent subspaces with HPExplorer. Combining the automated HP discovery algorithm with interactive visual exploration provides a semi-automated exploration system for ML practitioners to explore the relationship between HP configurations and model performance. Our semi-automated exploration system draws inspiration from Asimov's Grand Tour concept [17], proposing a systematic traversal of multidimensional data spaces to capture potentially interesting structures.

HPExplorer provides two main interactive visualization methods for exploring the results of the HP discovery algorithm. The first visualization method, Star Coordinate plot, provides a comprehensive overview of the discovered performance-consistent subspaces. Star Coordinate offers an intuitive projection of the performance-consistent subspaces onto the 2D screen and aids users in visually comparing the shapes and sizes of performance-consistent subspaces, facilitating the identification of similarities and differences between them. Users can interactively manipulate the projection of the performance-consistent subspaces onto the 2D screen, fostering hypothesis generation and testing regarding the relationship between HP configurations and model performance. The second visualization method, Stacked Scatterplot, enables users to compare HP configurations from different performance-consistent subspaces and comprehend their impact on model performance.

Throughout HPExplorer's development, we prioritized user involvement. By collaborating closely with ML practitioners and incorporating their feedback, we translated their analytical needs into requirements that were subsequently addressed in HPExplorer. We discuss the advantages and disadvantages of

HPExplorer in three real-world scenarios based on initial feedback from users. These include mortality prediction in climate change scenarios, analyzing carbon concentration in soil, and flood prediction in natural hazard scenarios.

2 Related Work

We briefly discuss the current research in machine learning (ML), focusing on areas related to HPExplorer, including hyperparameter optimization (HPO), explainable AI (XAI), and Visual Analytics (VA).

2.1 HP Optimization in ML Research

Identifying the best hyperparameters (HPs) for machine learning (ML) models is challenging, particularly for practitioners with limited ML expertise. Automated search techniques, such as Grid Search, Bayesian Optimization, and Genetic Algorithms, have become popular solutions. However, these methods often return a "good" HP configuration without providing insights into how specific configurations were selected. Additionally, the complexity of some methods can obscure the HP search process, turning it into a black box for users.

Our approach differs by helping users to understand the impact of HP configurations on model performance. By exploring the HP space with HPExplorer, users can differentiate between HP configurations that yield "high," "average," or "low" model performance. Thus, HPExplorer may serve as a valuable tool for ML practitioners who want to assess and refine the results of automated search techniques.

2.2 XAI Research

The field of Explainable AI (XAI) aims to elucidate the inner workings of AI systems. While much XAI research focuses on explaining specific model decisions or overall model behavior, there is a growing need to understand how HP configurations affect model mechanisms and performance [5,6]. This need is driven by users' demands for transparency at this stage of the ML process [4]. An emerging XAI topic relevant to our research is the explanation of decisions and search paths in HPO algorithms. For example, the RX-BO [9] method employs a rule-based approach to explain Bayesian Optimization (BO) decisions, automatically generating distribution-aware rules to clarify the HPO process. In contrast, XAI approaches like VisualHyperTuner [7] and VisEvol [8] actively involve users in the HPO process through a human-in-the-loop approach. These systems use visual analytics (VA) concepts and methods to enhance user understanding and control over HPO algorithms. Additionally, methods like HyperTendril [6] and XAutoML [10] integrate HPO explanations into AutoML pipelines using visual techniques.

Many existing XAI methods focus on enhancing the transparency of decisions and search paths in HPO algorithms. They provide local explanations of how HP configurations affect model performance along the HPO search path. However, few address the broader challenge of explaining the global relationship between HP configurations and model performance. Our innovative approach enables users to explore performance-consistent subspaces within the HP space, offering diverse visual perspectives of these subspaces. Our approach goes beyond merely visualizing decisions made along a search path; it helps users understand the broader landscape of HP configurations and their impact on model performance.

Another XAI approach to making HP selection more interpretable is to evaluate the importance of different HPs based on model performance. For example, ANOVA-based methods [24] identify HPs with statistically significant effects, while ablation studies [25] systematically remove components to assess their impact on performance. Although ablation studies provide detailed insights, they often overlook less significant HPs. HPExplorer uses ANOVA to gain initial insights into HPs, guiding the exploration of HP configurations.

2.3 Visual Analytics Research

VA combines automated analysis methods (data mining) with interactive visualizations, enabling humans to analyze complex data effectively (i.e., adopting human-in-the-loop concept for data analysis scenarios [20]). Scientists increasingly use VA concepts and methods in ML projects to understand, for instance, the internal mechanics of deep learning models [28]. Many XAI methods focusing on HPO [6–8] use VA concepts and methods to depict explanations about the decisions and search paths of HPO algorithms to users.

Our research closely relates to the family of multidimensional projections and multivariate visualization methods. To provide users with a concise visual overview of performance-consistent subspaces in the HP space, we decided to use a multidimensional projection with a radial 2-D layout. Multidimensional projections using radial layouts, such as Star Coordinate [11] or RadViz [12], arrange the dimensions of high-dimensional data space around a circle and position data points within the circle based on their values in each dimension. We specifically decided to use Star Coordinate because it effectively represents the global distribution of HPs across the HP space and supports users in identifying performance-consistent subspaces. Users can interactively explore the performance-consistent subspaces by manipulating the projection of the HP space on the 2D screen by adjusting axes along the circular layout or hiding axes.

Another widely used VA technique in HPO-related research is the parallel coordinate plot (PCP) [13]. PCP utilizes point-line duality to depict high-dimensional data points [21] to users by representing each dimension as a vertical axis and data points as lines traversing through these axes at their respective values in each dimension. PCP is commonly employed in HPO approaches to depict the HP configurations traversed by HPO algorithms to users. The strength of PCP lies in its ability to facilitate direct comparisons of HP configurations. However, for our specific purpose of comparing HP configurations based on model performance, we decided to use a Stacked Scatterplot. In this visualization, the x-axis represents the user's performance metric, allowing for a clear separation of HP configurations based on their model performance. Unlike PCP, which does not visually separate HPs leading to different model performances, the Stacked Scatterplot offers a more intuitive comparison of HPs and their impact on model performance.

3 Analytical Needs and Requirements for HPExplorer

We worked closely with ML practitioners to understand their challenges in exploring the relationship between HP configurations and model performance. We translated the abstract feedback from ML practitioners about their challenges into precise analytical requirements for the HPExplorer.

3.1 Analytical Needs of Users

We identified the following analytical needs of ML practitioners:

N1. Automated discovery of performance-consistent subspaces: ML practitioners need an automated algorithm to discover performance-consistent subspaces of the HP space. ML practitioners often emphasize the importance of focusing on regions within the HP space where model performance remains stable despite minor variations in HP configurations. Focusing on performance-consistent subspaces provides valuable insights into which HP configurations lead to desirable results.

N2. Exploration of performance-consistent subspaces through an interactive 2D projection of the HP space: ML practitioners expressed the need to explore performance-consistent subspaces via an interactive 2D projection of the HP space. ML practitioners want to interactively manipulate the projection of the HP space to explore performance-consistent subspaces from different angles.

N3. Assessment and comparison of HP configurations and their impact on model performance: ML practitioners emphasized the need to assess and compare different HP configurations to understand their influence on model performance and how variations in HP configurations affect model performance.

N4. Exploration of the importance of individual HPs: ML practitioners have expressed the need to explore the impact of individual HP configurations on the model's overall performance.

3.2 Requirements for HPExplorer

From the needs N1, N2, N3 and N4 of ML practitioners, we derived the following requirements for HPExplorer:

R1. Semi-automated exploration of performance-consistent subspaces: Manual exploration of performance-consistent subspaces within the HP space is impractical for users. HPExplorer seamlessly integrate an automated discovery algorithm (N1) with an interactive visual exploration component (N2) to support the semi-automated exploration of performance-consistent subspaces.

R2. Visual inclusion of inconsistent subspaces in the 2-D projection of the HP space: Users emphasize the importance of providing a complete overview of the HP space. HPExplorer includes performance-consistent and inconsistent subspaces into the 2-D projection of the HP space (N2) to ensure an informative and accurate visual exploration of the HP space.

R3. Enable comparison of individual HPs: HPExplorer should allow for an in-depth comparison of HP configurations, focusing on exploring the influence HP configurations on model performance and how variations in HP configurations affect model performance (N3).

R4. Support exploration of importance scores and ensure transparency of score calculation: Users want to explore the importance of HP configurations on overall model performance while understanding the algorithmic calculation of the importance scores. HPExplorer visually explain the essential steps of the score calculation algorithm (N4).

4 HPExplorer

HPExplorer integrates two components, a HP discovery algorithm and an interactive visual exploration component, into a VA approach that fulfill requirements R1 - R4. The HP discovery algorithm focuses on finding performance-consistent subspaces within the HP space. This algorithm is complemented by an interactive visual exploration component, which allows users to examine and comprehend the results of automated discovery algorithm.

4.1 HP Discovery Algorithm

The HP discovery algorithm addresses R1 and traverses HP subspaces by randomly selecting HP configurations, termed “anchors”, to test model performance. These anchors pinpoint different HP subspaces and act as centers for further exploration. Then, for each anchor, we examine neighboring HPs for performance consistency. We use a simple, cost-effective random sampling method with a user-configurable number of samples to ensure accessibility without extensive computational resources. HPExplorer’s design allows the integration of alternative sampling methods. The developed algorithm involves the following computational steps:

Anchor Samples Generation. We begin by generating a set of HP configurations $[\theta_1^a, \theta_2^a, ..., \theta_n^a]$ that we call “anchor” samples. Each anchor θ_i^a is a point in the user-defined HP space Θ. This process is conducted through Sobol sampling, creating quasi-random sequences across a multidimensional space [18]. Distinct from traditional random sampling, Sobol sequences, based on powers of two, progressively divide the unit interval into finer, uniform segments, thus achieving a more evenly dispersed parameter space coverage [19].

Subspace Sampling. For each Anchor sample θ_i^a, we define a subspace $R_{\theta_i^a}$ within Θ using a distance threshold δ. Gower distance calculates dissimilarity across data types by averaging normalized differences for numerical variables and evaluating matches/mismatches for categorical variables, ranging from 0 (identical) to 1 (entirely different). A subspace $R_{\theta_i^a}$ includes all points within the user-defined distance threshold from the anchor θ_i^a. In each subspace $R_{\theta_i^a}$, Sobol sampling generates neighbor samples $[\theta_{i_1}^a, \theta_{i_2}^a, ..., \theta_{i_m}^a]$, representing different HP configurations within the subspace of the anchor sample θ_i^a.

Performance Testing. We perform an evaluation (referred to as a *Trial*) for all generated HP configurations θ_i, encompassing both anchors and neighbors samples. Each *Trial* includes training the model M_i with the training dataset D_{train} using $M_i = \text{Train}(\theta_i, D_{\text{train}})$, then evaluating M_i on the test dataset D_{test} to obtain the performance score s_i through $s_i = \text{Eval}(M_i, D_{\text{test}})$. Each *Trial* outputs a tuple (s_i, θ_i), linking performance scores with HP configurations.

Subspace Consistency Evaluation. For each subspace, we compute the mean difference in performance scores between the anchor θ_i^a and its neighbors $[\theta_{i_1}^a, \ldots, \theta_{i_m}^a]$ as $\mu_i = \frac{1}{n} \sum_{\theta_{i_j}^a} \left| s(\theta_i^a) - s(\theta_{i_j}^a) \right|$, where $s(\theta)$ is the performance score of θ. Each subspace $R_{\theta_i}^a$ is classified based on a density threshold α. If $\mu_i \leq \alpha$, it is 'performance-consistent'; otherwise, it is 'non-performance-consistent'.

4.2 Interactive Visual Exploration Component

We use visualization and interaction methods from VA to address R1 - R4. The interactive visual exploration component enables users to explore the outcomes of the the HP discovery algorithm.

To address R1, HPExplorer utilizes the Star Coordinate plot to provide an overview of the HP space and its performance-consistent subspaces (Fig. 1 Part A). The Star Coordinate plot depicts the distribution of HP configurations generated by the HP exploration algorithm to users. Performance-consistent subspaces are depicted as visual clusters of HP configurations within the plot (i.e., HP configurations from a common subspace are close to each other in the Star Coordinate plot), enabling users to visually identify and explore these subspaces.

In response to the need for interactivity (R1), users can leverage the Star Coordinate plot's interactive axis management feature to explore performance-consistent subspaces from various perspectives. This feature allows users to manipulate the 2D projection by rearranging or removing axes from the 2-D projection, aiding in better understanding the position of HP configurations in the multidimensional HP space. Moreover, it enables users to focus their analysis on key HPs or tailor the projection according to their analytical needs. Additionally, in adherence to R2, HP configurations from inconsistent subspaces - characterized by a limited number of samples with significant performance deviations from each other - are also depicted in the Star Coordinate plot to provide users with a comprehensive 2-D visual representation of the HP space.

To address R4, HPExplorer utilizes a circular bar chart to depict the importance scores of individual HPs (refer to Fig. 1 Part B). Although comparing bar

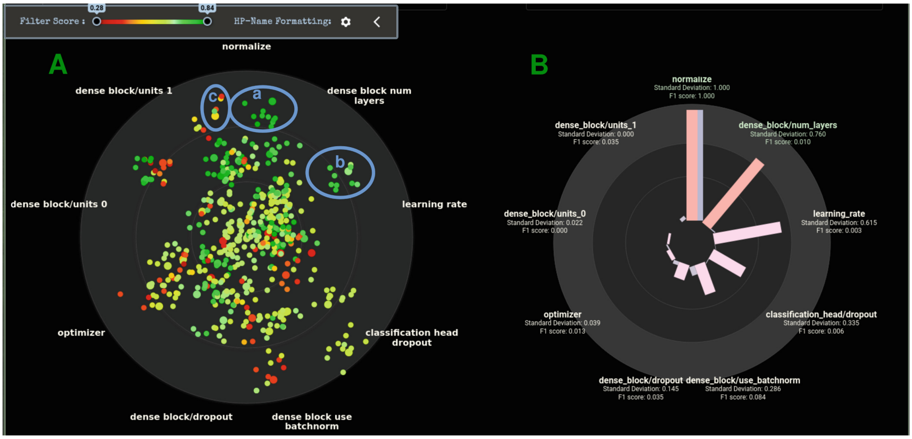


Fig. 1. Interactive Exploration Components in HPExplorer: A) **Star Coordinates Plot:** This visualization offers a global view of the HP space. Star coordinates depict HP configurations to users by projecting the HP space onto a 2D plane using a star-shaped layout where the axes radiate outward from the center. In HP Explorer, each axis is an HP, the points in the projection represent HP configurations, and the color represents model performance. Dense point clouds (a, b) indicate performance-consistent subspaces, while sparse points (c) highlight non-consistent subspaces. Circles around the point clouds, added manually for demonstration purposes, emphasize these subspaces. B) **Radial Bar Chart:** This chart displays the importance score and f-statistic of each HP as bars around the center in clockwise direction.

heights in circular layouts poses a common challenge, the rationale for employing a circular bar chart in our context is clear. The synchronization of the circular bar chart axes with the Star Coordinates axes ensures a symmetrical and coherent visual depiction of the HP space, enhancing the readability of the Star Coordinate plot and the importance scores. Furthermore, the arrangement of HP axes in descending order of importance score within the Star Coordinate plot offers a logical starting point for interactive exploration of the HP space. This layout of the two interactive visualization methods facilitates users' understanding of the relative significance of HPs and guides their exploration process. Additionally, we provide detailed insights in calculating importance scores of HPs through numerous small scatter plots. These scatter plots provide comprehensive information about calculating the importance scores.

The utilization of a Stacked Scatterplot in HPExplorer supports users in comparing HP configurations in detail, fulfilling the requirement R3 (Fig. 2). The Stacked Scatterplot depicts the HP configurations along x-axis representing the user's performance metric. This comprehensive view provides insights into the specific contribution of each HP configuration and their differences to other HP configurations on model success or failure. Statistical annotations, including maximum, mean values, and success ratios, increases the readability of the Stacked Scatterplot.

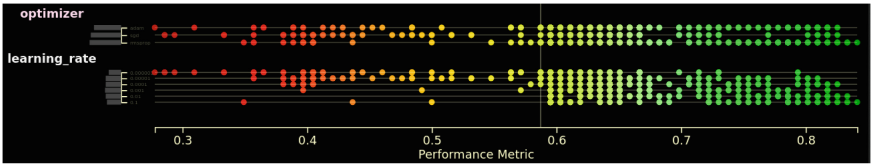


Fig. 2. Interactive Exploration Components in HPExplorer: Stacked Scatterplot This visualization facilitates the comparison of HP configurations and explores their effect on model performance. Each HP configuration is represented as a point, with the Y-axis indicating the HP value and the X-axis representing the model's performance with that HP configuration.

5 Real-World Applications of HPExplorer

This section aims to provide preliminary feedback from users to discuss the advantages and limitations of HPExplorer rather than an exhaustive evaluation of our approach.

We integrated HPExplorer into the research workflow of geoscientists to help them find appropriate HP configurations for their ML models. The geoscientists used HPExplorer to explore performance-consistent subspaces to decide on HP configurations. The first scenario presents a detailed discussion of using HPExplorer to predict mortality rates for climate change scenarios. The subsequent two use cases, such as soil organic carbon prediction and flood prediction, provide concise summaries of HPExplorer's utility and ML practitioners' challenges when using HPExplorer in these scenarios.

For the experiments described, we used default settings of 100 anchor samples and 10 neighbour samples. For the majority of use cases, a single experiment run sufficed. However, one scenario required a rerun with a reduced HP space. Experiment run time varied based on configurations and model complexity, but progress and model performance for HP configurations could be tracked in real-time.

5.1 Use Case 1: Mortality Rate Prediction

In this use case an Echo State Network (ESN) was used for the prediction of mortality rates in Germany under different climatic conditions. The ESN is a method of reservoir computing that is capable of imitating complex dynamical processes [29,30].

In this study, the ESN was used to train a connection between maps of monthly maximum temperatures over Europe and mortality rates in Germany. The training input data origin from climate storyline simulations. In those simulations, extreme weather events are simulated in different states of the climate system, i.e. under different atmospheric forcing conditions [31]. For the network training, temperature field outputs of simulations under current atmospheric conditions are used. The target data are all-cause mortality data for all ages in

Germany. After successful training, temperature field outputs of simulations with future atmospheric conditions (i.e. 2 K and 4 K above pre-industrial level) are used to make predictions of how mortality rates would develop in such warmer worlds.

The researcher used HPExplorer to explore performance-consistent regions within the HP space and identify appropriate HP configurations for the ESN model. Initial analysis (Fig. 3, Panel B) showed treservoir size and leakage parameters significantly impact model performance, while other parameters had a minor effect. The Star Coordinate plot (Fig. 3, Panel A) further revealed two performance-consistent subspaces (labeled (a) and (b)) where specific HP combinations consistently yielded favorable results. Close analysis of these subspaces highlighted numerous appropriate HP configurations for the ESN model.

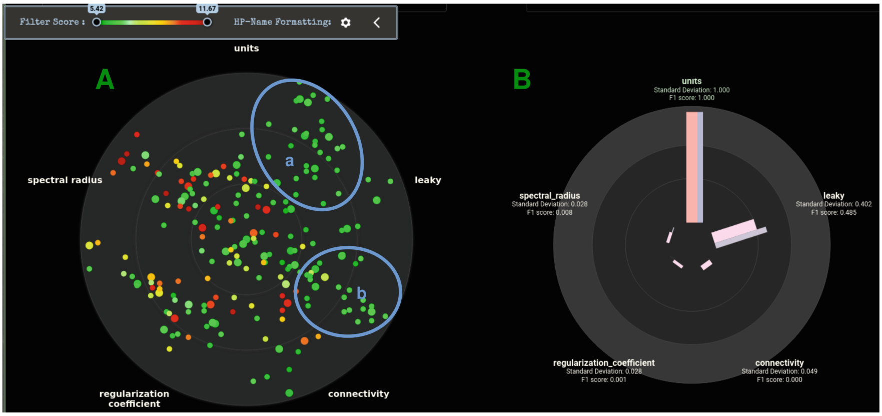


Fig. 3. Exploring performance-consistent regions: The Radial Bar Chart (B) on the right shows that the HPs "number of units" and "leakage" have the greatest impact on model performance. The Star Coordinate plot (A) on the left reveals two performance-consistent subspaces, labeled (a) and (b), where HP combinations consistently yield low error values. (a) corresponds to HP configuration with considerable extensive leakage and reservoir size values, while (b) corresponds to large leakage values, medium reservoir size values, and significant connectivity with the ESN.

To further investigate the relationships between the HPs and model performance, the researcher conducted an in-depth analysis using Stacked Scatterplots, as shown in Fig. 4. This visualization technique allowed the researcher to examine the variation in model performance (indicated by color and position along the x-axis) for different values of different HPs along the y-axis. The Stacked Scatterplot revealed that low leakage values yielded a mix of high and low error values, while high leakage values predominantly resulted in low error values. Conversely, when examining the spectral radius, the researcher observed a mix of favorable and unfavorable outcomes across all its values, aligning with the low significance of this HP in the circular bar chart on 3 (panel B).

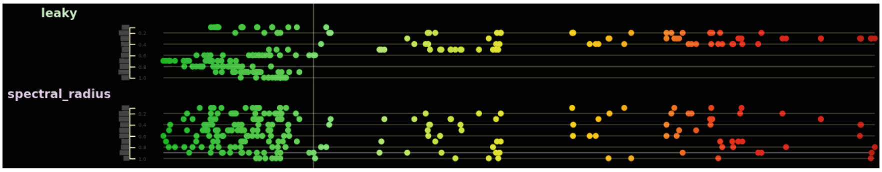


Fig. 4. Exploring performance-consistent regions: Researcher used stacked Scatterplot for examining the effects of leakage and spectral radius on model performance. The upper plot shows a clear interaction between the two HPs: lower leakage increases errors, whereas higher leakage enhances performance, emphasizing leakage's significant role. The lower plot indicates that spectral radius has little influence on errors, with no apparent pattern across the whole interval of spectral radius, highlighting its small impact on model performance.

Based on the researcher feedback, HPExplorer provided a structured and less time-consuming approach for setting HPs than manual configuration and testing. It facilitated targeted exploration of significant HPs to determine their optimal ranges, setting non-essential HPs to default values. The exploration results motivated the researchers to conduct more tests and study the impact of the leakage HP for ESNs for gaining a deeper understanding of the impact of HPs in this specific use case.

5.2 Use Case 2: Soil Organic Carbon Prediction

The scenario involved exploring HP configurations for Convolutional Neural Network (CNN) models to predict soil organic carbon (SOC) levels using field spectra data. SOC, a primary terrestrial carbon (C) component, plays a pivotal role in mitigating climate change by absorbing and sequestering carbon dioxide (CO_2). The proliferation of soil spectral libraries (SSLs) at regional, continental, and global scales presents a notable opportunity for SOC quantification through spectral-based prediction models. With the expansion of extensive SSLs, there is a growing need to employ ML models for spectral data processing.

HPExplorer showed that usage of an autoencoder for input dimensionality reduction and optimizer selection significantly impact model performance. The performance gap between models with and without autoencoders complicated the analysis of other HPs, leading to a second experiment where all HP combinations was tested only with autoencoders. The exploration of the HP space then highlighted the importance of the dropout rate, pinpointing the 0.1 to 0.5 interval as suitable for achieving optimal performance. In addition, HPExplorer exploration revealed that incorporating a CNN layer and one or two Fully Connected layers leads to superior results. These results are consistent with the optimal architecture identified through previous manual searches and various optimization methods.

Researcher feedback indicates that HPExplorer enabled a thorough exploration of the relationship between HP configurations and model performance.

This deeper understanding enabled the researchers to make informed decisions based on empirical evidence. HPExplorer's results were consistent with those from manual search and optimization techniques. In addition, the researchers appreciated HPExplorer's structured approach and increased transparency compared to manual selection and automated methods. The researcher also suggested potential improvements. Introducing a dynamic filtering feature would allow users to isolate results by specific HPs, such as "use autoencoder" or kernel size range, without additional experimental runs. This improvement would enrich HPExplorer's functionality and save considerable time and computational resources.

5.3 Use Case 3: Flood Detection

This use case for HPExplorer centers around examining flood prediction methods by comparing point-based versus image-based ML models.

Due to the grave consequences of floods, scientists are working to anticipate and mitigate hazards in flood-prone areas. The study focuses on the potential of image-based models compared to point-based models, which, by integrating spatial context, are hypothesized to offer enhanced predictive accuracy on unseen data from diverse geographical regions. The methodology encompasses ten critical factors likely to influence flood occurrence, encompassing topographic, hydrological, and geological dimensions. A dataset for this study was built based on flood inventory, including 2500 documented flood locations and an equivalent number of non-flooded locations.

The initial step in the research involved establishing a baseline for comparison. A CNN model was chosen to represent image-based models, compared with point-based models and other image-based models. HP Explorer was instrumental in ensuring the CNN model operates within its optimal performance range. Insights from HPExplorer underscored the critical role of kernel size, dropout rates, and normalization types in model quality and stability. High dropout rates compromise performance confidence while pooling layers, and the number of filters has relatively minor impacts. Models with one or two CNN layers outperform those with three or more. Using HP Explorer enabled the researcher to gain confidence that the selected HP configurations fall within an appropriate performance range, establishing a reliable benchmark for subsequent analyses.

According to the researcher's feedback, determining the appropriate number of subspaces for an experiment in HPExplorer can be challenging. Using too few subspaces may not fully capture the complex relationship between HPs and model performance, while using too many can prolong experimentation and complicate visual exploration with Star Coordinate plots.

6 Discussion

User feedback emphasized the effectiveness of the Star Coordinate plot in identifying performance-consistent subspaces, which are depicted as distinct visual clusters in the 2D projection of the HP space. Users appreciated Star Coordinates as a cornerstone of the interactive visual exploration component for providing an overview of HP configurations and guiding exploration priorities. The detailed comparison facilitated by Stacked Scatterplots, including histograms, aided users in specifying parameter values for individual HPs. Interactive features like manipulating the projection and HP value filtering enhanced the exploration process, offering users a tailored experience. While the Star Coordinate plot provided informative overviews, it introduced visual ambiguities such as other projection methods. The ambiguities challenged users to interpret HP values accurately. Despite this, users found that the plot correctly represented the HP space on the 2D screen.

Some scenarios revealed limitations in HPExplorer's ability to identify performance-consistent subspaces, particularly with imbalanced or noisy datasets. Users emphasized the need to fine-tune the HP algorithm's parameters with specific configurations, leading to high computational demands. Suggestions for future research include developing a parameter-free HP discovery algorithm or enhancing sampling strategies for improved initial results and computational efficiency. The parameter-free discovery algorithm could enable users to directly specify subspaces for sampling through the Star Coordinate plot. It also helps users when performance-consistent subspaces are absent in the discovery result because users frequently advocated for an interactive sampling directly in the StarCoordinate plot.

While the Star Coordinate plot offers an informative overview of HP configurations and aids in exploring performance-consistent subspaces, it introduces ambiguities in visual representation like other projection methods. The plot arranges data points from minimum to maximum value in one half of the circle and from maximum to minimum value in the other half, creating ambiguity in mapping and resulting in discontinuous 'jumps' in the 2D projection. This ambiguity in the Star Coordinate's mapping can challenging users in accurately interpreting HP values from the plot. Also, subspaces far away in the HP space are close to each other in the projection. The feedback from users indicates that despite the ambiguity, the Star Coordinate plot always presents a consistent visual representation of the HP space. The feedback also indicates that HPExplorer needs to extend the current interaction capabilities to support users in dealing with ambiguities, such as depicting the interval of HP values in the Star Coordinate plot.

We also observed instances where our HPExploration algorithm failed to identify performance-consistent subspaces, particularly with imbalanced, high-variance, or noisy datasets. In such cases, users found it necessary to fine-tune the algorithm's parameters. User feedback underscores the significant impact of these parameters, such as the number of epochs and anchor configurations, on exploration effectiveness. However, specific configurations resulted in extensive computational demands, prompting suggestions for future research directions. Users propose developing a parameter-free HPExploration algorithm to better integrate with the interactive visual exploration component. Alternatively, enhancing the sampling strategy could yield improved initial results for the interactive visualization and enhance computational efficiency. Additionally, it is noteworthy that the relationship between HP configuration and model performance does not have performance-consistent subspaces in some rare cases. In such cases, users would like to sample the HP space interactively to ensure they get all the essential information.

7 Conclusion

We developed HPExplorer in close collaboration with ML practitioners by translating the analytical needs of ML practitioners into requirements for HPExplorer. We addressed these requirements with close feedback from ML practitioners. We decided to discover performance-consistent subspaces through an automated discovery algorithm and depict the discovered performance-consistent subspaces using Star Coordinate projection. HPExplorer supports comparisons of HP configuration using Stacked Scatterplots. The feedback from HPExplorer users indicates that although the Star Coordinate plot provides a comprehensive overview of HP space and assists in navigating performance-consistent subspaces, interpreting it can be complex and requires further work on its interpretability and interactive features. The Stacked Scatterplot emerged as particularly useful, enabling straightforward comparisons of data point distributions and patterns among various HP configurations.

We developed HPExplorer as part of the XAI platform ClarifAI, an open-source project that develops XAI methods to make ML models more accessible for scientists. Our collaboration with ML practitioners, particularly geoscientists, underscores our commitment to developing user-friendly tools that address the specific needs and challenges faced by ML practitioners in the real world. HPExplorer is available on our public GitLab repository[1], allowing the ML community to access, use, and contribute to the project.

The initial user feedback indicates that the HPExplorer tool shows great promise in combining automated analysis algorithms with interactive visual exploration. This integration provides a transparent and intuitive process for exploring the HP space, helping users make well-informed decisions about HP configurations for their models. Insights from collaborating with geoscientists underscore the value of VA concepts for XAI and highlight the need for our ongoing development of interactive XAI methods. Addressing identified limitations and enhancing XAI methods will be crucial, particularly in ensuring low-barrier access to XAI for a broad audience. Our next research step will employ a rigorous methodology and systematic data analysis to evaluate users' experiences and interactions with HPExplorer.

Acknowledgments. We would like to express our gratitude to Daniel Eggert and Peter Morstein for their assistance in implementing HPExplorer. Aviad Etzion helps us in integrating HPExplorer into the real-world scenarios.

---

[1] https://git.gfz-potsdam.de/xai/clarifai.

References

1. Bergstra, J., Bengio, Y.: Random search for hyper-parameter optimization. J. Mach. Learn. Res. **13**(10), 281–305 (2012)
2. Snoek, J., Larochelle, H., Adams, R.P.: Practical bayesian optimization of machine learning algorithms. In: Advances in Neural Information Processing Systems, vol. 25, pp. 1–2. Curran Associates, Inc., Red Hook (2012)
3. Young, S.R., Rose, D.C., Karnowski, T.P., Lim, S.-H., Patton, R.M.: Optimizing deep learning hyper-parameters through an evolutionary algorithm. In: Proceedings of the Workshop on Machine Learning in High-Performance Computing Environments, pp. 1–5. ACM, New York (2015)
4. Dhanorkar, S., Wolf, C.T., Qian, K., Xu, A., Popa, L., Li, Y.: Who needs to know what, when?: Broadening the Explainable AI (XAI) Design Space by Looking at Explanations Across the AI Lifecycle. In: Proceedings of the Conference, pp. 1591–1602. ACM, New York (2021)
5. Moosbauer, J., Herbinger, J., Casalicchio, G., Lindauer, M., Bischl, B.: Towards Explaining Hyperparameter Optimization via Partial Dependence Plots. In: Proceedings of the international workshop on Automated Machine Learning (AutoML) at ICML'21, pp. 1–2. Publisher, Location (2021)
6. Park, H., Nam, Y., Kim, J., Choo, J.: HyperTendril: visual analytics for user-driven hyperparameter optimization of deep neural networks. IEEE Trans. Visual Comput. Graphics **27**(2), 1407–1416 (2021)
7. Park, H., et al.: VisualHyperTuner: visual analytics for user-driven hyperparameter tuning of deep neural networks. In: Proceedings of the Conference, pp. 1–2. Publisher, Location (2019)
8. Chatzimparmpas, A., Martins, R.M., Kucher, K., Kerren, A.: VisEvol: visual analytics to support hyperparameter search through evolutionary optimization. Comput. Graph. Forum **40**(3), 69–91 (2021)
9. Chakraborty, T., Wirth, C., Seifert, C.: Post-hoc rule based explanations for black box bayesian optimization. In: Editors (eds.) Book Title, pp. 320-337. Springer, Location (2024)
10. Zöller, M.-A., Titov, W., Schlegel, T., Huber, M.F.: XAutoML: a visual analytics tool for understanding and validating automated machine learning. ACM Trans. Interactive Intell. Syst. **13**(4), 1–39 (2023)
11. Kandogan, E.: Visualizing multi-dimensional clusters, trends, and outliers using Star Coordinates. In: Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 107–116. ACM, New York (2001)
12. Hoffman, P., Grinstein, G., Marx, K., Grosse, I., Stanley, H.: DNA visual and analytic data mining. In: Proceedings of the Conference, pp. 437–442. Publisher, Location (1997)
13. Inselberg, A.: The plane with parallel coordinates. Vis. Comput. **1**, 69–91 (1985)
14. Tufte, E.R.: The Visual Display of Quantitative Information. Graphics Press, USA (1986)
15. Etemadpour, R., Linsen, L., Paiva, J.G., Crick, C., Forbes, A.: Choosing Visualization Techniques for Multidimensional Data Projection Tasks: A Guideline with Examples. In: Editors (eds.) Book Title, vol. 598, pp. 166–186. Springer, Location (2016)
16. Dzemyda, G., Kurasova, O., Zilinskas, J.: Multidimensional data visualization. Methods and applications series: Springer optimization and its applications **75**(122), 10–5555 (2013)

17. Asimov, D.: The grand tour: a tool for viewing multidimensional data. SIAM J. Sci. Stat. Comput. **6**(1), 128–143 (1985)
18. Renardy, M., Joslyn, L.R., Millar, J.A., Kirschner, D.E.: To sobol or not to sobol? the effects of sampling schemes in systems biology applications. Math. Biosci. **337**, 108593 (2021)
19. Burhenne, S., Jacob, D., Henze, G.P.: Sampling based on Sobol' sequences for Monte Carlo techniques applied to building simulations. In: Proceedings of the Conference, pp. 1816–1823. Publisher, Location (2011)
20. HILDA '23: Proceedings of the Workshop on Human-In-the-Loop Data Analytics. ACM, New York, NY, USA (2023)
21. Inselberg, A.: Parallel Coordinates: Visual Multidimensional Geometry and Its Applications. Springer, Heidelberg (2017)
22. Decker, T., Gross, R., Koebler, A., Lebacher, M., Schnitzer, R., Weber, S.H.: The thousand faces of explainable AI along the machine learning life cycle: industrial reality and current state of research. In: AI-HCI 2023: Artificial Intelligence in HCI, pp. 184-208. Springer, Heidelberg (2023)
23. Weidele, D.K.I., Weisz, J.D., Oduor, E., Muller, M., Andres, J., Gray, A., Wang, D.: AutoAIViz: opening the blackbox of automated artificial intelligence with conditional parallel coordinates. In: Proceedings of the 25th International Conference on Intelligent User Interfaces, pp. 308–312. ACM, New York (2020)
24. Hutter, F., Hoos, H., Leyton-Brown, K.: An efficient approach for assessing hyperparameter importance. In: 31st International Conference on Machine Learning, ICML 2014, pp. 1130–1144 (2014)
25. Sheikholeslami, S., Meister, M., Wang, T., Payberah, A. H., Vlassov, V., Dowling, J.: AutoAblation: automated parallel ablation studies for deep learning. In: Proceedings of the 1st Workshop on Machine Learning and Systems, EuroMLSys '21, pp. 55-61. ACM, New York (2021)
26. Bansal, N., Agarwal, C., Nguyen, A.: SAM: the sensitivity of attribution methods to hyperparameters. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 8670–8680. IEEE Computer Society, Los Alamitos (2020)
27. Mishra, S., Dutta, S., Long, J., Magazzeni, D.: A survey on the robustness of feature importance and counterfactual explanations. arXiv preprint arXiv:2111.00358 (2021)
28. Hohman, F., Kahng, M., Pienta, R., Chau, D.H.: Visual Analytics in Deep Learning: An Interrogative Survey for the Next Frontiers. IEEE Trans. Visual Comput. Graphics **25**(8), 2674–2693 (2019)
29. Jaeger, H.: The "echo state" approach to analysing and training recurrent neural networks-with an erratum note. German National Research Center for Information Technology GMD, vol. 148, no. 34, pp. 13, Bonn, Germany (2001)
30. Maass, W., Natschläger, T., Markram, H.: Real-time computing without stable states: a new framework for neural computation based on perturbations. Neural Comput. **14**(11), 2531–2560 (2002)
31. Sánchez-Benítez, A., Gößling, H., Pithan, F., Semmler, T., Jung, T.: The July 2019 european heat wave in a warmer climate: storyline scenarios with a coupled model using spectral nudging. J. Clim. **35**(8), 2373–2390 (2022)
32. Montgomery, D.C.: Design and analysis of experiments. John Wiley & Sons, Location (2017)

Boosting Patient Representation Learning via Graph Contrastive Learning

Zhenhao Zhang[1], Yuxi Liu[2](✉), Jiang Bian[2], Antonio Jimeno Yepes[3], Jun Shen[4], Fuyi Li[5], Guodong Long[6], and Flora D. Salim[7]

[1] College of Life Sciences, Northwest A&F University,
Yangling, Shaanxi 712100, China
zhangzhenhow@nwafu.edu.cn
[2] College of Medicine, University of Florida,
Gainesville, FL 32610, USA
{yuxiliu,bianjiang}@ufl.edu
[3] School of Computing Technologies, RMIT University,
Melbourne, VIC 3001, Australia
antonio.jose.jimeno.yepes@rmit.edu.au
[4] School of Computing and Information Technology, UOW,
Wollongong, NSW 2522, Australia
jshen@uow.edu.au
[5] South Australian immunoGENomics Cancer Institute, Faculty of Health and Medical Sciences, The University of Adelaide,
Adelaide, SA 5005, Australia
Fuyi.li@adelaide.edu.au
[6] Australian AI Institute, FEIT, UTS, Sydney, NSW 2007, Australia
guodong.long@uts.edu.au
[7] School of Computer Science and Engineering, UNSW,
Sydney, NSW 2052, Australia
flora.salim@unsw.edu.au

Abstract. Building deep neural network models for clinical prediction tasks is an increasingly active area of research. While existing approaches show promising performance, the learned patient representations from deep neural networks are often task-specific and not generalizable across multiple clinical prediction tasks. In this paper, we propose a novel neural network architecture leveraging the graph contrastive learning paradigm to learn patient representations that are applicable to a wide range of clinical prediction tasks. In particular, our approach consists of three well-designed modules for learning graph-based patient representations, alongside a pretraining mechanism that exploits self-supervised information in generated patient graphs. These modules collaboratively integrate patient graph structure learning, refinement, and contrastive learning, enhanced by masked graph modeling as a pretraining mechanism to optimize learning outcomes. Empirical results show that the proposed approach outperforms baselines in both self-supervised and supervised learning scenarios, offering robust, effective, and more generalizable patient representations in healthcare applications.

Z. Zhang and Y. Liu—Contributed equally.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
A. Bifet et al. (Eds.): ECML PKDD 2024, LNAI 14949, pp. 335–350, 2024.
https://doi.org/10.1007/978-3-031-70378-2_21

Keywords: patient representation learning · self-supervised learning · graph contrastive learning · graph refinement · patient similarity

1 Introduction

The use of deep learning techniques for analyzing Electronic Health Records (EHRs) has received considerable attention in recent years. An EHR, the digitized version of a patient's medical history, includes clinical data such as patient demographics, vital signs, lab test results, medications, and more. Deep learning often does not make pre-defined assumptions and can discover common characteristics among individual patients in large amounts of EHR data, which can be used to support healthcare providers in a wide range of clinical decision-making tasks, such as diagnosis, assessments of disease severity, and treatment choices for patient disease management. Successful applications have ranged from disease diagnosis and prediction [16,25] to evaluating the risk of decompensation or mortality [13]. These studies often employ Recurrent Neural Network [4] or, more recently, Transformers [19] as the backbone models that can learn valuable patient representations from EHR data – a process, often known as patient representation learning. While these approaches have demonstrated promising performance, the learned patient representations are often task-specific; thus, they have to be retrained for new tasks. Accordingly, a fundamental research question is how to learn effective and robust patient representations that are generalizable to multiple, if not all, medical tasks – aligning with the concept of learning foundation models.

Self-supervised pretraining has emerged as a promising strategy to tackle such a question challenge and learn versatile patient representations. This approach can capture different patterns and features in the input data without relying on human-annotated labels, enabling the learning of generalized and transferable representations applicable to a variety of downstream tasks [6]. In this study, we adopt the graph contrastive learning paradigm based on self-supervised pretraining of graph neural networks. Multiple graph views of the input data are created via data augmentation techniques, and graph representations are then generated using contrastive learning [23,29,30]. Recently, the graph contrastive learning paradigm has gained attention for its effectiveness in representation learning, especially in areas where network graphs are readily available, such as in social recommendation systems [24] and molecular property prediction [26].

However, the application of the graph contrastive learning paradigm in EHR data presents unique challenges. Typical EHRs, characterized by sequential records for each patient (longitudinal), do not naturally conform to a graphical structure. While existing studies have proposed to adapt graph neural networks to EHR data, we argue that these approaches fall short of pretraining on EHR data due to the fact that their proposed graph structures, along with the graph neural networks, are optimized in the context of label-dependent downstream tasks.

In this paper, we introduce a novel neural network architecture that incorporates graph analysis techniques into the graph contrastive learning paradigm.

Patient graph structure learning and refinement are achieved with node-level clustering by assuming homophily. Additionally, we integrate attention mechanisms to enhance the model's focus on only relevant parts of the graph. To generate robust patient representations, we use contrastive learning, aiming to maximize the mutual information between different views of the graph.

The idea of node-level clustering involves grouping the set of nodes into clusters based on similarity, where nodes in the same cluster are likely to be similar, and those in different clusters are dissimilar [10]. The homophily assumption suggests that connected nodes in a graph tend to share similar attributes or labels [21]. We thus utilize the outcomes of clustering as pseudo labels and apply the homophily assumption as a constraint for adjusting the graph structure. Accordingly, in refining the patient graph structure (as illustrated in Fig. 1 and detailed below), edges are added between nodes when they share the same pseudo label and removed if they contradict the homophily assumption (i.e., those with dissimilar pseudo labels).

We enhance the graph view for contrastive learning by introducing a simple yet effective structure augmentation technique. Specifically, we incorporate a random walk strategy into our structure augmentation techniques, which replaces the traditional neighborhood concept in a graph with path-based neighborhoods (i.e., sequences of edges identified within the graph). For contrastive learning, we define the positive and negative samples based on the augmented and main graph views. Positives are derived from an anchor, its counterparts (nodes correspond to the anchor) in different graph views, the neighbors of the anchor, and the node connected to the anchor (having the same pseudo label as the anchor), are treated as positives. Conversely, negatives comprise non-neighbors of the anchor, nodes whose pseudo labels differ from the anchor's, and all the remaining nodes (except for the anchor's counterparts) in different views are treated as negatives. This framework facilitates the formation of positive and negative pairs, which is essential for the contrastive learning process.

We use masked graph modeling as a pretext task to facilitate self-supervised pretraining for graph neural networks, encouraging the model to derive generalized and transferable representations from unannotated graph data. This is achieved by intentionally masking parts of the graph and then challenging the model to predict these masked elements. Specifically, our approach is built upon path-wise masking, which is enabled by the random walk strategy previously mentioned. Unlike the more common edge-wise masking, which typically involves removing, adding, or modifying edges within the input graphs. Path-based masking focuses on sequences of edges connecting adjacent nodes, thus offering a unique approach to altering the graph's structure compared with edge-wise masking. The implications of path-based masking are significant: it forces the model to find more clues over longer sequences of connections, thereby encouraging it to consider broader dependencies within the graph. This requirement not only makes the self-supervised pretraining task more challenging but also imbues the process with deeper learning potential. This approach is compelled to identify more complex patterns and relationships, enhancing its ability to generate robust and comprehensive representations.

The core contributions of this work are as follows:

- We have integrated graph analysis techniques into graph contrastive learning to enhance the learning of patient representations from longitudinal EHRs.
- We proposed a novel neural network architecture, which consists of three well-designed modules for patient representation learning that collaboratively integrate patient graph structure learning, refining, and contrastive learning together to optimize learning outcomes.
- We designed a simple yet effective pretraining mechanism, which consists of masked graph modeling and graph contrastive learning, to achieve the optimized learning outcomes.
- We empirically demonstrated that the proposed approach outperforms baselines in both self-supervised and supervised learning experiments.

2 Related Work

In recent years, various deep learning models have been proposed for clinical risk prediction using EHR data, where representative models include convolutional neural networks [12], recurrent neural networks [13], and attention-based neural networks [14]. In addition to these neural network architectures, graph neural networks (GNNs) have gained popularity due to their ability to handle high-dimensional, graph-structured data. Prominent GNN approaches include graph convolutional network [11], graph attention network [20], and graph convolutional transformer [5], and studies have investigated GNNs on EHR data [15,17,27]. These studies derive graph structures from EHRs and feed them into GNNs to generate patient representations for downstream tasks. Most of them have focused on supervised learning settings for clinical prediction tasks, such as mortality, readmission, and diagnosis prediction.

It is worth noting that a recent study by Cai et al. [1] incorporated hypergraph contrastive learning into EHR data representation learning. The research presented in this study differs from that observed in Cai et al. [1] in the following aspects: (i) their study focused on identifying and evaluating the medical code-code relationship, the patient-patient relationship, and the patient-code relationship. Our research made efforts to the improvement of methodologies for representing EHR data in the form of a graphical structure. (ii) Their network architecture is built upon hypergraphs, which can be treated as predefined, as nodes are connected via hyperedges specified by medical codes. Our proposed network architecture consists of three well-designed modules for graph-based patient representation learning and a pretraining mechanism for exploiting self-supervised information in generated patient graphs. Accordingly, our approach is tailored to the self-supervised learning settings and focuses on achieving optimized self-supervised learning outcomes using non-predefined graph data.

3 Method

3.1 Basic Notations and Problem Definitions

In the EHR dataset, each patient's data is a sequence of time-ordered records. The records of the i-th patient are $X^{(i)} = [x_1^{(i)}, \cdots, x_t^{(i)}, \cdots, x_{T_i}^{(i)}]$, where $x_t^{(i)} = [x_{t,1}^{(i)}, \cdots, x_{t,N_x}^{(i)}]$ is the t-th record, T_i is the total number of records for the i-th patient, and N_x is the number of features of each record. The basic demographic of a patient is $C^{(i)} \in \mathbb{R}^{d_C}$. Given a patient's records and demographics, the patient deterioration prediction task is to predict a binary vector $y \in \{0, 1\}$ that represents the patient's health status; the hospital stay prediction task is to predict a binary vector $y \in \{0, 1\}$ that represents whether the patient's ICU/eICU stay is within 3 and 7 days.

3.2 Architecture Overview

Figure 1 displays an overview of the proposed network architecture.

Patient Graph Structure Learning. Let $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ be a graph with patients as nodes and the similarities between patients as edges, where $\mathcal{V} = \{v_1, v_2, \cdots, v_m\}$ and $\mathcal{E}$ are the node set and edge set and m is the total number of nodes. The objective of patient graph structure learning using EHR data is to learn an adjacency matrix $A \in [0, 1]^{m \times m}$, where $A_{ij} \in [0, 1]$ represents whether there exists an edge between v_i and v_j.

Given the record of patients X, we conduct Gated Recurrent Units over the timestamps and generate an intermediate representation $\bar{X}$ as well as concatenate $\bar{X}$ with C to generate $\hat{X}$ as:

$$\begin{aligned} \bar{X}_1, \bar{X}_2, \cdots, \bar{X}_T = GRU(X_1, X_2, \cdots, X_T), \\ \hat{X} = (\bar{X}_T \oplus C), \end{aligned} \tag{1}$$

where $\hat{X}$ is the new representation generated after concatenation. Subsequently, the similarity matrix $\tilde{A}$ can be calculated using a multi-head attention layer as:

$$\begin{aligned} \tilde{A} &= MultiHeadAtt(\hat{X}) \\ &= [head_1(\hat{X}) \oplus head_2(\hat{X}) \oplus \cdots \oplus head_n(\hat{X})] \cdot W^o, \end{aligned} \tag{2}$$

where $head_n$ is the n-th attention head that calculates the similarities between nodes. In particular, we embed $\hat{X}$ into a lower-dimensional space using linear transformation as:

$$q_n, k_n = W_n^q \cdot x, W_n^k \cdot k. \tag{3}$$

Each $head_n$ has its own projection matrix:

$$head_n(\hat{X}) = SoftMax(\frac{q_n \cdot k_n^\top}{\sqrt{d_k}}), \tag{4}$$

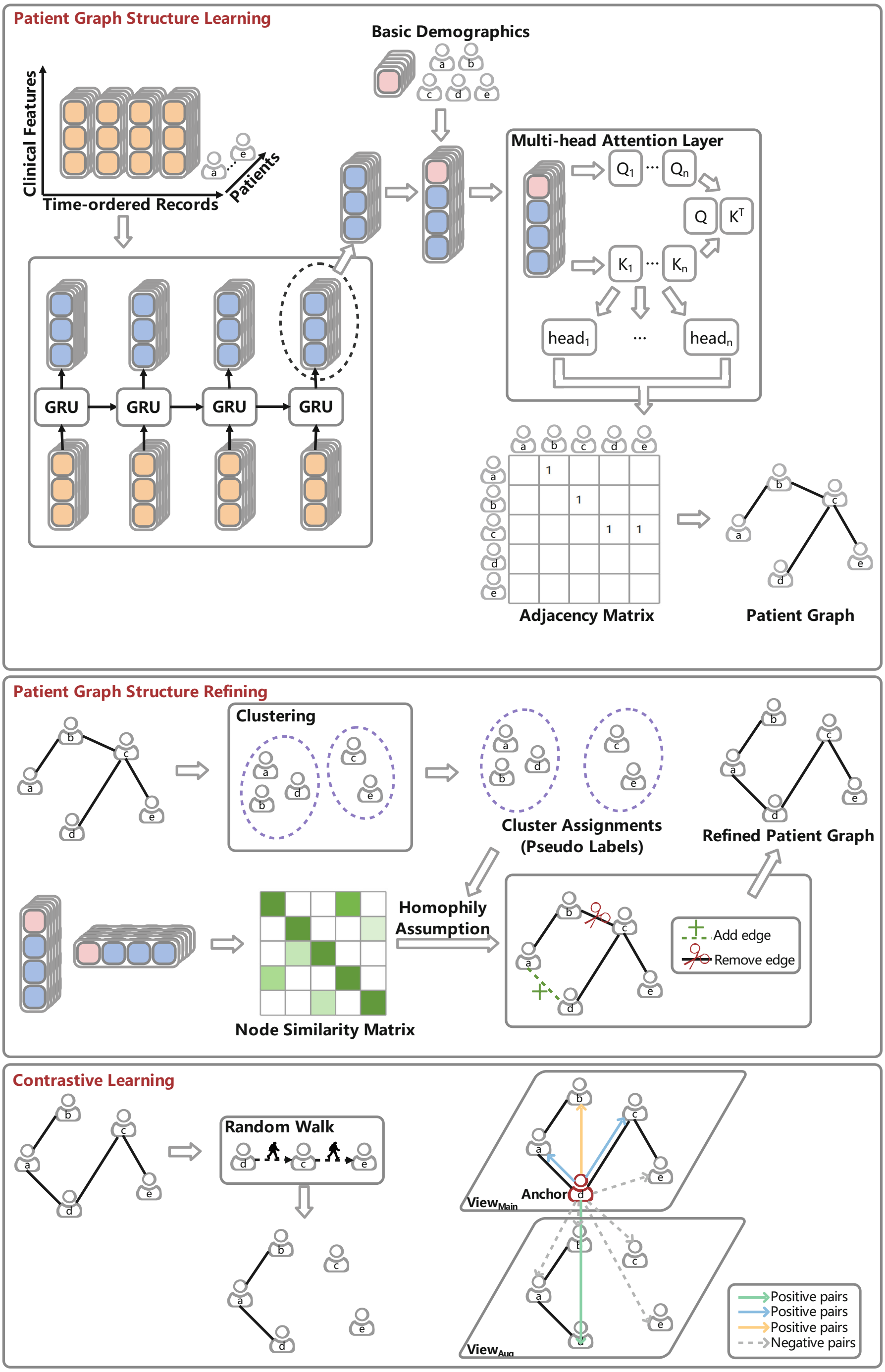


Fig. 1. The proposed network architecture.

where d_k is the dimension of k_n. A note of caution is due here since a learnable threshold ξ is also incorporated into the similarity matrix $\tilde{A}$, where values lower than ξ are filtered out as:

$$A = \begin{cases} 1, & \tilde{A} \geq \xi \\ 0, & \tilde{A} < \xi \end{cases}. \tag{5}$$

Patient Graph Structure Refining. Through the processes above, we have been able to obtain the adjacency matrix A. The objective of patient graph structure refining is to refine A into a well-established $A^* \in [0,1]^{m \times m}$.

Now, we group the set of nodes $\mathcal{V}$ into the number of K clusters. These clusters are separate, and nodes with similar patterns are grouped together. We calculate the similarity between the node embedding $\hat{X}_i$ and the k-th cluster center μ_k by a Student's t-distribution as:

$$q_{ik} = \frac{(1 + ||\hat{X}_i - \mu_k||^2)^{-1}}{\sum_{u=1}^{K}(1 + ||\hat{X}_i - \mu_u||^2)^{-1}}, \tag{6}$$

where q_{ik} is a soft clustering distribution of each node. To obtain the soft clustering distribution of all nodes Q, the k-means clustering is carried out once on the node embedding $\hat{X}$ along with the generation of the initial cluster centers μ. The clustering distribution is optimized in a self-training way [22] as:

$$\mathcal{L}_{KL} = KL(P||Q) = \sum_i \sum_k p_{ik} log \frac{p_{ik}}{q_{ik}}, \tag{7}$$

where $p_{ik} = \frac{q_{ik}^2 / \sum_i q_{ik}}{\sum_u (q_{iu}^2 / \sum_i q_{iu})}$ is the auxiliary target distribution.

Next we treat the clustering results as pseudo labels and adopt the homophily assumption as a constraint. Accordingly, edges between nodes are kept, added, or removed. Edges are added between nodes when they share the same pseudo label, and removed from the existing edge set if against the homophily assumption. Specifically, we measure the pseudo labels using the soft clustering distribution Q as $\tilde{y}_i = \arg\max_k q_{ik}$. We calculate the node similarity between all pairs of nodes using $\hat{X}$ as $Z = \hat{X}_i \cdot \hat{X}_j^\top$, where Z is the node similarity matrix. Accordingly, the edge sets can be refined as:

$$\begin{gathered} \varepsilon_{add}^k = \{(v_i, v_j) | Rank(Z_{ij}) \leq \gamma_{add} \cdot |\varepsilon| \cdot \tfrac{m_k}{m}, (v_i, v_j) \notin \varepsilon, \tilde{y}_i = \tilde{y}_j = k\}, \varepsilon_{add} = \bigcup_k^K \varepsilon_{add}^k, \\ \varepsilon_{del} = \{(v_i, v_j) | Rank(Z_{ij}) \geq (1 - \gamma_{del}) \cdot |\varepsilon|, (v_i, v_j) \in \varepsilon, \tilde{y}_i \neq \tilde{y}_j\}, \end{gathered} \tag{8}$$

where m_k is the number of nodes in the k-th cluster; ε is the existing edge set of the present structure; γ_{add} and γ_{del} are the add and delete ratio; $Rank(Z_{ij})$ is the descending similarity ranking of node pair v_i and v_j; ε_{add} and ε_{del} are the edge sets obtained after refining. The adjacency matrices of ε, ε_{add}, and ε_{del} are denoted by A, $A_{\varepsilon_{add}}$, and $A_{\varepsilon_{del}}$. Accordingly, the adjacency matrix A can be further formalized as: $A^* = A - A_{\varepsilon_{del}} + A_{\varepsilon_{add}}$.

Contrastive Learning. Since the backbone of the graph contrastive learning paradigm is contrastive learning, building multiple augmentation graph views to construct positive and negative sample pairs for contrast is necessary. The existing data augmentation technique on graphs is extensive and focuses particularly on structure augmentation [28]. In response, we establish a simple yet effective structure augmentation technique that uses paths that are sequences of edges found in the graph. Accordingly, the detailed process can be formalized as:

$$\varepsilon_{drop} \sim RandomWalk(\mathcal{V}_{walk}, l_{walk}), \tag{9}$$

where $\mathcal{V}_{walk} \subseteq \mathcal{V}$ is a set of root nodes sampled from a patient graph $\mathcal{G}$ that follows a Bernoulli distribution, i.e., $\mathcal{V}_{walk} \sim Bernoulli(r)$, where $0 < r < 1$ is the sampling ratio, and l_{walk} is the length. Through the processes above, we have been able to obtain the augmentation graph view with the adjacency matrix $A_{aug} = A^{*} C A_{\varepsilon_{drop}}$, where $A_{\varepsilon_{drop}}$ is the adjacency matrix of ε_{drop}. Given A^{*} and A_{aug}, two graph views can be constructed as $View_{Main}$ and $View_{Aug}$. Contrastive learning aims to maximize their mutual information. In particular, the anchor, its counterparts (nodes correspond to the anchor) in $View_{Aug}$, the neighbors of the anchor, and the node in $View_{Main}$ having the same pseudo label as the anchor, are positives. The non-neighbors of the anchor, the nodes with pseudo labels differ from that of the anchor, and the remaining nodes (except for the anchor's counterparts) in $View_{Aug}$ are negatives. These allow the formation of positive and negative pairs for contrastive learning. Subsequently, given $\hat{X}$, GNN-based encoder [11] f_G is utilized to generate node representations for $View_{Main}$ and $View_{Aug}$ as:

$$\begin{aligned} E_{Main} &= f_{\mathcal{G}}(\hat{X}, A^{*}), \\ E_{Aug} &= f_{\mathcal{G}}(\hat{X}, A_{Aug}), \end{aligned} \tag{10}$$

where E_{Main} and $E_{Aug} \in \mathbb{R}^{d_E}$ are node representations for $View_{Main}$ and $View_{Aug}$, respectively. d_E is the dimension. A^{*} and A_{aug} are adjacency matrices. We then employ an feed-forward network (FFN) layer to translate E_{Main} and E_{Aug} into a new latent space as:

$$\begin{aligned} S_{Main} &= FFN(E_{Main}), \\ S_{Aug} &= FFN(E_{Aug}), \end{aligned} \tag{11}$$

where S_{Main} and $S_{Aug} \in \mathbb{R}^{d_S}$ are node representations for $View_{Main}$ and $View_{Aug}$ after projection. d_S is the projection dimension. Last, we select S^{i}_{Main} as the anchor, the contrastive loss between $View_{Main}$ and $View_{Aug}$ as:

$$\begin{aligned} \mathcal{L}_{CL} = -\textstyle\sum_{i=1}^{m} \frac{1}{|\mathcal{N}^{i}_{Main}| + N_{\tilde{y}_i} + 1} \\ log \frac{exp(\varphi(S^{i}_{Main}, S^{i}_{Aug})/\tau) + \sum_{j \in \mathcal{N}^{i}_{Main}} exp(\varphi(S^{i}_{Main}, S^{j}_{Main})/\tau)}{\sum_{j=1}^{m} \mathbb{1}_{[j \neq i]} exp(\varphi(S^{i}_{Main}, S^{j}_{Main})/\tau)}, \\ + \frac{\sum_{j=1, j \notin \mathcal{N}^{j}_{Main}}^{m} \mathbb{1}_{[\tilde{y}_i = \tilde{y}_j]} exp(\varphi(S^{i}_{Main}, S^{j}_{Main}))}{\sum_{j=1}^{m} exp(\varphi(S^{i}_{Main}, S^{j}_{Aug})/\tau)} \end{aligned} \tag{12}$$

where $\mathcal{N}^i_{Main}$ is a set of neighbors of v_i in $View_{Main}$. $|\mathcal{N}^i_{Main}|$ is the number of neighbors of v_i in $View_{Main}$. $N_{\tilde{y}_i}$ is the number of samples with the same pseudo label in each batch. τ is a temperature parameter. $\varphi(\cdot)$ is the inner product.

Self-supervised and Supervised Learning Settings. Through the processes above, we have built the network architecture. Since the proposed network runs as a unit and has multiple learning objectives, we design a hybrid loss that solves the problem of tracking objectives, the combination of $\mathcal{L}_{KL}$ and $\mathcal{L}_{CL}$ as $\mathcal{L}_{Hybrid} = \alpha_1 \cdot \mathcal{L}_{CL} + \alpha_2 \cdot \mathcal{L}_{KL}$, where α_1 and α_2 are two scaling parameters that makes the trade-off between $\mathcal{L}_{CL}$ and $\mathcal{L}_{KL}$. Moreover, the downstream prediction tasks are three binary classification tasks. Accordingly, the cross entropy (CE) is employed as the objective function between the target label y and predicted label $\hat{y}$ as $\mathcal{L}_{CE} = -\frac{1}{m}\sum_{i=1}^{m}(y_i^\top \cdot log(\hat{y_i}) + (1-y_i)^\top \cdot log(1-\hat{y}_i))$, where $\hat{y} = SoftMax(W_y \cdot E_{Main} + b_y)$.

Masked graph modeling is used to mask sequences of edges and reconstruct the masked parts using visible graph structures. It is built upon the encoder-decoder architecture and the use of A_{Aug} as an object. The encoder is f_G, a graph neural-network-based encoder, and E_{Aug} in Eq. (10) is the encoded node representation. The two decoders used for the adjacency matrix and node degree make them as close as possible to the adjacency matrix and node degree in A_{Aug} as:

$$\begin{aligned} \hat{A} = f_{D_{AM}}(E_{Aug}) = Sigmoid(E_{Aug} \cdot E_{Aug}^\top), \\ f_{D_{ND}}(E_{Aug}) = FFN(E_{Aug}), \end{aligned} \tag{13}$$

where $f_{D_{AM}}$ and $f_{D_{ND}}$ are the two decoders used for the adjacency matrix and node degree. We apply cross entropy and mean squared error to $f_{D_{AM}}$ and$f_{D_{ND}}$:

$$\begin{aligned} \mathcal{L}_{D_{AM}} = -\frac{1}{m}\sum_{i=1}^{m}(A_i^* \cdot \hat{A}_i + (1-A_i^*) \cdot log(1-\hat{A}_i)), \\ \mathcal{L}_{D_{ND}} = ||f_{D_{ND}}(E_{Aug}) - deg_{Aug}||_F^2, \\ \mathcal{L}_{MGM} = \beta_1 \cdot \mathcal{L}_{D_{AM}} + \beta_2 \cdot \mathcal{L}_{D_{ND}}, \end{aligned} \tag{14}$$

where deg_{Aug} is the node degree in A_{Aug}. $||\cdot||_F$ is the Frobenius norm. $\mathcal{L}_{MGM}$ is the sum of $\mathcal{L}_{D_{AM}}$ and $\mathcal{L}_{D_{ND}}$, where β_1 and β_2 are two scaling parameters that makes the trade-off between them.

For the self-supervised learning setting, the objective function is $\mathcal{L}_{SSL} = \lambda_1 \cdot \mathcal{L}_{MGM} + (1-\lambda_1) \cdot \mathcal{L}_{Hybrid}$, where λ_1 is a scaling parameter that makes the trade-off between $\mathcal{L}_{MGM}$ and $\mathcal{L}_{Hybrid}$. For the supervised learning setting, the objective function is $\mathcal{L}_{SL} = \lambda_2 \cdot \mathcal{L}_{CE} + (1-\lambda_2) \cdot \mathcal{L}_{Hybrid}$, where λ_2 is a scaling parameter that makes the trade-off between $\mathcal{L}_{CE}$ and $\mathcal{L}_{Hybrid}$.

4 Experiments

4.1 Datasets, Tasks, Evaluation Metrics

All approaches are evaluated on two well-established EHR databases, MIMIC-III and eICU. We follow the settings presented in previous research [7,18] to select available variables for physiologic deterioration and length of stay (LOS) predictions, and their missing values are filled with the empirical mean values [2]. The selected variables are vital signs (up to 17 and 16, respectively) and demographics (age, gender, ethnicity). The prediction window for physiologic deterioration prediction was defined as the first 48 h after admission [7] and for LOS prediction was defined as 3 and 7 days after admission [8]. The AUROC, AUPRC, F1, and Min(Se, P+) are employed to compare the prediction results. In self-supervised learning settings, all approaches (see below) are evaluated on the linear evaluation protocol [3]. Accordingly, logistic regression models were implemented using the patient representation generated from approaches in self-supervised learning settings.

4.2 Comparison Approaches

Under the supervised learning setting, we compare our approach with Transformer [19], GRU-D [2], GCT [5], SimCLR [3], GraphCL [23], GRACE [29], and ConCAD [9]. Under the self-supervised learning setting, we compare our approach with logistic regression (LR), SimCLR, and GRACE. Transformer is an attention-based neural network; GRU-D is a well-known early study often cited in research on EHR data, and its network architecture is built upon Gated Recurrent Unit; GCT pioneered a Graph Convolutional Transformer to learn the graphical structure of EHR data; SimCLR and ConCAD are contrastive learning-based approaches; GraphCL and GRACE are graph contrastive learning based approaches. SimCLR, GraphCL, and GRACE can be implemented in self-supervised learning settings. Note that GraphCL focuses on providing data augmentation techniques on graphs but has difficulty convergent in self-supervised learning settings. A possible explanation of our findings is that the input data needed to be richer for GraphCL. We provide four variants of our approach as follows: $\textbf{Our}_\alpha$: we treat only the anchor and its counterparts in different graph views as positives; $\textbf{Our}_\beta$: we omit the node connected to the anchor, which tests the efficacy of node-level clustering on patient graphs; $\textbf{Our}_\gamma$: we omit the neighbors of the anchor; $\textbf{Our}_\delta$: we use edge-based masking instead of path-based masking. **The source code of our approach, data construction, implementation details, and analysis of hyperparameters are presented in the Github repository**[1].

5 Results and Discussion

As can be seen from the Tables 1 and 2, our approach reported significantly more AUROC, AUPRC, F1, and Min(Se, P+) scores than the other baselines. For instance, the best baseline for ICU deterioration prediction is achieved by GraphCL with an AUROC of 0.7801, an AUPRC of 0.3738, and a Min(Se, P+) of 0.3979. In contrast, our approach reaches an AUROC of 0.7986, an AUPRC of

---

[1] https://github.com/LZlab01/GCL-EHR.

Table 1. Supervised learning results on the MIMIC-III dataset.

ICU Deterioration	AUROC	AUPRC	F1 Score	Min(Se, P+)
Transformer [19]	0.7104(0.0073)	0.2739(0.0118)	0.2858(0.0162)	0.2955(0.0069)
GRU-D [2]	0.7609(0.0229)	0.3319(0.0352)	0.3335(0.0664)	0.3736(0.0308)
GCT [5]	0.7375(0.0266)	0.2603(0.0290)	0.3389(0.0302)	0.3207(0.0378)
SimCLR [3]	0.7638(0.0301)	0.3522(0.0466)	0.3569(0.0386)	0.3871(0.0456)
GraphCL [23]	**0.7801(0.0189)**	**0.3738(0.0337)**	0.3619(0.0278)	**0.3979(0.0239)**
GRACE [29]	0.7129(0.0558)	0.2598(0.0523)	**0.3740(0.0154)**	0.3133(0.0625)
ConCAD [9]	0.7688(0.0252)	0.3499(0.0387)	0.3678(0.0323)	0.3908(0.0450)
Our$_\alpha$	0.7401(0.0383)	0.3074(0.0469)	0.3557(0.0246)	0.3663(0.0454)
Our$_\beta$	0.7604(0.0269)	0.3273(0.0374)	0.3589(0.0297)	0.3813(0.0349)
Our$_\gamma$	0.7747(0.0532)	0.3710(0.0498)	0.3746(0.0185)	0.3921(0.0313)
Our$_\delta$	0.7823(0.0254)	0.3925(0.0316)	0.3705(0.0142)	0.3957(0.0229)
Our	**0.7986(0.0188)**	**0.4014(0.0368)**	**0.3827(0.0189)**	**0.4064(0.0244)**
ICU LOS (3 days)	AUROC	AUPRC	F1 Score	Min(Se, P+)
Transformer [19]	0.6735(0.0032)	0.5202(0.0056)	0.5099(0.0263)	0.5153(0.0066)
GRU-D [2]	0.6903(0.0683)	0.5403(0.0556)	0.5358(0.0571)	0.5289(0.0530)
GCT [5]	0.6841(0.0233)	0.5159(0.0190)	0.5732(0.0259)	0.5197(0.0225)
SimCLR [3]	0.6920(0.0446)	0.5252(0.0326)	0.5897(0.0465)	0.5253(0.0342)
GraphCL [23]	0.6670(0.0665)	0.5142(0.0471)	**0.6058(0.0230)**	0.5027(0.0476)
GRACE [29]	0.6378(0.0636)	0.4841(0.0420)	0.5836(0.0243)	0.4779(0.0462)
ConCAD [9]	**0.6998(0.0407)**	**0.5297(0.0344)**	0.5992(0.0289)	**0.5336(0.0333)**
Our$_\alpha$	0.6553(0.0507)	0.4946(0.0356)	0.5373(0.0337)	0.4782(0.0393)
Our$_\beta$	0.6802(0.0439)	0.5251(0.0335)	0.5794(0.0414)	0.4901(0.0332)
Our$_\gamma$	0.7190(0.0546)	0.5342(0.0464)	0.5855(0.0524)	0.5163(0.0526)
Our$_\delta$	0.7207(0.0455)	0.5393(0.0328)	0.5876(0.0310)	0.5228(0.0346)
Our	**0.7329(0.0331)**	**0.5531(0.0213)**	**0.6094(0.0270)**	**0.5456(0.0352)**
ICU LOS (7 days)	AUROC	AUPRC	F1 Score	Min(Se, P+)
Transformer [19]	0.6988(0.0038)	0.8784(0.0021)	0.6893(0.0558)	0.8430(0.0016)
GRU-D [2]	0.7236(0.0326)	0.8765(0.0126)	0.6931(0.0623)	0.8485(0.0139)
GCT [5]	0.7288(0.0092)	0.8862(0.0037)	0.7837(0.0352)	0.8468(0.0051)
SimCLR [3]	**0.7434(0.0235)**	**0.8922(0.0120)**	0.8018(0.0205)	**0.8570(0.0072)**
GraphCL [23]	0.6870(0.0559)	0.8690(0.0225)	0.8141(0.0279)	0.8372(0.0203)
GRACE [29]	0.6684(0.0656)	0.8635(0.0260)	**0.8143(0.0166)**	0.8296(0.0207)
ConCAD [9]	0.7354(0.0208)	0.8920(0.0072)	0.8047(0.0206)	0.8553(0.0103)
Our$_\alpha$	0.7178(0.0385)	0.8462(0.0191)	0.7997(0.0189)	0.8318(0.0115)
Our$_\beta$	0.7420(0.0522)	0.8619(0.0237)	0.8061(0.0331)	0.8456(0.0194)
Our$_\gamma$	0.7550(0.0557)	0.8731(0.0235)	0.8162(0.0347)	0.8522(0.0169)
Our$_\delta$	0.7574(0.0615)	0.8865(0.0389)	0.8150(0.0586)	0.8473(0.0223)
Our	**0.7626(0.0285)**	**0.8962(0.0198)**	**0.8397(0.0282)**	**0.8652(0.0146)**

Table 2. Supervised learning results on the eICU dataset.

eICU Deterioration	AUROC	AUPRC	F1 Score	Min(Se, P+)
Transformer [19]	0.7315(0.0033)	0.2788(0.0172)	0.3367(0.0159)	0.3074(0.0061)
GRU-D [2]	0.7583(0.0160)	**0.2974(0.0169)**	0.3404(0.0133)	0.3251(0.0190)
GCT [5]	0.7515(0.0103)	0.2718(0.0169)	0.3428(0.0121)	0.3247(0.0222)
SimCLR [3]	**0.7601(0.0084)**	0.2954(0.0146)	0.3581(0.0202)	**0.3268(0.0139)**
GraphCL [23]	0.7581(0.0239)	0.2869(0.0327)	0.3557(0.0298)	0.3204(0.0346)
GRACE [29]	0.7232(0.0851)	0.2704(0.0569)	0.3346(0.0208)	0.3176(0.0750)
ConCAD [9]	0.7592(0.0075)	0.2944(0.0151)	**0.3606(0.0079)**	0.3217(0.0171)
Our$_{\alpha}$	0.7311(0.0389)	0.2606(0.0285)	0.3235(0.0181)	0.2813(0.0324)
Our$_{\beta}$	0.7432(0.0215)	0.2713(0.0166)	0.3378(0.0163)	0.2809(0.0190)
Our$_{\gamma}$	0.7592(0.0156)	0.2839(0.0162)	0.3549(0.0226)	0.2978(0.0168)
Our$_{\delta}$	0.7638(0.0255)	0.2953(0.0210)	0.3533(0.0169)	0.3129(0.0217)
Our	**0.7751(0.0172)**	**0.3006(0.0172)**	**0.3662(0.0175)**	**0.3311(0.0264)**
eICU LOS (3 days)	AUROC	AUPRC	F1 Score	Min(Se, P+)
Transformer [19]	0.8036(0.0013)	0.9265(0.0010)	0.7717(0.0226)	**0.8598(0.0012)**
GRU-D [2]	0.8166(0.0078)	0.9313(0.0034)	0.7808(0.0257)	0.8567(0.0023)
GCT [5]	0.7587(0.0139)	0.9008(0.0082)	0.7394(0.0235)	0.8388(0.0057)
SimCLR [3]	0.8131(0.0046)	0.9282(0.0021)	0.7969(0.0206)	0.8532(0.0026)
GraphCL [23]	0.8085(0.0115)	0.9246(0.0073)	**0.8154(0.0354)**	0.8563(0.0029)
GRACE [29]	0.7813(0.0457)	0.9137(0.0218)	0.8118(0.0254)	0.8456(0.0158)
ConCAD [9]	**0.8195(0.0051)**	**0.9336(0.0023)**	0.7917(0.0175)	0.8594(0.0015)
Our$_{\alpha}$	0.7707(0.0324)	0.9088(0.0128)	0.8247(0.0152)	0.8401(0.0135)
Our$_{\beta}$	0.7861(0.0407)	0.9150(0.0195)	0.8035(0.0235)	0.8473(0.0138)
Our$_{\gamma}$	0.7943(0.0278)	0.9179(0.0151)	0.8241(0.0153)	0.8495(0.0099)
Our$_{\delta}$	0.8022(0.0389)	0.9239(0.0185)	0.8232(0.0150)	0.8368(0.0169)
Our	**0.8295(0.0193)**	**0.9381(0.0168)**	**0.8315(0.0159)**	**0.8616(0.0126)**
eICU LOS (7 days)	AUROC	AUPRC	F1 Score	Min(Se, P+)
Transformer [19]	0.8097(0.0034)	0.9828(0.0004)	0.8018(0.0221)	0.9451(0.0005)
GRU-D [2]	0.8206(0.0128)	**0.9830(0.0016)**	0.8250(0.0457)	0.9503(0.0013)
GCT [5]	0.7943(0.0182)	0.9793(0.0019)	0.8241(0.0321)	0.9469(0.0054)
SimCLR [3]	0.8229(0.0068)	0.9824(0.0008)	0.8553(0.0211)	0.9498(0.0011)
GraphCL [23]	0.8194(0.0065)	0.9819(0.0008)	**0.8689(0.0364)**	0.9502(0.0014)
GRACE [29]	0.8078(0.0340)	0.9801(0.0045)	0.8656(0.0132)	0.9495(0.0032)
ConCAD [9]	**0.8230(0.0045)**	0.9827(0.0006)	0.8479(0.0279)	**0.9516(0.0009)**
Our$_{\alpha}$	0.7779(0.0158)	0.9761(0.0026)	0.8665(0.0241)	0.9455(0.0021)
Our$_{\beta}$	0.7831(0.0224)	0.9772(0.0029)	0.8601(0.0276)	0.9458(0.0027)
Our$_{\gamma}$	0.7909(0.0577)	0.9767(0.0077)	0.8628(0.0253)	0.9446(0.0041)
Our$_{\delta}$	0.8033(0.0332)	0.9805(0.0036)	0.8705(0.0188)	0.9472(0.0043)
Our	**0.8335(0.0230)**	**0.9836(0.0029)**	**0.8901(0.0272)**	**0.9529(0.0026)**

Table 3. Self-supervised learning results on MIMIC-III and eICU datasets.

ICU Deterioration	AUROC	AUPRC	F1 Score	Min(Se, P+)
LR	0.5323(0.0778)	0.1218(0.0324)	0.2178(0.0350)	0.1613(0.0446)
SimCLR [3]	**0.5891(0.0337)**	**0.1513(0.0387)**	**0.2314(0.0230)**	**0.2102(0.0457)**
GRACE [29]	0.5355(0.0629)	0.1273(0.0261)	0.2282(0.0288)	0.1796(0.0323)
Our	**0.6079(0.0741)**	**0.1687(0.0309)**	**0.2495(0.0266)**	**0.2219(0.0360)**
ICU LOS (3 days)	AUROC	AUPRC	F1 Score	Min(Se, P+)
LR	0.5473(0.0361)	0.4115(0.0279)	0.4958(0.0464)	0.4096(0.0253)
SimCLR [3]	**0.5572(0.0474)**	**0.4140(0.0382)**	**0.5163(0.0751)**	**0.4210(0.0364)**
GRACE [29]	0.5528(0.0345)	0.4053(0.0247)	0.5001(0.0299)	0.4038(0.0226)
Our	**0.5763(0.0317)**	**0.4219(0.0256)**	**0.5325(0.0683)**	**0.4392(0.0239)**
ICU LOS (7 days)	AUROC	AUPRC	F1 Score	Min(Se, P+)
LR	0.5516(0.0497)	0.8097(0.0237)	0.7088(0.0395)	0.7860(0.0109)
SimCLR [3]	**0.5718(0.0569)**	**0.8137(0.0293)**	**0.7102(0.0681)**	**0.7997(0.0153)**
GRACE [29]	0.5552(0.0310)	0.8068(0.0150)	0.7146(0.0401)	0.7893(0.0070)
Our	**0.5901(0.0335)**	**0.8227(0.0144)**	**0.7330(0.0527)**	**0.8035(0.0105)**
eICU Deterioration	AUROC	AUPRC	F1 Score	Min(Se, P+)
LR	0.5434(0.0815)	0.1416(0.0265)	0.2715(0.0437)	0.1644(0.0283)
SimCLR [3]	**0.6015(0.0351)**	**0.1647(0.0161)**	**0.2352(0.0589)**	**0.1859(0.0278)**
GRACE [29]	0.5878(0.0677)	0.1662(0.0339)	0.2667(0.0244)	0.1923(0.0424)
Our	**0.6262(0.0325)**	**0.1762(0.0140)**	**0.2822(0.0367)**	**0.2016(0.0258)**
eICU LOS (3 days)	AUROC	AUPRC	F1 Score	Min(Se, P+)
LR	0.5047(0.0909)	0.7523(0.0839)	0.8036(0.0215)	**0.7859(0.0281)**
SimCLR [3]	**0.5894(0.0519)**	**0.8097(0.0265)**	**0.8097(0.0233)**	0.7816(0.0236)
GRACE [29]	0.5501(0.0507)	0.7949(0.0211)	0.7451(0.0383)	0.7757(0.0167)
Our	**0.6138(0.0414)**	**0.8121(0.0235)**	**0.8335(0.0476)**	**0.7924(0.0146)**
eICU LOS (7 days)	AUROC	AUROC	F1 Score	Min(Se, P+)
LR	0.5274(0.0893)	0.9313(0.0285)	0.8195(0.0340)	0.9369(0.0041)
SimCLR [3]	**0.6687(0.0077)**	**0.9515(0.0024)**	**0.8189(0.0208)**	**0.9383(0.0011)**
GRACE [29]	0.6025(0.0320)	0.9446(0.0039)	0.7912(0.0572)	0.9341(0.0020)
Our	**0.6790(0.0568)**	**0.9569(0.0107)**	**0.8237(0.0629)**	**0.9425(0.0023)**

0.4014, and a Min(Se, P+) of 0.4064. In the present report, our approach achieves absolute improvement in AUROC and AUPRC scores. Data from Table 1 can be compared with the data in Table 2, which shows there is a significant difference in performance between the baselines. In particular, it is difficult to argue the best baseline from the data in Table 2. A possible explanation for these results may be that the performance of deep learning models largely depends on the size and quality of input data (e.g., noises). Additionally, the possible interference/impact of hyperparameters on deep learning models cannot be ruled out.

The results obtained from the self-supervised models are set out in Table 3. Our approach reported significantly more AUROC, AUPRC, F1, and Min(Se, P+) scores than the other baselines, but its performance is lower than that of models in supervised learning settings. The reason for this is clear: the performance of LR (used as the basis model) is not very encouraging. Nevertheless, all baselines and our approach in self-supervised learning settings can outperform LR trained with annotated data. Together, these results indicate the effectiveness and superiority of our approach in self-supervised learning settings.

Besides, our approach outperforms all variants (i.e., $\text{Our}_\alpha \sim \text{Our}_\delta$). The results of this ablation experiment indicate the effectiveness and robustness of our proposed modules in model decisions.

6 Conclusions and Future Works

This paper presents a novel neural network architecture that introduces graph analysis techniques into the graph contrastive learning paradigm. The intuition behind our approach is to incorporate graph contrastive learning paradigm in patient representation learning using EHR data. Our approach consists of three well-designed modules for learning graph-based patient representations, alongside a pretraining mechanism that exploits self-supervised information in generated patient graphs. These modules collaboratively integrate patient graph structure learning, refinement, and contrastive learning, enhanced by masked graph modeling as a pretraining mechanism to optimize learning outcomes. Extensive experimental results demonstrate that our approach consistently outperforms existing approaches in both self-supervised and supervised learning scenarios.

References

1. Cai, D., Sun, C., Song, M., Zhang, B., Hong, S., Li, H.: Hypergraph contrastive learning for electronic health records. In: Proceedings of the 2022 SIAM International Conference on Data Mining (SDM), pp. 127–135. SIAM (2022)
2. Che, Z., Purushotham, S., Cho, K., Sontag, D., Liu, Y.: Recurrent neural networks for multivariate time series with missing values. Sci. Rep. **8**(1), 6085 (2018)
3. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: International Conference on Machine Learning, pp. 1597–1607. PMLR (2020)
4. Cho, K., et al.: Learning phrase representations using rnn encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078 (2014)
5. Choi, E., et al.: Learning the graphical structure of electronic health records with graph convolutional transformer. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 606–613 (2020)
6. Ericsson, L., Gouk, H., Loy, C.C., Hospedales, T.M.: Self-supervised representation learning: Introduction, advances, and challenges. IEEE Signal Process. Mag. **39**(3), 42–62 (2022)
7. Harutyunyan, H., Khachatrian, H., Kale, D.C., Ver Steeg, G., Galstyan, A.: Multitask learning and benchmarking with clinical time series data. Sci. Data **6**(1), 96 (2019)

8. Hilton, C.B., et al.: Personalized predictions of patient outcomes during and after hospitalization using artificial intelligence. NPJ Digital Med. **3**(1), 51 (2020)
9. Huang, G., Ma, F.: Concad: contrastive learning-based cross attention for sleep apnea detection. In: Machine Learning and Knowledge Discovery in Databases. Applied Data Science Track: European Conference, ECML PKDD 2021, Bilbao, Spain, September 13–17, 2021, Proceedings, Part V 21, pp. 68–84. Springer (2021)
10. Huang, X., Lai, W.: Clustering graphs for visualization via node similarities. J. Visual Lang. Comput. **17**(3), 225–253 (2006)
11. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016)
12. Liu, Y., Qin, S., Yepes, A.J., Shao, W., Zhang, Z., Salim, F.D.: Integrated convolutional and recurrent neural networks for health risk prediction using patient journey data with many missing values. In: 2022 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), pp. 1658–1663. IEEE (2022)
13. Liu, Y., Qin, S., Zhang, Z., Shao, W.: Compound density networks for risk prediction using electronic health records. In: 2022 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), pp. 1078–1085. IEEE (2022)
14. Liu, Y., Zhang, Z., Yepes, A.J., Salim, F.D.: Modeling long-term dependencies and short-term correlations in patient journey data with temporal attention networks for health prediction. In: Proceedings of the 13th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics, pp. 1–10 (2022)
15. Liu, Z., Li, X., Peng, H., He, L., Philip, S.Y.: Heterogeneous similarity graph neural network on electronic health records. In: 2020 IEEE International Conference on Big Data (Big Data). pp. 1196–1205. IEEE (2020)
16. Luo, J., Ye, M., Xiao, C., Ma, F.: Hitanet: hierarchical time-aware attention networks for risk prediction on electronic health records. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 647–656 (2020)
17. Ochoa, J.G.D., Mustafa, F.E.: Graph neural network modelling as a potentially effective method for predicting and analyzing procedures based on patients' diagnoses. Artif. Intell. Med. **131**, 102359 (2022)
18. Sheikhalishahi, S., Balaraman, V., Osmani, V.: Benchmarking machine learning models on multi-centre eicu critical care dataset. PLoS ONE **15**(7), e0235424 (2020)
19. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)
20. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y.: Graph attention networks. arXiv preprint arXiv:1710.10903 (2017)
21. Wang, T., Jin, D., Wang, R., He, D., Huang, Y.: Powerful graph convolutional networks with adaptive propagation mechanism for homophily and heterophily. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 36, pp. 4210–4218 (2022)
22. Xie, J., Girshick, R., Farhadi, A.: Unsupervised deep embedding for clustering analysis. In: International Conference on Machine Learning, pp. 478–487. PMLR (2016)
23. You, Y., Chen, T., Sui, Y., Chen, T., Wang, Z., Shen, Y.: Graph contrastive learning with augmentations. Adv. Neural. Inf. Process. Syst. **33**, 5812–5823 (2020)
24. Yu, J., Xia, X., Chen, T., Cui, L., Hung, N.Q.V., Yin, H.: Xsimgcl: towards extremely simple graph contrastive learning for recommendation. IEEE Trans. Knowl. Data Eng. (2023)

25. Zhang, Y.: Attain: attention-based time-aware lstm networks for disease progression modeling. In: In Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI-2019), pp. 4369-4375, Macao, China. (2019)
26. Zheng, Z., Tan, Y., Wang, H., Yu, S., Liu, T., Liang, C.: Casangcl: pre-training and fine-tuning model based on cascaded attention network and graph contrastive learning for molecular property prediction. Briefings Bioinform. **24**(1), bbac566 (2023)
27. Zhu, W., Razavian, N.: Variationally regularized graph-based representation learning for electronic health records. In: Proceedings of the Conference on Health, Inference, and Learning, pp. 1–13 (2021)
28. Zhu, Y., Xu, Y., Liu, Q., Wu, S.: An empirical study of graph contrastive learning. arXiv preprint arXiv:2109.01116 (2021)
29. Zhu, Y., Xu, Y., Yu, F., Liu, Q., Wu, S., Wang, L.: Deep graph contrastive representation learning. arXiv preprint arXiv:2006.04131 (2020)
30. Zhu, Y., Xu, Y., Yu, F., Liu, Q., Wu, S., Wang, L.: Graph contrastive learning with adaptive augmentation. In: Proceedings of the Web Conference 2021, pp. 2069–2080 (2021)

Time Series Clustering for Enhanced Dynamic Allocation in A/B Testing

Emmanuelle Claeys[1](✉), Myriam Maumy-Bertrand[2], and Pierre Gançarski[3]

[1] IRIT, University of Toulouse III, 31400 Toulouse, France
emmanuelle.claeys@irit.fr
[2] University of Technology of Troyes, 10430 Rosières-prés-Troyes, France
[3] ICUBE, University of Strasbourg, 67411 Illkirch-Graffenstaden, France

Abstract. An A/B-Test is a method for evaluating online experiments on target items and observing which A/B/C/... variations are better through log reports and statistical analysis of the rewards earned by each variation. Recent advancements in A/B-Tests through reinforcement learning encompass dynamic allocation employing multiarmed bandits (MAB). MABs provides A/B-Tests with fast identification of the best variation (A or B) and helps limit the loss of the test i.e. the cost of exploring low-reward variation. When partial information is available before assigning variations, dynamic allocation is extended to the contextual multiarmed bandit problem (CMAB). Current state-of-the-art approaches for empirically estimating the context-dependent reward function for each variation demonstrate strong performance in limiting test loss and personalized tests. However, few studies have addressed this problem in the context of variable-sized time series. This paper presents a new reinforcement learning methodology to handle A/B-Tests with variable-sized time series as context information. We provide two new methods that obtain a minimization of the cumulative regret with a soft computational cost. This paper also provides numerical results on real A/B-Test datasets, in addition to public data, to demonstrate an improvement over traditional methods.

Keywords: A/B-TEST · Multiarmed bandit · Time series

1 Introduction

In many domains, experimental evaluation is necessary for assessing the relevance of modifications made to an existing entity according to one or more objectives. For instance, an e-marketing team can look for the best modification of a web page design to increase sales [6]. The original variant (A) and its variations (B/C/...) are compared in parallel in a real environment. This leads to the exploration-exploitation dilemma which opposes the cost of learning the best variation (exploration phase) and the benefit obtained by using it in the future (exploitation phase). To tackle this dilemma, novel A/B-TEST-*based*

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
A. Bifet et al. (Eds.): ECML PKDD 2024, LNAI 14949, pp. 351–367, 2024.
https://doi.org/10.1007/978-3-031-70378-2_22

approaches have emerged especially for real-world problems [7,21] that involve sequential decision-making, such as selecting a variation A/B/... The decision-making here consists of assigning the *items* (patients, visitors, recommendations, ...) to the different variations (A/B/...) in order to evaluate the performance of each one (survival rate, average basket, click rate ...). During this *exploration*, it is assumed that the result, called *reward*, of each *assignment*, can be observed after a fixed period, to evaluate the performance of each variation. At the end of this exploration, the *user* can decide which variation should be implemented (i.e., during *production*) based on their performance. In [11], the authors highlighted conducting the A/B Testing in a sequential manner and without a random stopping rule to determine the completion of the experiment using a reinforcement learning policy (also known as a multiarmed bandit, MAB). The MAB is often formulated as the following problem: given a set of bandit "arms" (variation), each associated with a fixed but unknown reward probability distribution [12], an agent selects an arm to play at each iteration (when a visitor comes to the webpage), and receives a random reward variable (click, purchase, ...), sampled according to the selected arm's distribution, independently of the previous actions. More formally, $\mathcal{A}$ is the space of actions (finite), $\mathcal{X}$ a set of rewards, at each iteration $t \in \mathbb{N}^+$, an agent select an arm $a_t \in \mathcal{A}$ and receive a reward $r_t \in \mathcal{X}$ where ξ_t is a noise centered ($\mathbb{E}[\xi] = 0$) and further conditionally sub-Gaussian such as:

$$r_t = \underbrace{f(a_t)}_{\text{reward funct.}} + \underbrace{\xi_t}_{\text{noise}}. \tag{1}$$

Note that f is initially unknown and can be the average (stationary or not) of the chosen arm reward a_t. If the agent chooses, at iteration t, a suboptimal variation, it suffers *simple regret* equal to the difference between the reward from the optimal variation $a_\star$ and the reward from the chosen variation a_t at iteration t. The goal of the agent is to minimize the cumulative sum of regret R_t at $t = T$, i.e. the end of the A/B-Test: $R_T = T\mu_{a^\star} - \sum_{a\in\mathcal{A}} N_a(T) \times \mu_a$ where $a_\star = \text{argmax}_{a\in\mathcal{A}}\, \mu_a$ (and $\mu_{a^\star}$ the average of best arm) and $N_a(T)$ is the number of plays of an arm a at the end T of the A/B-Tests. Thus, an efficient bandit policy must have an average regret less than the average regret of a random policy when $T \to \infty$. A decrease in regret implies that the agent selects arms that maximize gains, leading to an increase in average gain (e.g., average CTR) by the end of the test. Thus, one can study both regret (to minimize) and average gain (to maximize), the choice depending on the ability to observe rewards across all variations. A particularly useful version of MAB is the contextual multiarmed bandit (CMAB) [14], where at each iteration, before choosing an arm, the agent observes a d-dimensional context feature vector $c_t \in \mathbb{R}^d$ sampled from some unknown distribution. In that case the best arm is $a_\star = \text{argmax}_{a\in\mathcal{A}}\langle\theta_a, c_t\rangle$ with $\theta \in \mathbb{R}^d$ as the parameter of an arm. Then a reward becomes: $r_t = f(a_t, c_t) + \xi_t$. The c_t context vector encapsulates essential features of an item, such as age, origin, and gender, revealed before allocation choices. However, conventional CMAB approaches encounter challenges when c_t contains time series data. This inability to use time series to describe an item makes it impossible to employ CMABs in the context of A/B-TESTing related to e-commerce, where time series are commonly used to describe website visitors. However, deploying suboptimal variations into production without knowing a stopping criterion in advance significantly hinders users from conducting A/B tests on their websites. A CMAB that utilizes time series as context would allow both the consideration of the evolving nature of website visitors and the encouragement of users to take risks to test variations. We propose in this paper a new approach by the inclusion of temporal data in c_t. Our methodology performs time series clustering based on the evolving features of visitors who have interacted with the original production page (Version A) in the past before conducting the test. This new approach not only recognizes the significance of time series but also introduces a nuanced preprocessing step, departing from traditional methodologies. By incorporating time series data and employing advanced clustering techniques, our method increases the average gain at the end of A/B-TEST (or decreases the regret). In addition, our innovative approach improves A/B testing practices by facilitating a deeper understanding of consumers behavior. Through the identification of distinct patterns in marketing, our methodology insights into tailoring experiments to specific audience segments. This contributes to a more nuanced interpretation of test. According to the user's needs, we propose two algorithms: DBA-CTREE-UCB and DBA-LINUCB, with significant improvements in terms of regret. Section 2 provides an overview of the state-of-the-art in CMAB methods, both with and without presegmentation. It also introduces the technique employed to address the temporal aspect of CMABs, and Sect. 3 presents our two novel algorithms. Furthermore, Sect. 4 presents the experimental results obtained from various datasets. Finally, Sect. 5 concludes with a discussion.

2 State of the Art

2.1 Classical MAB Problem

The goal of the agent is to use knowledge from past observations to maximize long-term rewards: $a_{t+1} = F_t(a_1, r_1, \ldots, a_t, r_t)$. To achieve this, the agent must determine and select systematically the arm with the highest average reward $a_\star = \text{argmax}_{a \in \mathcal{A}}\, \mu_a$ as soon as possible, thus striking a balance between exploration (testing different arms) and exploitation (selecting the arm with the highest expected reward). Rather than focusing on cumulative rewards, which have no theoretical guarantee, theoretical analyses of bandit models focus on cumulative regret [12]. Observing convergent cumulative regret implies that the agent consistently makes the correct choices systematically after a time $t \in [1; T]$. As such, cumulative regret leads to theoretical bounds that are presented for many bandit algorithms in the literature. An in-depth technical analysis of the classical MAB was given in [13], where policies assuming only one best arm regardless of the item features (c_t) asymptotically reach a regret of $\mathcal{O}(\log T)$. However, to establish a theoretical bound on regret, it is essential to compare the performance against a reference model, often referred to as an oracle, which can learn from all the available data. Among the MABs classically used is UCB algorithm

[12] which, for each arm $a \in \mathcal{A}$, constructs an adaptive upper confidence interval on the mean: $UCB(a,t) = \hat{\mu}_a(t) + \alpha_{\text{ucb}}\sqrt{\frac{2\log(t)}{N_a(t)}}$ with $\alpha_{\text{ucb}} \in \mathbb{R}_+^*$ a positive parameter and $N_a(t)$ is the number of selections of the arm a up to round t. At t, the U.C.B algorithm chooses $\text{argmax}_{a\in\mathcal{A}} UCB(a,t)$. Other methods, such as UCB variants, Thompson Sampling, KL-UCB or EXP3 algorithms [20], can be used as alternatives to UCB. However, these algorithms have a regret bound that is strongly dependent on the joint distribution of the arms. The learning process takes longer when the means of the arms are close. One possible solution to maximize this difference is to leverage the partial information available before selecting an arm, which is defined as the *context*.

2.2 Contextual Multi Armed Bandit

In the CMAB problem, the agent receives partial information such as item features, referred to as context, before making a decision at each iteration. More formally, at each iteration t, the agent observes a particular item described by a d-dimensional feature vector $c_t \in \mathbb{R}^d$. The agent chooses an arm $a_t \in \mathcal{A}$ to apply to this item based on past contexts and rewards observed in previous iterations. Like in classical MAB, the agent cannot observe rewards from arms other than a_t. The context in the CMAB can be defined differently depending on how contexts are revealed and how assumptions are made about the reward function. In our case, each different context $\mathbf{c_t}$ is no longer a vector but a d time series of maximal size m, represented by a matrix $d \times m$. We distinguish the vector representation c_t from its matrix representation $\mathbf{c_t}$ which results in $r_t = f(a_t, \mathbf{c_t}) + \xi_t$ where $\mathbf{c_t} \in \mathbb{R}^{d\times m}$. This problem of context dimension was first studied in The Query-Ad-Algorithm [3]. The Query-Ad-Clustering algorithm achieves a regret of $\mathcal{O}(T^{1-\frac{1}{2+|\mathcal{C}|}+\epsilon})$, where $\mathcal{C}$ is the set of possible contexts and an ϵ positive constant. In the Query-Ad-Clustering algorithm, the reward estimates are accurate as long as the context partitions are similar to each other. However, when the context dimension is large, the regret bound becomes almost linear. This issue is addressed in [16], where the arm rewards are assumed to depend on an unknown subset of the context. It is demonstrated that the regret in this case depends only on the number of relevant groups [8] and requires a learning prestep; however, no details are given on how to achieve this group segmentation. Two approaches are possible in AB testing to define these groups: learning them during preprocessing (data collected before conducting the test) or online with contextual bandit.

Preprocessing Approaches. Preprocessing approaches assume the existence of natural groups, each having a Gaussian reward distribution [16], which can be determined online [15] or before the MAB allocation. The approach involves learning a set of groups $\mathcal{G}$ and a mapping group function association g before the A/B-TEST. Let's $g : \mathbb{R}^d \to \{0;1\}^{|\mathcal{G}|}$ with $|\mathcal{G}| \in \mathbb{N}$ the number of possible groups. When a new visitor described by c_t is submitted to the agent, the function $g(c_t)$ classifies it into one of these groups ($g(c_t) = \widetilde{g}_t$ with $\widetilde{g}_t \in \mathcal{G}$) and then a non-contextual bandit strategy assigns it to an arm according to previous rewards. So partitioning between several groups limits regret when divergence is maximized. In [6] the authors propose the use of the Ctree-Ucb algorithm to construct the association function g using a conditional inference tree [9] (see Alg. 1 in Appendix) before applying several Ucb bandits. Their partitioning involves collecting data from the original variation A and defining groups based on past visitors' features. Each of these groups has a reward distribution that can be modeled by a Gaussian distribution. Group detection here is a type of recursive partitioning method that involves the following basic steps:

- Step 1. Select the feature predictor that best separates different values of the reward distribution with a statistical p-value. The p-value error was corrected with a Bonferroni correction (a method to counter the problem of multiple comparisons).
- Step 2. This variable is split, and the data are divided into two datasets.
- Repeat steps 1 and 2 recursively until no further splits can be made based on predefined p-value rejection (according to the α_{CTREE} risk).
- A tree-like group partitioning model g is produced (see Fig. 1).

To train the inference tree, Ctree-Ucb uses the data present before starting the test (denoted as $\mathcal{L}$, collected on variation A). While the set of data from variation B is not observed, the authors in [6] show that in many cases of A/B-Tests the groups observed on A are also observed in B. During the test, all new items are matched to a predefined group (from the tree). Each group can be supported by a noncontextual bandit. When a new item is presented to the agent, it is automatically assigned to a group based on its features, and an arm is assigned to it through the associated MAB, ensuring a satisfactory response time (millisecond time scale). This allows for a lower cumulative regret than other online learning methods, an interpretability due to the inference tree, and does not slow down the user experience (<200 ms), since the group learning step is performed only once and only non-contextual MABs are used afterwards. The alternative approach to the preprocessing step is learning the context function online, which is described in the following section.

Combinatory Function. Another possible approach is to learn the reward context function without any information before the test. A popular framework for contextual bandits is LinUCB proposed in [14], and variants [5] which assume a linear combination between the context and the d-parameters of each arm. LinUcb estimates the expected reward of each arm a as a linear regression of the context vector c_t, where $\theta_a \in \mathbb{R}^d$ is the regression coefficient of an arm reward function to be learned. We denote by α_{LinUcb} the parameter for the importance of exploration as $\alpha_{\text{LinUcb}} = 1 + \sqrt{(\log 2/\delta_{CI})/2}$ where $1 - \delta_{CI}$ is the confidence interval. We assume that M is an invertible matrix; and denote by $M^{-1} \in \mathbb{R}^{d\times d}$ the updated weight, which can be interpreted as the covariance of the coefficient θ_a. Hence, LinUcb considers the upper confidence bound as $\hat{\theta_a}^T c_t + \alpha_{\text{LinUcb}}\sqrt{c_t^\top M^{-1} c_t}$. The arm a_t selected is the one maximizing the upper

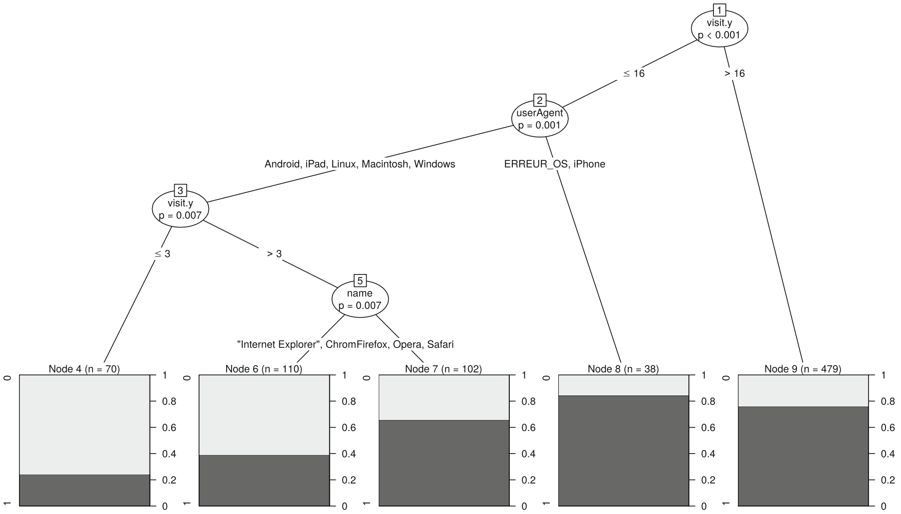


Fig. 1. Example of a conditional inference tree generated from the original page (AB Test dataset [6]) There are 5 groups with different CTR, determined by factors such as the number of past visits on the site (`visit.y`), user agent, and browser name. A separate UCB MAB will be employed for each identified group.

confidence bound: $a_t = \text{argmax}_{a \in \mathcal{A}}(\theta_a^\top c_t + \alpha_{\text{LinUcb}}\sqrt{c_t^\top M^{-1} c_t})$. LinUcb gives a regret in $\mathcal{O}(\sqrt{Td})$. Modified versions of this algorithm, such as SupLinUcb with kernel functions, are studied in [13] where the regret is $\mathcal{O}(\sqrt{Td})$. When d is very large, inverting the matrix M can become computationally expensive and require a significant number of items. To handle nonlinear reward functions, recent works such as kernelized stochastic bandits [24] or deep neural networks [23], have considered the past selected arms and received rewards as training samples. [22] computes a gradient-based upper confidence bound with respect to the trained neural network strategy to select arms. However, when the context is a large vector, the aforementioned approaches need to work on batch style or become computationally expensive, which leads to a significant amount of regret.

2.3 Large-Scale Context

The algorithms detailed in the previous section have strong restrictions on the dimension of the context, where c_t must be a vector of numerical non correlated values. These restrictions limit the use of bandit algorithms when the features to be considered are time series ($\mathbf{c_t} \in \mathbb{R}^{d \times m}$). One solution to this problem is to flatten the time series of each temporal feature. One of the several problems with this approach is that if the size of feature vectors is large, it introduces significant variability in the bandit's performance, leading to a large amount of data to be collected. To address the problem of size context vectors, [4] proposed empirically building a LASSO-type statistical model and integrating it into a bandit problem. However, this approach requires that the features be independent of each other, which is not the case when the features come from the same time series. In [18], the use of LSTM for handling time series data has the drawback of requiring a large number of visitors and frequent model retraining. This results in significant computational costs associated with deep learning methods and slows down the display of the test page, making it impractical for e-commerce applications. Considering temporal features is a difficult task without prior knowledge of the data. Additionally, we found no information in literature about A/B-TESTing contextual bandit algorithms that can adapt to data of variable size. The following section presents our contribution to address this issue.

3 Contribution

3.1 Illustrative Dataset

We detail our approach using the AB tasty dataset 1 (owned by AB TASTY©). AB Tasty Dataset 1 comes from an AB testing platform that compares two versions of the same web page for an e-commerce site. The allocation of A or B to a visitor is randomized with a static allocation. Each visitor is assigned to a single variation until the end of the test (conducted over 15 days). If the visitor has made a purchase from the test page, a reward of 1 is assigned; otherwise, the value is 0. The data contains the history of visitor sessions generated before arriving on the test page: for each visitor, $\mathbf{c_t}$ includes the following information from the visitor's first visit day on the site until the day he/she arrives on the tested page.

- `presence_time_serie`: series of binary values indicating for each day whether the visitor visited the site or not.
- `connexion_time_serie`: series of hours of connections (when the visitor arrived at the site).
- `time_spend_time_serie`: series of visitor's session duration (milliseconds).

There are $n = 5156$ visitors, the shortest session is 2 days, and the longest session is 14 days.

3.2 Two New Algorithms DBA-CTREE-UCB and DBA-LINUCB for Handling Times Series for A/B-TESTS

Currently, there is no existing state-of-the-art method that effectively handles an evolving item context. However, the data presented prior to the beginning of the test could enable the transformation of this context, making it usable by a traditional context bandit method. This preprocessing step has already been employed in CTREE-UCB [6] to create an item segmentation that maximizes the difference between arm means and has demonstrated a significant decrease in cumulative regret. Our contribution proposes a model for replacing a visitor's time series with clusters and demonstrates how this improves dynamic allocation compared to transforming series into averages. We introduce two novel extensions, DBA-Ctree-Ucb and DBA-LinUCB, which complement the search for context-dependent optimal variations with preprocessing utilizing DBA-DTW-kmeans. As we said in the introduction, domains such as e-marketing, and the recognition of evolving customer profiles are crucial for effectively targeting consumers, personalizing offers, and improving the user experience in an A/B test. Time series clustering can handle complex and multidimensional data, taking into account various variables such as purchase history, online interactions, geographical data, etc. By utilizing this method, marketers obtain a holistic view of consumer profiles and identify specific subgroups within the population. The clustering parameters can be adjusted based on the study objectives, such as the desired number of clusters or sensitivity to changes. To include time series clustering in a dynamic allocation algorithm, we propose two algorithms: DBA-Ctree-Ucb and DBA-LinUCB. These algorithms improves the Ctree-Ucb and LinUcb algorithms, respectively, which do not handle temporal aspects. It is important to note that the user is expected to obtain an original variation (such as web page A) and the historical data (such as logs) of the items that have undergone this original variation. These data, collected before their arrival on the test page, form learning clusters required for the DBA-LinUCB and DBA-Ctree-Ucb algorithms. Each temporal series in the known items is replaced as a categorical feature: a cluster. The following section details our choice for time series clustering.

Choice of Time Series Clustering. Since the context of a user is represented by d time series, a solution could be to transform these series by d clusters. In our case, we decided to use a partitioning approach based on the similarity between items. Partitioning methods are computationally efficient and can handle large A/B-Test datasets with ease. To implement such a method, we needed to set a similarity measure. Indeed, in time series clustering methods, if one wants to compare series with irregular sampling or of different sizes, particular attention must be given to the choice of similarity measure. There are many methods for measuring similarity between time series. The most well-known method is Euclidean distance, which involves calculating the sum of the squared distances between the corresponding elements of the considered sequences at each time step. If this distance is commonly admitted, it cannot be used in our case because it requires that the series be the same length. Length resampling would be problematic for a marketing application as it would risk losing the evolutionary aspect of the customer path. However, in our work, we place ourselves in a case where the observation of the context variables can be irregular. For example, one visitor may visit the site more frequently than another. The similarity measure DTW (dynamic time warping) is a metric between two series of different sizes widely recognized as relevant for many application domains. The DTW method matches the elements of the sequences by aligning them in a way that respects the total order of the sequence of values, without crossing the associations (see Alg. 2 in Appendix). DTW can be sensitive to noise, but this is not a problem in our web application where the data is not noisy. A warping path is constructed by computing the minimum cumulative distance between all possible pairs of points in the two-time series. It is usually calculated using the Euclidean distance, but other distance metrics can also be used. Once the warping path is obtained, the DTW distance is computed as the sum of the distances along the path. The objective function optimally solved by DTW corresponds to the minimization of the sum of the costs of the different associations. Figure 8 in Appendix illustrates the concept of the similarity measure DTW.

After defining the similarity metric used in our approach, we propose to use an averaging method based on the barycentre (Dynamic Barycenter Average, see Alg. 3 in Appendix) [17]. Once the series have been aligned using DTW and represented by their average series using DBA, the K-means divide the data into K clusters by similarity, where K is a predefined number (see Alg. 4 in Appendix). It assigns each series to the cluster whose average series is closest to it in terms of DTW similarity. Our number of clusters is based on the Silhouette score: it measures how similar a series is to each other. The Silhouette score, denoted as $S(i)$, is computed using the formula:

$$S(i) = \frac{b(i) - a(i)}{\max\{a(i); b(i)\}}$$

where $a(i)$ is the average DTW distance from the ith series to others in the same cluster, and $b(i)$ is the average distance from the ith series to series in the nearest cluster to which it does not belong. The optimal number of clusters is the one that maximizes the average Silhouette score. A higher Silhouette score indicates better-defined clusters. Our experiments have demonstrated that this is the most reliable indicator for determining the number of clusters (see Experimentation section). By combining these three components, the DBA-DTW-kmeans method identifies clusters of similar profiles, taking into account temporal variations and using a more flexible similarity measure. The Fig. 2 shows an example with centroids obtained from DBA-DTW-Kmeans on the preprocessing step (774 visitors) on AB Tasty Dataset 1. We apply DBA-DTW-kmeans to `presence_time_serie`, `connection_time_serie` and `time_spend_time_serie` that describe visitors to AB Tasty dataset 1.

Dynamic Allocation. Each time series is replaced by the centroid to which it is closest. For DBA-CTREE-UCB the tree model g is learned from $\mathcal{L}$, where each time series has been substituted with the number of clusters to which it belongs. The second step of these two algorithms corresponds to the online A/B-TEST step involving CTREE-UCB and LINUCB. Cluster replacement can then be directly utilized by a contextual bandit model. Consequently, in the DBA-LINUCB/DBA-CTREE-UCB version, the allocation is down according to a LINUCB/CTREE-UCB modelling (see Alg. 5 and Alg. 6 in Appendix). The global schema of our idea is drawn in Fig. 9 in Appendix. The offline part is common to both methods: the past log is used to generate the clusters. An

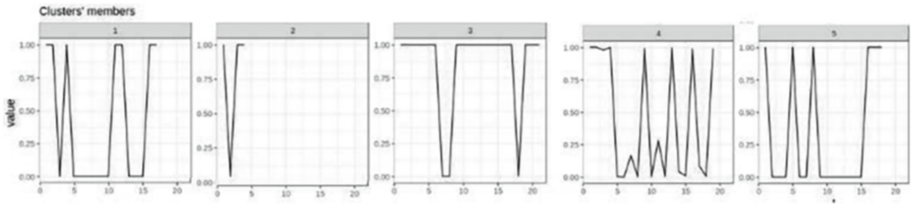


Fig. 2. The series represents 5 centroïds based on the presence/non-presence per day series associated with visitors (from the first day of their visit to the site to the day they appeared on the page being tested). If the visitor does not log on during the day, the series is set to 0.

additional step for DBA-Ctree-Ucb also creates the segmentation tree that will be used in the online part. The online part corresponds to the start of the test. The time series of new items are replaced by the nearest cluster, and the allocation of variation depends on the bandit strategy used (according to the segmentation tree g for DBA-Ctree-Ucb and according to the upper bound of a linear regression for DBA-LinUCB). The DBA-LinUCB algorithm uses a linear approach to estimate the parameters between clusters and potential rewards. This approach assumes that this reward function can be approximated by a linear function. The DBA-Ctree-Ucb divides the cluster sets into distinct groups, enabling the capture of more complex and nonlinear function links between the context and rewards. It evaluates each group and identify optimal variation according to each group by the Ucb strategy. This approach facilitates faster exploration when one or more groups demonstrate sensitivity to the test while avoiding unnecessary slowdown in cases where the test would yield no changes for some items (e.g. Fig. 3).

4 Experiments

Here, we present the results of the DBA-Ctree-Ucb and DBA-LinUCB models on real-world datasets. We compare our algorithms with Ctree-Ucb [6] and LinUcb [1] which require transforming each context time series by taking the average of its values. These experiments observe whether clustering (which is defined by a partial observation of the data) significantly reduces regret. The choice of the number of clusters is first evaluated using the Silhouette index [19]. All experiments are reproducible in our R package from our GitHub repository[1].

AB Tasty Dataset 1. We present the results on the dataset introduced in Sect. 3.1. The available payoffs are those associated with the pages presented in the case of a static allocation, so observing the regret requires a k-nearest neighbors replacement technique: we replaced the missing values by sampling from the visitors at a minimal distance (<10% of the sequence for each series). If no

---

[1] https://github.com/manuclaeys/banditWithR.

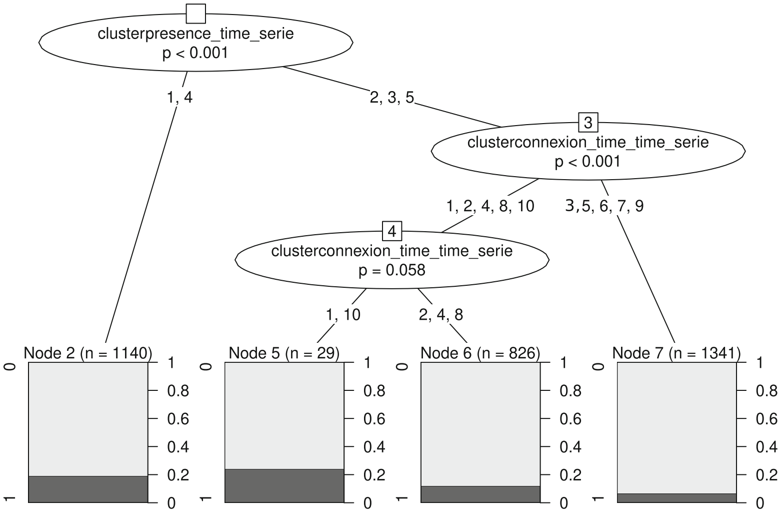


Fig. 3. Preprocessing tree obtained by DBA-Ctree-Ucb on AB TASTY dataset 1. Each leaf represents a click rate. 4 groups are identified from the pre-processing dataset.

similar visitor profile exists, a reward is randomly drawn from the alternative page dataset. The "k-nearest neighbors" method, employed to replace missing data, keeps the correlation between the time series and the corresponding generated rewards. Unlike predefined theoretical replacement models, this method does not rely on such models, which may introduce a strong dependency between missing values and their replacements in small datasets; however, this is not the case in our experiment.

Number of Clusters and Regret. The cumulative regret of DBA-Ctree-Ucb and DBA-LinUCB is influenced by the choice of number of clusters, as depicted in the three graphs of Fig. 4. Our method chooses to parameterize the number of clusters based on the Silhouette index (here 5, 5, 10 for the three series). When employing DBA-Ctree-Ucb, the use of an inference tree to partition visitors into subgroups introduces the possibility of grouping clusters with no significant differences (statistically) if the number of clusters is set too large (see Fig. 3). On the other hand, for DBA-LinUCB, the learning time and regret increase linearly with the number of clusters. Consequently, choosing an inappropriate value will lead to a longer learning process and result in a greater level of regret. The centroids generated by DBA-Ctree-Ucb and DBA-LinUCB must be interpretable from a marketing perspective by the user. Those presented in Fig. 2 represent, for example, visitor patterns such as: the regular visitor (3), those prospecting for a product (1, 4, and 5) with varying intervals between visits, and those who do not return (2). By coupling with other types of series DBA-Ctree-Ucb identifies "high-potential" or "low-potential" visitors. In the generated tree (see Fig. 3), DBA-Ctree-Ucb identifies 4 visitor groups, each with a varying

CTR. New visitors placed in these groups had an independent allocation by Ucb policy, and the user was able to identify at the end of the test which variation was better for each group. It should be noted that generally, groups with very low CTR are usually less sensitive to the test. For example, visitors in group 7 (see Fig. 3) showed no preference between variations. This visitor group was highly represented in the traffic, and its 'isolation' helped accelerate learning for the other groups.

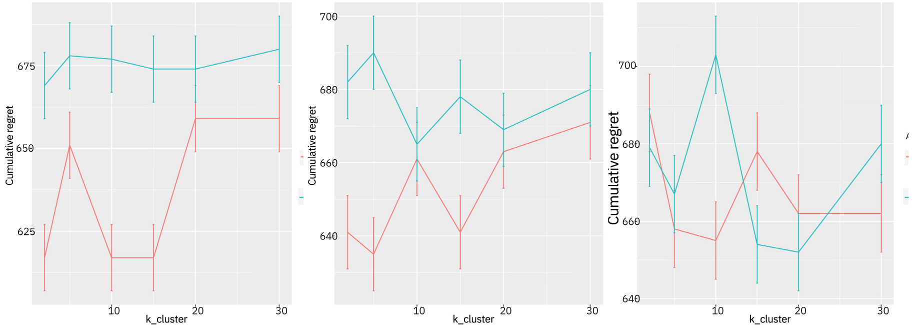


Fig. 4. Cumulative regret obtained by DBA-Ctree-Ucb (red) and DBA-LinUCB (light blue) over number of clusters for AB Tasty dataset 1 on `presence_time` (left), `connexion_time` (middle) and `time_spend` (right). For each graph, the algorithms are trained using only one of the three series as feature context information. In comparison, Silhouette's indexes suggest the following settings (`presence_time` $=5$, `connexion_time` $= 5$, `time_spend` $= 10$). (Color figure online)

The evolution of cumulative regret according to time for all methods is presented in Fig. 5. It is observed that the gap between DBA-Ctree-Ucb and the other methods widens during learning. The cumulative regret (to minimize) according to clusters setting for DBA-Ctree-Ucb and DBA-LinUCB is shown in Fig. 4 and confirms that the silhouette index is a good indicator for choosing the number of clusters before starting the test. The average click-through rates (to maximize) as a function of the number of clusters are also referenced in Table 1 for comparing all methods with different settings (number of clusters, web page used for preprocessing). The lower performance of DBA-LinUCB compared to DBA-Ctree-Ucb can be explained by the fact that if the number of clusters is large, DBA-LinUCB will require more data than DBA-Ctree-Ucb. The differences in terms of regret may seem small, but as the objective of the test is a click to buy, the probability of success is relatively low, regardless of the variation displayed. The average click rate at the end of A/B-Test was 14.14% for DBA-Ctree-Ucb, 12.79% for Ctree-Ucb and <12% for DBA-LinUCB, LinUcb and UNIFORM (see Table 1).

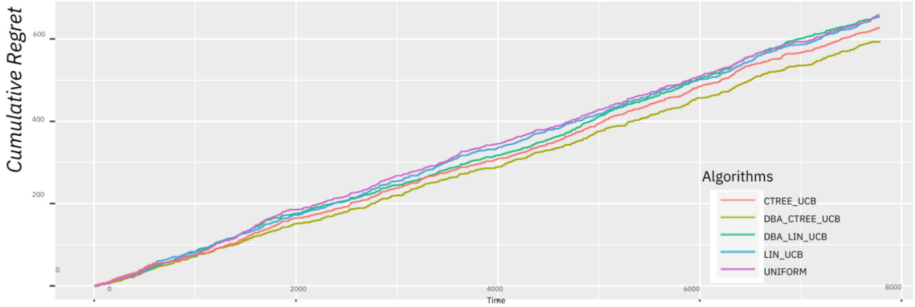


Fig. 5. Cumulative Regret Ucb, LinUcb, Ctree-Ucb, DBA-Ctree-Ucb and DBA-LinUCB on AB TASTY dataset 1

Complementary Experiment: AB Tasty 2. We provide a complementary experiment on another AB Tasty dataset (AB Tasty dataset 2). These data have the same type of time series and come from a video streaming website. The objective of the AB Test is to increase the click-through rate of the tested page. To learn the clusters and groups, we test several configurations. One uses 30% of the visitors (1591), and the other uses all the visitors (5306). The reader can see the results in the Table 1, in particular the interest of the two new approach proposed to increase the average click rate at the end of the experiment. DBA-Ctree-Ucb identified 2 visitor groups (see Fig. 6) where Ctree-Ucb did not identify any groups. The Silouhette index suggested setting 10,5 and 10.

Localization Data for Posture Reconstruction. In this experiment, we assess the performance of DBA-Ctree-Ucb and DBA-LinUCB in a non-e-commerce setting using the 'Localization Data for Posture Reconstruction' dataset [10]. The

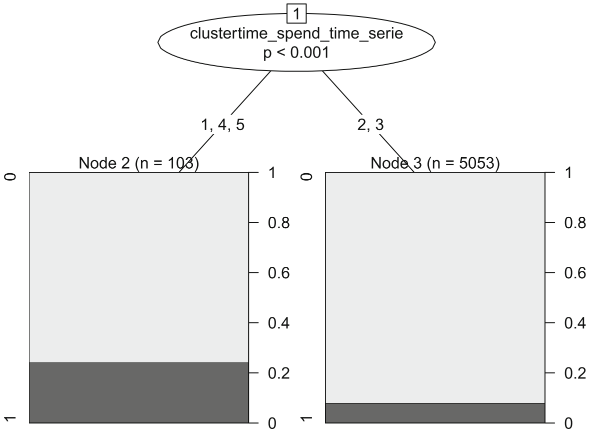


Fig. 6. Preprocessing tree obtained by DBA-Ctree-Ucb on AB TASTY dataset 2. Each leaf represents a click rate. 2 groups are identified from the pre-processing dataset.

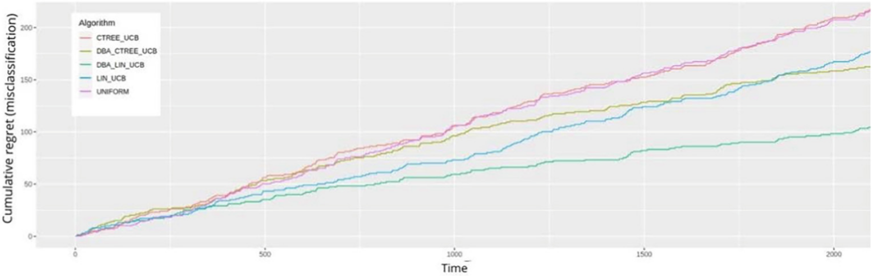


Fig. 7. Cumulative regret Ucb, LinUcb, Ctree-Ucb, DBA-Ctree-Ucb and DBA-LinUCB for Position Dataset

dataset comprises 164860 positional measurements from five patients, each represented by three time series $(V1, V2, V3)$ capturing displacements along the x, y, and z axes. The objective is to differentiate between 'sitting' and 'sitting on the floor' activities based on these measurements. Our A/B test categorizes activities as A ('sitting on the floor') and B ('sitting'), with rewards assigned for successful detection. We frame this as a classification problem, aiming to identify patient activities accurately without imputation, unlike traditional e-commerce datasets. The study involves 1000 items and various bandit configurations. The results are on Table 2. The parameter settings are according to the max Silhouette index: $K_{V1} = 10$, $K_{V2} = 10$, $K_{V3} = 10$. The regret comparison is shown in Fig. 7 pro-

Table 1. Average click rate at the end of the test according to different settings for AB Tasty dataset 1 and 2

	AB Tasty Dataset 1		AB Tasty Dataset 2	
	$\text{Conf}_{30,70}$			
DBA-Ctree-Ucb Nb clusters	$\mathcal{L} = P_1$	$\mathcal{L} = P_2$	$\mathcal{L} = P_1$	$\mathcal{L} = P_2$
3;5;5	**13.80 ± 1.1%**	12.56 ± 1.1%	**9.53 ± 1%**	**9.53 ± 1%**
5;5;10	**14,14 ± 1.2%**	**13.47 ± 1.2%**	**9.53 ± 1%**	**9.53 ± 1%**
10;5;10	**13.67 ± 1.2%**	**12.82 ± 1%**	**9.92 ± 1%**	8.66 ± 1%
DBA-LinUCB Nb cluster				
3;5;5	11,61 ± 1.1%		8.39 ± 1%	
5;5;10	11,89 ± 1.2%		8.51 ± 1%	
10;5;10	11.81 ± 1.2%		9.19%	
Ctree-Ucb	12.79% ± 1%	11.47% ± 1%	9.07%	9.09%
LinUcb	11.56% ± 1%		8.57 ± 1%	
UNIFORM	11.49% ± 1%		8.26% ± 1%	

vides additional evidence to support the effectiveness of the method effectiveness for signal-type series. For this dataset, DBA-LinUCB achieves the best performance. The lower performance of DBA-Ctree-Ucb can be explained by the fact that the sensors are not strongly correlated with each other, and the choice of sensor used for learning strongly influences the algorithm's performance.

Table 2. Average classification rate at the end of the test according to different settings for Localization Dataset

	Localization Dataset		
	Conf$_{30,70}$		
DBA-Ctree-Ucb Nb cluster	$\mathcal{L} = V1$	$\mathcal{L} = V2$	$\mathcal{L} = V3$
5; 5; 10	68.9% ±3%	44.1% ±4%	43,6% ±3%
10; 10; 10	69,8% ±3%	58.6% ±4%	66,8% ±3%
10; 15; 15	68,8% ±4%	57.3% ±4%	35,6% ±4%
DBA-LinUCB Nb cluster			
5; 5; 10	**76% ± 3%**		
10; 10; 10	**81,7% ± 3%**		
10; 15; 15	**74,6% ± 3%**		
LinUcb	66,2% ±2%		
Ctree-Ucb	33.7% ±3%	32.2 ±3%	31.5 ±4%
UNIFORM	32,5% ±1%		

5 Discussion of Results and Conclusion

The choice of cluster number K, based on the Silhouette index, indicates that the algorithm is suitable for parameter selection. We tested 6 other quality indices, none of which yielded such results. One crucial aspect of our method is the reduction in context size, which leads to a significant decrease in computation time. Since the cluster identification component is executed offline, it does not decrease the efficiency of the online A/B-Test test step. The DBA-Ctree-Ucb method enables the grouping of previously learned clusters if the reward distributions (on variation "A") are statistically identical. Therefore, setting the number of clusters too large has minimal impact on the regret, as clusters with identical reward distributions are grouped together. Our experiments also demonstrated that DBA-Ctree-Ucb and DBA-LinUCB facilitate business interpretation of clusters. For instance, separating a "perfect prospect" from a "visitor who arrived by mistake" can be challenging. Their visits are nearly short before they reach the test page. By combining different types of clusters (one based on presence and the other on time spent on the site) and predicting their click probabilities, we can differentiate between these two profiles and determine the most suitable variation. DBA-Ctree-Ucb/DBA-LinUCB appears to be a more advantageous method than Ctree-Ucb/LinUcb in terms of regrets. Clusters allow for the construction of more homogeneous groups in terms of reward distribution rather than relying solely on series averages. Separating groups based on series mean values makes learning highly sensitive to extreme values, which the clustering model avoids. However, DBA-Ctree-Ucb requires a correlation between the earnings of different variations: the earnings distribution of a group, whether on variation A or B, follows the same distribution (with the same variance), but the means may differ. In a further work we will show how this pre-processing step has helped the user to create more personalized variation B by generative DNN.

References

1. Agarwal, A., Dudík, M., Kale, S., Langford, J., Schapire, R.E.: Contextual bandit learning with predictable rewards. ArXiv e-prints, February 2012
2. Auer, P., Cesa-Bianchi, N., Fischer, P.: Finite-time analysis of the multiarmed bandit problem. Mach. Learn. **47**(2), 235–256 (2002)
3. Avadhanula, V., Colini-Baldeschi, R., Leonardi, S., Sankararaman, K.A., Schrijvers, O.: Stochastic bandits for multi-platform budget optimization in online advertising. CoRR (2021)
4. Bastani, H., Bayati, M.: Online decision-making with high-dimensional covariates. SSRN Electron. J. (2015)
5. Bietti, A., Agarwal, A., Langford, J.: A contextual bandit bake-off. JLMR (2021)
6. Claeys, E., Gancarski, P., Maumy-Bertrand, M., Wassner, H.: Dynamic allocation optimization in A/B-tests using classification-based preprocessing. IEEE TKDE **35**(1), 335–349 (2021)
7. Fabijan, A., Dmitriev, P., Arai, B., Drake, A., Kohlmeier, S., Kwong, A.: A/B integrations: 7 lessons learned from enabling A/B testing (2023)
8. Gentile, C., Li, S., Kar, P., Karatzoglou, A., Zappella, G., Etrue, E.: On context-dependent clustering of bandits. In: Proceedings of the 34th International Conference on Machine Learning (2017)
9. Hothorn, T., Hornik, K., Zeileis, A.: Unbiased recursive partitioning: a conditional inference framework. J. Comput. Graph. Stat. **15**, 651–674 (2006)
10. Kaluza, B., Mirchevska, V., Dovgan, E., Lustrek, M., Gams, M.: UCI machine learning repository, an agent-based approach to care in independent living (2010)
11. Kaufmann, E., Cappé, O., Garivier, A.: On the complexity of A/B testing. ArXiv e-prints, May 2014
12. Lai, T., Robbins, H.: Asymptotically efficient adaptive allocation rules. Adv. Appl. Math. **6**(1), 4–22 (1985)
13. Lattimore, T., Szepesvári, C.: Bandit algorithms (2020)
14. Li, L., Chu, W., Langford, J., Schapire, R.E.: A contextual-bandit approach to personalized news article recommendation, pp. 661–670 (2010)
15. Mahadik, K., Wu, Q., Li, S., Sabne, A.: Fast distributed bandits for online recommendation systems. In: Proceedings of the 34th ACM International Conference on Supercomputing (2020)
16. Maillard, O.A., Mannor, S.: Latent Bandits, January 2014. Extended version of the paper accepted to ICML 2014
17. Petitjean, F., Ketterlin, A., Gancarski, P.: A global averaging method for DTW, with applications to clustering. Pattern Recogn. **44**, 678–693 (2011)

18. Rotman, M., Wolf, L.: Energy regularized RNNS for solving non-stationary bandit problems (2023)
19. Rousseeuw, P.J.: Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. JCAM **20**, 53–65 (1987)
20. Thompson, W.R.: On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. Biometrika **25**, 285–294 (1933)
21. Xu, D., Yang, B.: On the advances and challenges of adaptive online testing. In: WSDM 2022 (2022)
22. Zhang, Z., Yang, J., Ji, X., Du, S.S.: Variance-aware confidence set: variance-dependent bound for linear bandits and horizon-free bound for linear mixture MDP. CoRR (2021)
23. Zhou, D., Li, L., Gu, Q.: Neural contextual bandits with upper confidence bound-based exploration. CoRR (2019)
24. Zhou, X., Ji, B.: On kernelized multi-armed bandits with constraints. In: Advances in Neural Information Processing Systems. Curran Associates, Inc. (2022)

Reinforcement Learning Meets Microeconomics: Learning to Designate Price-Dependent Supply and Demand for Automated Trading

Łukasz Lepak[1,2](✉) and Paweł Wawrzyński[2]

[1] Warsaw University of Technology, Pl. Politechniki 1, 00-661 Warsaw, Poland
`lukasz.lepak.dokt@pw.edu.pl`, `lukasz.lepak@ideas-ncbr.pl`
[2] IDEAS NCBR, Chmielna 69, 00-801 Warsaw, Poland
`pawel.wawrzynski@ideas-ncbr.pl`

Abstract. The ongoing energy transition towards renewable sources increases the importance of energy exchanges and creates demand for automated trading tools on these exchanges. Day-ahead exchanges play a prominent role in this area. Participants in these exchanges place buy/sell bids collections before each trading day. However, machine learning-based approaches to automated trading are based on placing a single bid for each time instant. The bid is either executed or not, depending on the relation between the market price and the bid price. This is contrary to economic rationality, which usually requires buying more when the market price is lower and selling more when it is higher. Single bids do not allow the expression of such preferences. In this paper, we fill this gap and design a policy that translates the information available to the trading agent into price-dependent supply and demand curves. Also, we demonstrate how to train this policy with reinforcement learning and real-life data. Our proposed method is now being deployed in a real system for energy storage management. Here, we demonstrate how it performs in four data-driven simulations. The proposed method outperforms alternatives in all cases.

Keywords: Automated trading · Reinforcement learning · Energy market

1 Introduction

In 2023, wind and solar energy represented 14.26% of global electricity generation, after these shares doubled in 5 preceding years [42]. The power of wind and sunlight reaching the Earth's surface is, to some extent, random. Therefore, while the rise of renewable energy sources presents the prospect of cheap and clean energy, it also exacerbates the problem of balancing power supply and demand.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-70378-2_23.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
A. Bifet et al. (Eds.): ECML PKDD 2024, LNAI 14949, pp. 368–384, 2024.
https://doi.org/10.1007/978-3-031-70378-2_23

In many countries, the main institution that balances volatile electricity supply and demand is a day-ahead energy market [13,14,27,30]. Every day, agents participating in this market place their buy and sell bids separately for every hour between 0 am and 11 pm the next day. Market clearing prices are then designated for each of these hours, and the bids are consequently executed or not, depending on the proposed prices.

Here, we consider an agent that (i) consumes electricity, (ii) produces electricity, and (iii) has electricity storage. What is of main interest here is a strategy for automated trading on a day-ahead energy market on behalf of this agent.

Reinforcement learning (RL) [32] is a natural tool to optimize a policy of sequential decision-making in dynamical, stochastic systems that elude modeling. RL has been applied to optimize strategies of on-line energy trading within local energy markets [4,15,20,21,24,28], real-time bidding for internet ads [6], stock market trading [10,18,38,41], power grid control [1,14,25], trading on the day-ahead energy market [8,9].

In existing studies on RL for automated trading, an action either selects a bid from a predefined set or directly defines parameters (type, price, and quantity) of a single bid or a pair (sell and buy) of bids.

The fact that for each bidding, the agent is able to submit only one or two bids is a serious limitation. Most electronic markets allow their participants to define many bids for each time interval. By submitting a collection of bids, the participant can define how much of the commodity he wishes to sell and/or buy, depending on the market price. The actual trading agents usually take advantage of this possibility since buying more when the price is low and selling more when the price is high usually results from economic rationality.

In this paper, we design a strategy that translates the information available to the trading agent into parameters of the supply and demand curves. These parameters are then translated into a collection of bids. The number of bids within the collection is variable. This strategy enables the trading agent to behave rationally in an economic sense, which is not possible when the strategy only produces single bids. We have designed our strategy with the day-ahead electricity market. However, it can also be applied to other electronic markets.

In this paper, we demonstrate the performance of our proposed automated trading strategy in several real data-based scenarios of the day-ahead electricity market trading. The strategy is currently being deployed in a real system for energy storage management.

The paper contributes as follows:

- We design a parametric automated trading strategy suitable for electronic markets with significant lags between bidding and its corresponding transaction. This strategy produces supply and demand curves by means of bid collections of variable sizes, thereby enabling the trading agent to behave rationally.
- We formalize a framework in which on-line RL can be applied to optimize a policy on the basis of recorded observations of the external environment without data on earlier decision-making.

- We apply reinforcement learning to optimize the above strategy and select the best algorithm for this purpose. The resulting strategy is fitted to the data and ready to use in real life.

2 Related Work

Automated Trading on the Electricity Market. Research on automated trading on the electricity market covers various approaches. Some works introduce theoretical frameworks of bidding strategies [5,17,36]. Many authors propose various forms of parametric bidding strategies. These strategies are optimized with methods like linear programming [3], genetic and evolutionary algorithms [2,37] or stochastic optimization [13,19]. However, as a more complex bidding strategy is expected and a more complex transformation of observations into bids is required, these techniques become less effective.

With the advent of electricity prosumers, energy microgrids, energy cooperatives, and flexible price-driven energy consumption, there is an increasing need for automated decision-making and control in various activities undertaken by the energy market participants. Strategies for these agents can be optimized with reinforcement learning. Various applications of RL in power systems are reviewed in [14,26,39]. The authors of [23] analyze bidding on a DA energy market as a zero-sum stochastic game played by energy producers willing to exercise their market power and keep their generators productive. RL is used there to optimize their bidding strategy. In [35], bidding on a DA energy market from the point of view of a flexible buyer (who charges a fleet of electric vehicles) is analyzed. His strategy is optimized with RL. A number of papers is devoted to peer-to-peer trading with electricity on a local, event-driven energy market, with RL applied to optimize the behavior of such peers [4,7,8,15,28]. RL and neural price predictions are used in [20] to optimize the scheduling of home appliances of private users. The authors assume that the electricity prices are changing and are known one hour ahead. The work [4] analyzes a similar setting in which the users also trade energy with each other. This setting is used in [28] to optimize the user strategies with multi-agent RL. The authors of [21] optimize peer-to-peer energy microgrid operations with multi-agent reinforcement learning, with their method generating higher net profits than simple fixed price biddings. Q-Learning and SARSA algorithms are used in [24] to create simple bidding strategies and test them on German real-life data.

The authors of [9] consider simultaneous trading on a DA and hour-ahead energy markets by an energy storage operator as a Markov Decision Process (MDP). The authors use RL to optimize a strategy of bidding on a DA energy market by a battery energy storage system. They use RL to optimize a strategy of bidding on a DA energy market by a battery energy storage system (BESS). However, the authors address the dynamics of that process only to a limited extent. Consecutive days are separate episodes, so the between-day dynamics of the market are not accounted for. Discrete actions define the parameters of the bids. They are not based on external observations such as weather forecasts.

Also, only a single bid can be placed each hour. In the current paper, we address all of these limitations, which leads to significantly better performance of our proposed strategy and allows it to be deployed in real-life scenarios.

Automated Stock Market Trading. In this area, the trading agent observes a set of time series of prices of different assets. The agent makes on-line decisions on buying these assets at the current prices in anticipation of their price increase or selling them in anticipation of their price decrease. Because the problem is formalized as an MDP, it is addressed with RL [10,40].

Additional related works are discussed in Appendix A of the supplementary material.

3 Problem Definition

In this paper, we consider automated trading on the commodity markets with lags between biddings and their corresponding transactions. We specifically focus on the day-ahead energy market, understanding that other commodity markets could be approached alike, with some minor variations.

3.1 Day-Ahead Electricity Market

A trading agent is an entity such as a small- or medium-sized consumer of electricity e.g., a group of households connected together to the power network. We assume that it may consume electricity randomly, produce electricity with weather-dependent sources such as solar panels and windmills, and store energy in batteries.

The trading agent participates in the day-ahead energy market. Every day before 10.30 am[1] the agent submits bids for 24 separate biddings: for hours 0 am, 1 am, ..., 11 pm of the following day. Each bid is defined by the hour, type (sell/buy), price (per 1 kWh), and quantity (in kWhs). Any number of bids for each hour is acceptable. Right after the biddings close at 10.30 am, market prices are designated for each hour. The buy bids with prices higher than or equal to the market price will be executed at the market price. Likewise, the sell bids with prices lower than or equal to the market price will be executed at the market price. On the next day, at each hour, the agent consumes, produces, and transmits the energy to/from the power network according to its bids being executed. The net energy is transmitted to or released from the energy storage. When the agent tries to get energy from empty storage or put the energy into full storage, it actually exchanges it with the market and pays a special fine for that.

The problem is to designate the bids on behalf of the trading agent to maximize the profit gained (or minimize the cost incurred) from participation in the market.

---

[1] We take details from the specific DA market considered in the experimental study.

3.2 Reinforcement Learning to Bid

We adopt the general framework of reinforcement learning [32]. The objective is to optimize a policy that translates relevant available information into bids. The said information defines the state of the environment. It is relevant for future market prices, e.g., weather forecasts or the day of the week. Also, it is relevant to the current situation of the trading agent and its potential to produce and consume energy, e.g., battery charge and, again, weather forecasts.

Every day, the trading agent is receives a reward equal to the financial net result of its bids (and fines). The goal is to optimize the policy to yield the largest possible sums of future discounted rewards in each environmental state the trading agent encounters.

4 Method

4.1 Analysis

Within traditional microeconomics, we analyze the relation between the amount of goods the agent sells or buys and the unit price of these goods. If the agent is only able to express its offered supply and demand in a pair of bids, the agent either sells/buys its defined quantity or not, depending on whether the market price is lower/higher than its defined threshold. The supply/demand curves that visualize these relations can be seen in the top part of Fig. 1. To the best of our knowledge, placing a single bid, or a sell-and-buy pair of bids, at a time has only been considered in the literature of automated trading.

However, it is folklore of microeconomics [16] that a rational economic agent is most often willing to sell a higher quantity of commodity when its market price is higher. Also, the economic agent most likely is willing to buy a higher quantity of commodity when its market price is lower. For our considered trading agent, both the above cases create a lucrative opportunity to sell high and buy cheap. These typical preferences are depicted in the middle part of Fig. 1, in the form of increasing supply curve and decreasing demand curve. How can the trading agent express such preferences with bids?

4.2 Price-Dependent Supply and Demand in Bids

Let us consider, for a given hour h, a collection of sell bids

$$\langle \text{sell}, h, p_s^{h,i}, q_0 \rangle, \quad i = 1, \ldots, n_s^h, \quad p_s^{h,i} \le p_s^{h,i+1}, \tag{1}$$

where $q_0 > 0$ is a certain constant quantity, n_s^h is the number of bids, and $p_s^{h,i}$ are unit prices. Let p_m^h be a market price, and integer j be such that

$$p_s^{h,j} \le p_m^h < p_s^{h,j+1}. \tag{2}$$

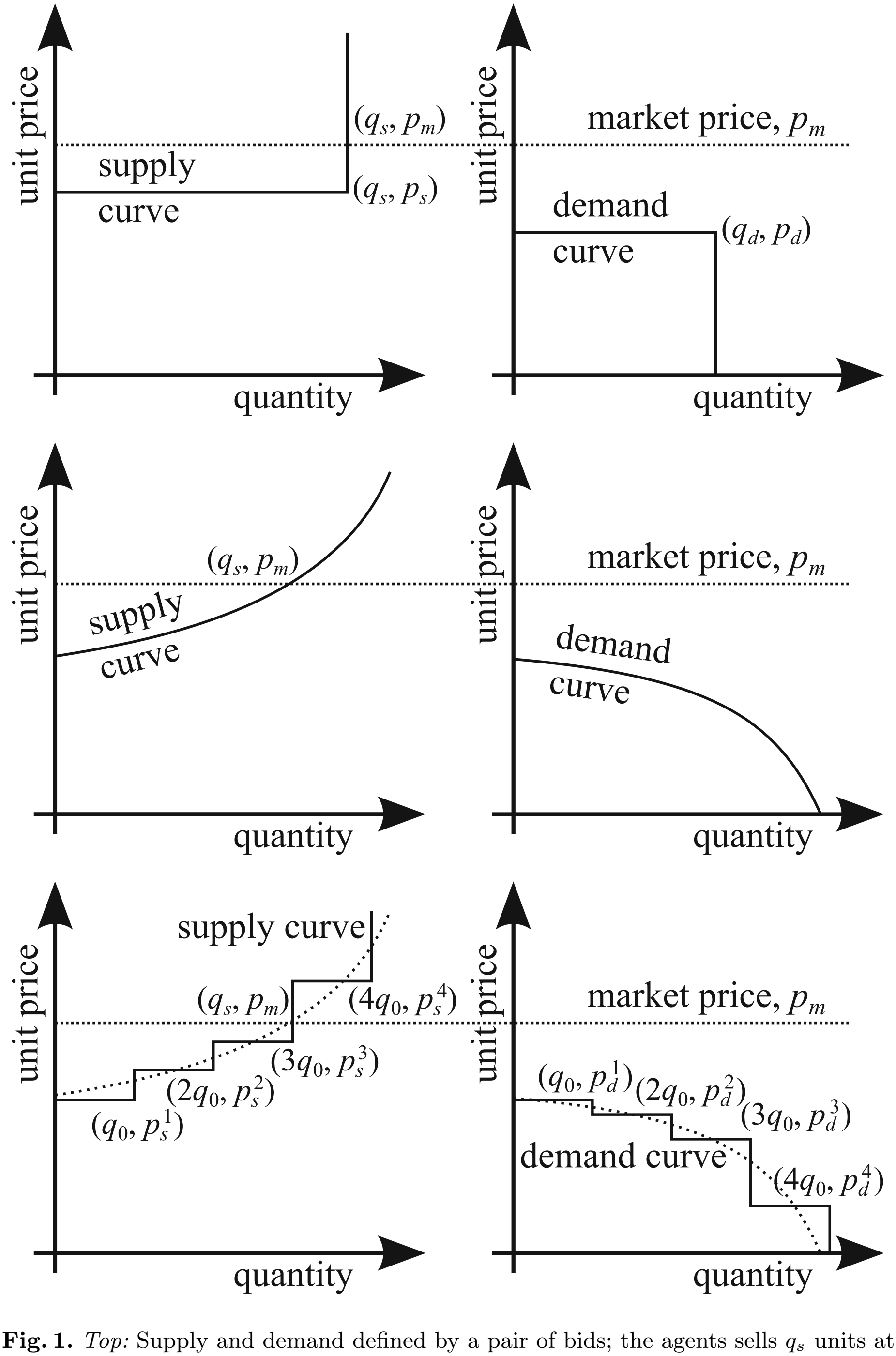


Fig. 1. *Top:* Supply and demand defined by a pair of bids; the agents sells q_s units at the unit price of p_m. *Middle:* Nondecreasing supply and nonincreasing demand. *Bottom:* Nondecreasing supply and nonincreasing demand as defined by a collection of bids.

Then, only the first j bids are executed and the bidding agent sells a quantity of jq_0 at the market price p_m^h. The above collection of bids (1) can thus be represented as a nondecreasing supply curve, similar to that depicted on the left-bottom part of Fig. 1.

Any nondecreasing function can be approximated by a piecewise constant step function. Consequently, any reasonable preferences of selling can be approximately represented by the collection of bids (1). Moreover, for technical reasons, in most electronic markets, quantities can only be defined in bids as integer numbers (or as integer multiples of the minimum tradable quantity). Consequently, any supply curves feasible in the electronic market is a piecewise constant step function, and it can be represented in the form (1).

The above reasoning can be repeated, with similar conclusions, for demand. It can effectively be represented as a collection of bids in the form

$$\langle \text{buy}, h, p_d^{h,i}, q_0 \rangle, \quad i = 1, \ldots, n_d^h, \quad p_d^{h,i} \geq p_d^{h,i+1}, \tag{3}$$

where n_d^h is the number of bids, and $p_d^{h,i}$ are unit prices.

4.3 Parametric Representation of a Collection of Bids

In order to apply reinforcement learning to learn to designate collections of bids in the form (1) and (3), we need a way to translate vectors of predefined dimension into bid collections of variable size. We design this translation as follows. Let the action space be 100-dimensional, $a \in [-1, 1]^{100}$. Coordinates of a single action define all bids for the whole day. The collection of sell bids for the hour $h = 0, \ldots, 23$ is given by (1) with

$$n_s^h = \lfloor c_q \exp(c_e a_h)/q_0 + 1/2 \rfloor \tag{4}$$

$$p_s^{h,i} = c_p^h \exp(a_{h+24}) \Big(1 + \exp(a_{96}) \left(-(2a_{98} + 4)^{-1} + (i/n_s^h)^{2a_{98}+3} \right) \Big) \tag{5}$$

The collection of buy bids for the hour $h = 0, \ldots, 23$ is given by (3) with

$$n_d^h = \lfloor c_q \exp(c_e a_{h+48})/q_0 + 1/2 \rfloor \tag{6}$$

$$p_d^{h,i} = c_p^h \exp(a_{h+72}) \Big(1 + \exp(a_{97}) \left((2a_{99} + 4)^{-1} - (i/n_d^h)^{2a_{99}+3} \right) \Big) \tag{7}$$

where a_k denotes k-th coordinate of the action a, and

- a_h/a_{h+48} defines the width of the supply/demand curve, i.e., the number of sell/buy bids for the hour h,
- a_{h+24}/a_{h+72} defines the average height at which the supply/demand curve is located,
- $a_{h+24} + a_{96}/a_{h+72} + a_{97}$ defines vertical span of the supply/demand curve,
- a_{98}/a_{99} defines convexity/concavity of the supply/demand curve,
- c_q—quantity scaling factor (we assume its value equal to the maximum hourly production of the installed sources),

- c_p^h—price scaling factor (we assume its value equal to the median price for hour h over the last 28 days),
- c_e—quantity exponent scaling factor (we assume $c_e = 3$).

The resulting supply and demand curves are depicted in Fig. 2. Note that the above symbols, except q_0, c_q, c_e, depend on t, but we skip this dependence in the notation.

The supply and demand curves above are designed symmetrically. Thus, let us only analyze $p_s^{h,i}$ (5). The term

$$-(2a_{98}+4)^{-1} + (i/n_d^h)^{2a_{98}+3} \tag{8}$$

makes the supply curve an increasing power function with the exponent $2a_{98}+3$ controlling the convexity/concavity of the curve; for $a_{98} \in [-1, 1]$ the exponent is in the $[1, 5]$ interval. The component $-(2a_{98}+4)^{-1}$ makes the average of (8) over $i \in [0, n_d^h]$ equal to zero. The term $\exp(a_{96})$ controls a vertical span of the supply curve. The values of a_{96} and a_{98} do not impact the average height at which the supply curve is located, which is designated only by the term $c_p^h \exp(a_{h+24})$.

The widths and vertical locations of the curves are specified separately for different hours by their corresponding action coordinates. However, the vertical span of these curves and their convexity/concavity are specified for all hours by the same action coordinates $a_{96} \ldots a_{99}$. This parameter sharing is intended to maintain a low enough dimensionality of the action space.

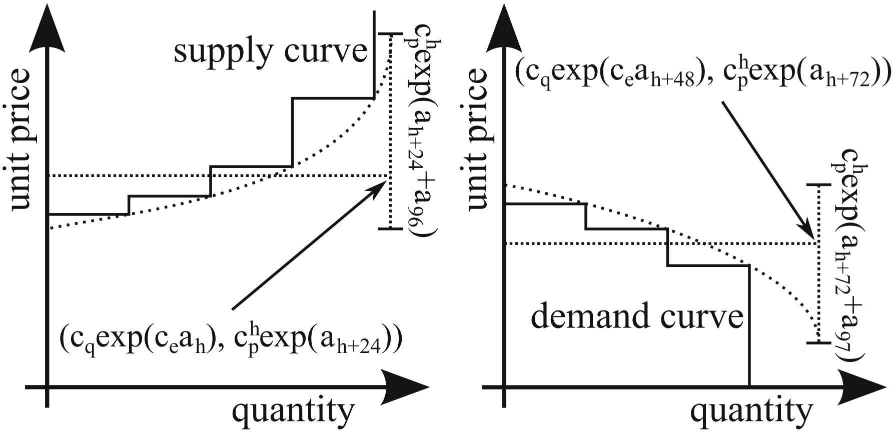

unit price
supply curve
$c_p^h\exp(a_{h+24}+a_{96})$
$(c_q\exp(c_e a_h), c_p^h\exp(a_{h+24}))$
quantity
unit price
$(c_q\exp(c_e a_{h+48}), c_p^h\exp(a_{h+72}))$
$c_p^h\exp(a_{h+72}+a_{97})$
demand curve
quantity

Fig. 2. Supply and demand defined by our proposed collections of bids.

4.4 Bidding Policy

In general in reinforcement learning, a *policy*, π, is a probability distribution of actions conditioned on states:

$$a_t \sim \pi(\cdot|s_t), \tag{9}$$

where s_t and a_t are, respectively, the state and the action at the instant t of discrete time.[2] We adopt a policy in the form

$$a_t = g^1(s_t;\theta) + \xi_t \circ \exp(g^2(s_t;\theta)), \quad \xi_t \sim \mathcal{N}(0, I), \tag{10}$$

where g^1 and g^2 are two vectors produced by the g neural network which is fed with the state s_t and parameterized by the vector θ of trained weights; "$\circ$" denotes the Hadamard (elementwise) product; ξ_t denotes random normal noise.

4.5 Bidding Policy Optimization with Reinforcement Learning

Participation in the day-ahead market can be naturally modeled as a Markov Decision Process in which the state, s_t, of the environment at time $t = 1, 2, \dots$ is a vector composed of two sub-vectors, *uncontrollable* variables s^u_t, and *controllable* variables s^c_t. The uncontrollable state variables denote external conditions like weather forecasts. They evolve according to an unknown stationary conditional probability

$$s^u_{t+1} \sim P(\cdot | s^u_t). \tag{11}$$

The controllable variables s^c_t are directly determined by the actions a_t taken and the uncontrollable state coordinates that is

$$s^c_{t+1} = f(s^c_t, a_t, s^u_t, s^u_{t+1}), \tag{12}$$

where f is known. The key controllable state variable is the power storage charge. It trivially results from the agent's bids (actions) and uncontrollable variables: market prices and the agent's own energy production and consumption.

The critical assumption that allows us to distinguish uncontrollable and controllable variables is that the trading agent is small enough not to impact the market prices. Therefore, we may simulate its bidding and determine whether the bids are executed based on the recorded market prices. If the agent was large enough to actually impact the market prices, then this simulation would not be realistic, at least without an elaborate model of the impact of this agent on the market prices.

Note that the above-defined division of state variables into controllable and uncontrollable is unusual. In a typical MDP, we assume that the state changes according to

$$s_{t+1} \sim P_s(\cdot | s_t, a_t), \tag{13}$$

where the conditional probability P_s may be quite difficult to analyze and estimate. Therefore, a strategy of choosing actions cannot be evaluated without bias within a simulation based on a model of P_s.

Based on a recorded trajectory of uncontrollable states, $(s^u_t : t = 1, \dots, T)$, we can designate a strategy of selecting actions a_t based on states s_t and evaluate this strategy in a simulation with the record $(s^u_t : t = 1, \dots, T)$ replayed.

---

[2] In Sect. 4.3, a_k denoted k-th coordinate of action and here a_t denotes action at the time t.

This valuation will be an unbiased estimate of the performance of this strategy deployed in reality. Furthermore, we can replay this record repeatedly and simulate episodes of on-line RL just using f (12) to designate consecutive values of s_t^c.

In order to optimize the strategy (10), we may use any algorithm of on-line reinforcement learning [33] e.g., A2C [22], PPO [31] or SAC [11]. In the experiments below, we used the A2C algorithm, which showed the best stability by far. Our comparison of RL algorithms is presented in Appendix G of the supplementary material. A training consists of a sequence of simulated trials in which the trajectory of uncontrollable states is just replayed from the data, and the corresponding trajectory of controllable states is designated based on the uncontrollable states, the actions selected, and the function f (12).

4.6 Alternative Bidding Strategies

In order to verify our proposed bidding strategy, we compare it to two more intuitive ones.

Simple Arbitrage Strategy. Perhaps the simplest conceivable bidding strategy is to buy energy when it is cheap, keep it in the battery, and sell it when it is expensive. On most days, the market value of electricity is the lowest at 2 am, and it is the highest at 10 am. Therefore, our reference simple arbitrage strategy assumes placing the two bids:

$$\langle \text{buy}, 2am, +\infty, \theta_1 - \widehat{l}\rangle, \quad \langle \text{sell}, 10am, -\infty, \theta_2\rangle, \tag{14}$$

where $\widehat{l}$ is an estimated storage state of charge at 0 am, and θ_1, θ_2 are optimized parameters. We apply the CMA-ES evolutionary algorithm [12] for their optimization.

Pair of Bids Strategy. A simple approach to bidding on the day-ahead electricity market, which also involves reinforcement learning, is to present just two bids for each hour $h = 0am, \ldots, 11pm$, namely

$$\langle \text{buy}, h, p_d^h, n_d^h q_0\rangle, \quad \langle \text{sell}, h, p_s^h, n_s^h q_0\rangle, \tag{15}$$

where p_d^h, n_d^h, p_s^h and n_s^h are defined by an action, $a \in [-1, 1]^{96}$, as follows:

$$n_d^h = \lfloor c_q \exp(c_e a_{h+48})/q_0 + 1/2\rfloor, \qquad p_d^{h,i} = c_p^h \exp(a_{h+72}), \tag{16}$$

$$n_s^h = \lfloor c_q \exp(c_e a_h)/q_0 + 1/2\rfloor, \qquad p_s^{h,i} = c_p^h \exp(a_{h+24}). \tag{17}$$

For comparison, see n_d^h (6), $p_d^{h,i}$ (7), n_s^h (4), $p_s^{h,i}$ (5). The collection of bids strategy introduced in Sect. 4.2 would be equivalent to (16) and (17), if all buy bids for a given hour had equal price and all sell bids for a given hour had equal price. In our simulations, we use the same reinforcement learning setup to train strategies that place the above pairs of bids and the collections of bids introduced in Sect. 4.3.

5 Simulations

5.1 Simulation Environment

Experiments are conducted using a custom environment simulating day-ahead energy market operations. This simulator is based on real-life data from the Polish market. It allows for customization of various market settings, such as a bid creation time, a scale of the trading agent (defined by the number of households), or its solar and wind energy generation capabilities. The environment is based on the Gymnasium environment interface [34], making it compatible with popular reinforcement learning libraries, including Stable-Baselines3 [29], which we use as our source of RL algorithms.

We provide details and parameters on the simulation environment, the trading agent's energy consumption and production profile, and weather forecast randomization in Appendices B–E of the supplementary material.

We run our experiments by replaying the events that occurred in the years 2016–2019. We selected this period as preceding the COVID-19 pandemic, which destabilized markets. The runs involve replaying original price data and weather data. In order to diversify every replay and thus avoid overfitting to the data, we randomize weather forecasts and electricity demand according to their statistical profile.

During the simulation, the trading agent may be forced to buy missing energy or sell excess energy immediately. It happens when the agent sells or uses energy it does not have or buys energy it does not have room for. The agent is being penalized for such events. Immediate buying is realized for double the current market price, and immediate selling is realized for half the current market price so that the agent has the incentive to better plan its bids instead of relying on instant buys or sells. Also, we do not include market entry and transaction fees, as they are fixed costs independent of the bidding strategy.

5.2 Experiments

Reinforcement learning is used to optimize the bidding policy for a collection of bids parameterized as in Sect. 4.3, later referred to as COLLECTION. It utilizes data from 2016 to the third quarter of 2018 as the training set, data from the fourth quarter of 2018 as the validation set, and data from 2019 as the testing set. The training is done in randomly generated intervals from the training set, which are 90 days long. Periodically, evaluation is done on a single validation interval 90 days long. After the training timesteps budget is depleted, the model for which the highest reward on validation interval was achieved is evaluated on the single testing interval 365 days long. Common parameters used for the RL experiments are available in Table 2 of the supplementary material.

The observation of the environment's state (117 values) is passed to the agent at bid placing time and contains the following information:

- prices of energy at the current day for every hour (24 values) – these are the prices for the current day, for which the bids were created the day before; the agent does not know energy prices for the bids currently submitted,

- current relative battery charge (1 value),
- estimated relative battery charge at midnight (1 value),
- one-hot encoded information about the current month (12 values),
- one-hot encoded information about the current day of the week (7 values),
- cloudiness, wind speed, and temperature forecasts for each hour of the next day (72 values).

Rewards are computed as

$$r_t = 10^{-3}\left(p_t - \bar{p}_t - \rho_t\right), \tag{18}$$

where p_t is the daily profit from selling and buying energy, $\bar{p}_t$ is a reference profit, and ρ_t is a regularizing penalty. The reference profit $\bar{p}_t$ is a daily profit that would be achieved if the difference between daily produced and consumed energy was sold or bought at the average market price from that day. The reference profit is not trivial to achieve since the agent mostly consumes energy when it is expensive and produces energy when it is cheap. The regularizing penalty

$$\rho_t = \sum_{i=1}^{\dim(a_t)} \left[|a_{t,i}| > 0.99\right] \tag{19}$$

where [*condition*] equals 1 if the condition is true, else 0, prevents the action coordinates from saturating at their bounds. The effect of regularization on the performance of tested strategies is presented and discussed in Appendix I of the supplementary material.

We compare the collection of bids strategy to the strategy to the alternative strategies presented in Sect. 4.6. The simple arbitrage strategy is later referred to as ARBITRAGE, and the pair of bids strategy is later referred to as PAIR.

We also applied the algorithm from [9], later referred to as FARL, which is a conceptually different approach to optimize a bidding strategy. FARL considers each day a 24-step episode and places a single sell/buy bid at each hour. FARL is based on the assumption that each bid is placed when market prices for preceding hours are known. This assumption is wrong for any day-ahead electricity market we are aware of. We used this algorithm to produce bids for consecutive hours without access to the market prices of previous biddings. We fed it with the same training, evaluation, and test data as discussed above. However, when used this way, it was unable to produce even remotely reasonable strategy. Implementation details, parameters, and discussion about the FARL algorithm are provided in Appendix F of the supplementary material.

5.3 Different Operation Scenarios

We tested the proposed collection of bids strategy in comparison to the alternatives from Sect. 4.6 in the following scenarios:

- an agent has an energy storage only (BES),
- an agent has an energy storage and production capabilities (BES+PROD),
- an agent has an energy storage and consumes energy (BES+CON),
- an agent has an energy storage, produces and consumes electricity (ALL).

5.4 Results

Table 1. Differences between achieved balances and the reference profit for the tested strategies in different scenarios; last column contains the reference.

Scenario\strategy	ARBITRAGE	PAIR	COLLECTION	Reference
BES	13251.29 ± 6238.36	30791.40 ± 851.31	**32826.71** ± 1127.88	0.00
BES+PROD	17578.67 ± 8039.00	28388.18 ± 765.45	**29170.47** ± 1630.18	37470.07
BES+CON	3446.76 ± 7042.91	28485.64 ± 667.74	**28547.43** ± 1154.93	−45089.87
ALL	16217.54 ± 6677.35	30203.35 ± 644.51	**31036.12** ± 1310.28	−7619.80

Table 1 presents differences between total profits achieved by tested strategies and the total reference profit described above; the last column contains the reference. It is seen that depending on the scenario, the reference varies a lot because the trading agent either sells the energy produced, buys the energy consumed or does both or neither. The proposed collection of bids strategy achieved the best profits, beating the pair of bids strategy in all tested scenarios. The pair of bids strategy achieved reasonable results but slightly worse than the proposed strategy.

Of all tested scenarios, the collection of bids achieved the best advantage over the pair of bids strategy in the battery-only scenario. Here, the agent earns money solely based on bids created, without any production or consumption to include in the bids. It is noticeable that the collection of bids strategy is able to adapt to these circumstances, making the biggest buys when the energy price is low and the biggest sells when the energy price is high, with some additional smaller transactions also happening in beneficial hours. This means that the collection of bids strategy is able to recognize significant price fluctuations, allowing it to capitalize on occasional prices.

In all of the tested scenarios, both strategies were able to adapt to the circumstances, buying enough energy when only consumption was active and selling surpluses of energy when only production was active. Immediate transactions due to lack or excess of energy were, in fact, very rare.

In Fig. 3, the mean hourly relative charge for the battery is presented. These were calculated for the COLLECTION strategy based on the test run that achieved the best profit. The proposed strategy is able to make the best use of its available capacities, with smooth transitions between hours, indicative of reasonable bid creation. It is seen that the battery is charged at night, which means that the agent buys energy when it is cheap. The battery is discharged at about 10 am, which means that the agent sells energy when it is the most expensive. The PAIR strategy is generally able to leverage that regularity and achieve reasonable profits. However, our proposed COLLECTION strategy is also able to leverage unpredictable variations of prices to the agent's benefit: It

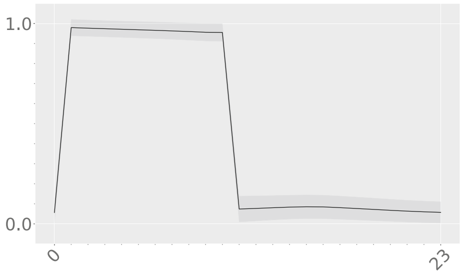


Fig. 3. Mean hourly relative battery charge. Strategy: COLLECTION. Scenario: ALL.

buys more when the prices are unexpectedly low and sells more when the prices are unexpectedly high.

The supplementary material attached to this paper contains the following:

- Appendix A - additional related works
- Appendix B - details and parameters of the simulation environment
- Appendix C - model of the trading agent's energy consumption
- Appendix D - model of the trading agent's energy production
- Appendix E - model for creating weather forecasts from real weather data
- Appendix F - description of adapting the FARL algorithm [9] to our simulation environment
- Appendix G - comparison of other RL algorithms (PPO, SAC, TD3) together with their hyperparameters
- Appendix H - detailed results for the pair of bids strategy
- Appendix I - study of using regularization in the collection of bids and the pair of bids strategies
- Plots for the collection of bids and the pair of bids strategies with different scenarios.

6 Conclusions

In this paper, we have proposed a parametrization of supply and demand curves, which allows for multiple sell and buy bids at each time, thus introducing increased flexibility and efficiency to automated trading on electronic markets. We have described a framework for optimization of this parametrized bidding strategy on a day-ahead energy market based on simulations and real-life data. We have used reinforcement learning to optimize this strategy and have compared it with different strategies. The proposed collection of bids strategy achieved the best results, getting the highest financial profit while showing reasonable behavior with battery management and bid placement.

The proposed strategy's generality and adaptability to data allow it to be deployed in real life. Indeed, the strategy is now being deployed in a system for energy storage management.

Data Availability. Data resources are available at the following link: https://github.com/Bestest96/ecml24_rl4trade.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Antonopoulos, I., et al.: Artificial intelligence and machine learning approaches to energy demand-side response: a systematic review. Renew. Sustain. Energy Rev. **130**, 109899 (2020)
2. Attaviriyanupap, P., Kita, H., Tanaka, E., Hasegawa, J.: New bidding strategy formulation for day-ahead energy and reserve markets based on evolutionary programming. Int. J. Electr. Power Energy Syst. **27**(3), 157–167 (2005)
3. Bakirtzis, A.G., Ziogos, N.P., Tellidou, A.C., Bakirtzis, G.A.: Electricity producer offering strategies in day-ahead energy market with step-wise offers. IEEE Trans. Power Syst. **22**(4), 1804–1818 (2007)
4. Bose, S., Kremers, E., Mengelkamp, E.M., Eberbach, J., Weinhardt, C.: Reinforcement learning in local energy markets. Energy Inform. **4**(1), 1–21 (2021)
5. Castellini, M., Di Corato, L., Moretto, M., Vergalli, S.: Energy exchange among heterogeneous prosumers under price uncertainty. Energy Econ. **104**, 105647 (2021)
6. Chen, S., Xu, Q., Zhang, L., Jin, Y., Li, W., Mo, L.: Model-based reinforcement learning for auto-bidding in display advertising. In: Autonomous Agents and Multiagent Systems (AAMAS) (2023)
7. Chen, T., Su, W.: Indirect customer-to-customer energy trading with reinforcement learning. IEEE Trans. Smart Grid **10**(4), 4338–4348 (2018)
8. Chen, T., Su, W.: Local energy trading behavior modeling with deep reinforcement learning. IEEE Access **6**, 62806–62814 (2018)
9. Dong, Y., Dong, Z., Zhao, T., Ding, Z.: A strategic day-ahead bidding strategy and operation for battery energy storage system by reinforcement learning. Electr. Power Syst. Res. **196**, 107229 (2021)
10. Gao, S., Wang, Y., Yang, X.: StockFormer: learning hybrid trading machines with predictive coding. In: International Joint Conference on Artificial Intelligence (IJCAI), pp. 4766–4774 (2023)
11. Haarnoja, T., Zhou, A., Abbeel, P., Levine, S.: Soft actor-critic: off-policy maximum entropy deep reinforcement learning with a stochastic actor. In: International Conference on Machine Learning (ICML), pp. 1861–1870 (2018)
12. Hansen, N.: The CMA evolution strategy: a tutorial. arXiv preprint arXiv:1604.00772 (2016)
13. Iria, J.P., Soares, F.J., Matos, M.A.: Trading small prosumers flexibility in the day-ahead energy market. In: 2017 IEEE Power & Energy Society General Meeting, pp. 1–5. IEEE (2017)
14. Jogunola, O., et al.: Consensus algorithms and deep reinforcement learning in energy market: a review. IEEE Internet Things J. **8**(6), 4211–4227 (2020)
15. Jogunola, O., Tsado, Y., Adebisi, B., Nawaz, R.: Trading strategy in a local energy market, a deep reinforcement learning approach. In: 2021 IEEE Electrical Power and Energy Conference (EPEC), pp. 347–352. IEEE (2021)
16. Kolmar, M.: Principles of Microeconomics. CCE, Springer, Cham (2022). https://doi.org/10.1007/978-3-030-78167-5

17. Lamont, J.W., Rajan, S.: Strategic bidding in an energy brokerage. IEEE Trans. Power Syst. **12**(4), 1729–1733 (1997)
18. Lee, N., Moon, J.: Transformer actor-critic with regularization: automated stock trading using reinforcement learning. In: Autonomous Agents and Multiagent Systems (AAMAS) (2023)
19. Liu, G., Xu, Y., Tomsovic, K.: Bidding strategy for microgrid in day-ahead market based on hybrid stochastic/robust optimization. IEEE Trans. Smart Grid **7**(1), 227–237 (2015)
20. Lu, R., Hong, S.H., Yu, M.: Demand response for home energy management using reinforcement learning and artificial neural network. IEEE Trans. Smart Grid **10**(6), 6629–6639 (2019)
21. May, R., Huang, P.: A multi-agent reinforcement learning approach for investigating and optimising peer-to-peer prosumer energy markets. Appl. Energy **334**, 120705 (2023)
22. Mnih, V., et al.: Asynchronous methods for deep reinforcement learning. arXiv:1602.01783 (2016)
23. Nanduri, V., Das, T.K.: A reinforcement learning model to assess market power under auction-based energy pricing. IEEE Trans. Power Syst. **22**(1), 85–95 (2007)
24. Okwuibe, G.C., Bhalodia, J., Gazafroudi, A.S., Brenner, T., Tzscheutschler, P., Hamacher, T.: Intelligent bidding strategies for prosumers in local energy markets based on reinforcement learning. IEEE Access **10**, 113275–113293 (2022)
25. Orfanoudakis, S., Chalkiadakis, G.: A novel aggregation framework for the efficient integration of distributed energy resources in the smart grid. In: Autonomous Agents and Multiagent Systems (AAMAS) (2023)
26. Perera, A., Kamalaruban, P.: Applications of reinforcement learning in energy systems. Renew. Sustain. Energy Rev. **137**, 110618 (2021)
27. Prabavathi, M., Gnanadass, R.: Energy bidding strategies for restructured electricity market. Int. J. Electr. Power Energy Syst. **64**, 956–966 (2015)
28. Qiu, D., Wang, J., Wang, J., Strbac, G.: Multi-agent reinforcement learning for automated peer-to-peer energy trading in double-side auction market. In: International Joint Conference on Artificial Intelligence (IJCAI), pp. 2913–2920 (2021)
29. Raffin, A., Hill, A., Gleave, A., Kanervisto, A., Ernestus, M., Dormann, N.: Stable-Baselines3: reliable reinforcement learning implementations. J. Mach. Learn. Res. **22**(268), 1–8 (2021)
30. Rahimiyan, M., Baringo, L.: Strategic bidding for a virtual power plant in the day-ahead and real-time markets: a price-taker robust optimization approach. IEEE Trans. Power Syst. **31**(4), 2676–2687 (2015)
31. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal policy optimization algorithms. arXiv:1707.06347 (2017)
32. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction, 2nd edn. The MIT Press, Cambridge (2018)
33. Sutton, R.S., Singh, S.P., McAllester, D.A.: Comparing policy-gradient algorithms (2001)
34. Towers, M., et al.: Gymnasium, March 2023. https://doi.org/10.5281/zenodo.8127026, https://zenodo.org/record/8127025
35. Vandael, S., Claessens, B., Ernst, D., Holvoet, T., Deconinck, G.: Reinforcement learning of heuristic EV fleet charging in a day-ahead electricity market. IEEE Trans. Smart Grid **6**(4), 1795–1805 (2015)
36. Vytelingum, P., Ramchurn, S.D., Voice, T.D., Rogers, A., Jennings, N.R.: Trading agents for the smart electricity grid. In: The Ninth International Conference on

Autonomous Agents and Multiagent Systems (AAMAS 2010) (10–14 May 2010), pp. 897–904 (2010). https://eprints.soton.ac.uk/268361/, Event Dates: 10–14 May 2010
37. Wen, F., David, A.: Strategic bidding for electricity supply in a day-ahead energy market. Electr. Power Syst. Res. **59**(3), 197–206 (2001)
38. Yang, M., Zhu, M., Liang, Q., Zheng, X., Wang, M.: Spotlight news driven quantitative trading based on trajectory optimization. In: International Joint Conference on Artificial Intelligence (IJCAI), pp. 4930–4939 (2023)
39. Yang, T., Zhao, L., Li, W., Zomaya, A.Y.: Reinforcement learning in sustainable energy and electric systems: a survey. Annu. Rev. Control. **49**, 145–163 (2020)
40. Zhong, Y., Bergstrom, Y., Ward, A.: Data-driven market-making via model-free learning. In: International Joint Conference on Artificial Intelligence (IJCAI), pp. 4461–4468 (2020)
41. Xu, Z., Cheng, X., He, Y.: Performance of deep reinforcement learning for high frequency market making on actual tick data. In: Autonomous Agents and Multiagent Systems (AAMAS) (2022)
42. Çam, E.: Electricity 2024 - analysis and forecast to 2026. Technical report, International Energy Agency (2024)

Spatial Transfer Learning for Estimating $PM_{2.5}$ in Data-Poor Regions

Shrey Gupta[1(✉)], Yongbee Park[4], Jianzhao Bi[2], Suyash Gupta[3], Andreas Züfle[1], Avani Wildani[1,5], and Yang Liu[1(✉)]

[1] Emory University, Atlanta, USA
`{shrey.gupta,azufle,yang.liu}@emory.edu`
[2] University of Washington, Seattle, USA
`jbi6@uw.edu`
[3] University of California, Berkeley, Berkeley, USA
`suyash.gupta@berkeley.edu`
[4] Ingkle, Cheonan-si, South Korea
`yongbee.park@ingkle.com`
[5] Cloudflare, San Francisco, USA

Abstract. Air pollution, especially particulate matter 2.5 ($PM_{2.5}$), is a pressing concern for public health and is difficult to estimate in developing countries (*data-poor* regions) due to a lack of ground sensors. Transfer learning models can be leveraged to solve this problem, as they use alternate data sources to gain knowledge (i.e., data from *data-rich* regions). However, current transfer learning methodologies do not account for dependencies between the source and the target domains. We recognize this transfer problem as *spatial* transfer learning and propose a new feature named *Latent Dependency Factor* (LDF) that captures spatial and semantic dependencies of both domains and is subsequently added to the feature spaces of the domains. We generate LDF using a novel two-stage autoencoder model that learns from clusters of similar source and target domain data. Our experiments show that transfer learning models using LDF have a 19.34% improvement over the baselines. We additionally support our experiments with qualitative findings.

Keywords: Spatial transfer learning · Autoencoder model · $PM_{2.5}$ data

1 Introduction

Air pollution, especially atmospheric aerosols smaller than 2.5 µm *i.e.* $PM_{2.5}$ poses a significant concern to public health [30]. Emissions from vehicles [20], wildfires [7], and industrial processes [11] are major contributors to high $PM_{2.5}$ levels. Current approaches for measuring $PM_{2.5}$ involves using either remote

---

S. Gupta and Y. Park—Contributed equally to this work.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-70378-2_24.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
A. Bifet et al. (Eds.): ECML PKDD 2024, LNAI 14949, pp. 385–400, 2024.
https://doi.org/10.1007/978-3-031-70378-2_24

sensing methodologies [3] or ground sensors [1]. While satellite-based remote sensing methodologies are a low-cost way to measure $PM_{2.5}$, however, their data collection is affected by factors like cloudy weather and high surface reflectance, thereby significantly reducing the accuracy of measured $PM_{2.5}$ levels [3]. Alternatively, installing $PM_{2.5}$ ground sensors yields highly accurate data as these sensors employ gravimetric data collection methodologies [1]. However, due to their high installation and maintenance costs [21], it is challenging to scale them in developing countries [8], creating an imbalance of *data-rich* (developed) and *data-poor* (developing) regions with $PM_{2.5}$ data for air pollution estimation.

Transfer learning (TL) can ameliorate this situation by utilizing *data-rich* (source data) regions to learn a prediction model on *data-poor* (target data) regions [26]. Prior research on estimating $PM_{2.5}$ through TL is geared towards time-series forecasting where the model learns historical data of an observed location (sensors) and forecasts the horizon (*i.e.* future values) for the observed locations [12,24,37,38]. Therefore, these models cannot estimate the $PM_{2.5}$ levels for locations where historical data is unavailable [33]. Alternatively, one can employ *Instance transfer learning* (ITL) models that avoid the limitations of time-series forecasting models by not relying on continuous temporal data [13,14]. ITL models reweigh source domain samples based on the target domain and subsequently combine the two domains.

Unfortunately, ITL models are limited in estimating $PM_{2.5}$ as they overlook the *spatial and semantic correlations* in the datasets. $PM_{2.5}$ estimation data is uniquely heterogeneous and complex, containing topographical, meteorological, and geographical features. These features exhibit *spatial* autocorrelations (dependencies), *i.e.* nearby locations tend to have similar $PM_{2.5}$ levels, as well as *semantic* correlations (dependencies), *e.g.* locations with similar meteorological and topographical conditions exhibit similar $PM_{2.5}$ levels with high likelihood [22]. Spatial dependencies are prevalent within a domain, whereas semantic dependencies will likely arise when combining two domains (case for ITL). We call this complex transfer problem as *spatial* transfer learning.

In this paper, we solve *spatial* transfer learning to improve $PM_{2.5}$ estimation by allowing source and target data points to learn from each other in the combined domain space. We achieve this by introducing a new feature called *Latent Dependency Factor* (LDF) in both the source and target datasets to bridge the gap between the two domains. To generate LDF, we first learn a cluster of similar (spatially and semantically similar) data points for each sample, which are fed to our novel two-stage autoencoder model. The first stage, *encoder-decoder*, aims to learn a latent representation from the combined feature space of the cluster, while the second stage, *encoder-estimator*, learns from the target label ($PM_{2.5}$ value). The LDF is highly correlated to the target (dependent) variable and contains learned dependencies from both domains. To illustrate the benefits of LDF, we utilize real-world $PM_{2.5}$ data for the United States and Lima city in Peru. Our experiments include a comparative analysis of ML and TL models within the US boundaries, where we observe a 19.34% improvement in prediction accuracy over baseline models. We also present a qualitative analysis showcasing the deployment of our methodology in *data-poor* regions such as Lima. In summary, we make the following contributions:

1. We present *Latent Dependency Factor* (LDF), a new feature to learn the spatial and semantic dependencies within the combined source and target domains and close the gap between the two domains.
2. We introduce a novel two-stage autoencoder model to generate LDF. It learns dependencies from the combined feature space of the clustered input data and the dependent variable.
3. We explore the settings for *spatial* transfer learning for $PM_{2.5}$ estimation in data-poor regions, a challenging problem with untrained test locations and sparse target and source locations causing minimal spatial autocorrelation.
4. We deploy our technique in Lima, Peru, and validate the results by domain experts due to the scarcity of true labels. This offers insights into the real-world application of our technique and its effectiveness.

2 Related Work

Estimating $\mathbf{PM_{2.5}}$ via Transfer Learning. Prior studies have utilized transfer learning for $PM_{2.5}$ estimation through time-series forecasting models, which learn from historical data of target sensors to forecast their future values. Fong et al. [12] incorporate Recurrent Neural Networks (RNN) in their temporal transfer learning model. Yao et al. [38] employ Variational Auto-Encoders (VAE) using nearby sensors as source data, while Ma et al. [24] combine Long-Short Term Memory (LSTM) and RNN to forecast long-range $PM_{2.5}$ levels from short-range historical data. Yadav et al. [37] leverage low-cost sensors as source data for temporal transfer learning for ground sensors. However, our problem is not suited for time-series forecasting due to missing temporal points and the lack of temporal matching between regions with varying meteorological conditions.

Transfer Learning via Feature Augmentation. Previous studies have improved model predictions by imputing features from another dataset [21,23] or generating synthetic samples to augment data [18,34,35]. The former leverages datasets with low marginal distribution, while the latter focuses on augmenting samples rather than features. In the domain of transfer learning, Daume et al. [6] and Duan et al. [10] introduce domain adaptation models—Feature Augmentation Method (FAM) and Heterogeneous Feature Augmentation (HFM), respectively—to create a common feature space using source and target features. These models are useful when the source and target domains have a dissimilar feature space, as noted by Pan et al. [25], whereas our approach incorporates spatial and semantic dependencies during ITL for domains with similar feature spaces, high marginal distribution, and low spatial autocorrelation.

3 Problem Formulation

Our problem comprises the source region with higher $PM_{2.5}$ sensors and the target region with fewer sensors. The data is heterogeneous due to diverse features and complex due to spatial and semantic dependencies between its samples.

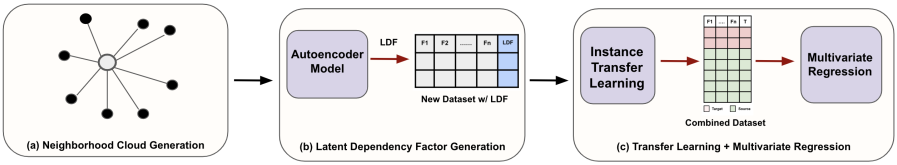

(a) Neighborhood Cloud Generation
Autoencoder Model
LDF
F1
F2
......
Fn
LDF
New Dataset w/ LDF
(b) Latent Dependency Factor Generation
Instance Transfer Learning
F1
....
Fn
T
Target
Source
Combined Dataset
Multivariate Regression
(c) Transfer Learning + Multivariate Regression

Fig. 1. Framework for *spatial* transfer learning via *Latent Dependency Factor*

Let X_f^S be the feature set for the source domain with m samples, and let X_f^T be the feature set for the target domain with n samples, such that $m >> n$, and contains f features. Let Y^S and Y^T be the source and target domain labels (PM$_{2.5}$ levels). Hence, $D^S = (x_i^S, y_i^S)_{i=1}^m$ is the source domain dataset, where $x_i^S \in X_f^S$ is the feature vector for the i-th PM$_{2.5}$ monitor, and $y_i^S \in Y_S$ is the corresponding PM$_{2.5}$ value at the sensor. Similarly, $D^T = (x_i^T, y_i^T)_{i=1}^n$ is the target domain dataset with x_i^T and y_i^T representing i-th monitor and its PM$_{2.5}$ value, respectively.

Instance Transfer Learning (ITL) methodologies are employed when the two domains have varying marginal distributions. They find a reweighing function $w(x)$ that adjusts the importance of each sample in the source domain based on its relevance to the target domain. The importance weights $w(x_i^S)$ are calculated for each sample x_i^S in the source domain D^S, where $w(x_i^S)$ represents the degree of relevance of x_i^S to the target domain D^T. This degree of relevance is often calculated using probability densities, expressed as $w(x_i^S) = \frac{P_{D^T}(x_i^S)}{P_{D^S}(x_i^S)}$, where $P_{D^T}(x_i^S)$ and $P_{D^S}(x_i^S)$ is the probability density of x_i^S in the target domain and source domain respectively. The importance weights are applied to the source domain samples to obtain $\bar{D}_S = (\bar{x}_i^S, y_i^S)_{i=1}^m$ where $\bar{x}_i^S = w(x_i^S) \cdot x_i^S$. The reweighed source domain samples are used in the target domain for training; the combined domain is represented as $D^{\bar{S}T} = (x_i^{\bar{S}T}, y_i^{\bar{S}T})_{i=1}^{m+n}$.

Our goal is to improve the estimation of PM$_{2.5}$, such that the combined domain $D^{\bar{S}T}$ after reweighing source domain data D^S successfully captures the spatial and semantic dependencies.

4 Methodology

We introduce *Latent Dependency Factor* (LDF), a new feature imputed in the dataset to achieve *spatial* transfer learning for PM$_{2.5}$ estimation. The LDF has the following attributes: (1) It is highly correlated to the observed variable (PM$_{2.5}$ value), (2) It captures the spatial dependencies (spatial autocorrelation between nearby locations), (3) It captures the semantic dependencies (semantic correlation in the combined data).

Imputing a new feature allows to learn a new loss function. Hence, if a function $f : X_f^{\bar{S}T} \rightarrow Y_T$ can predict the missing PM$_{2.5}$ values in the target domain D_T. Then, f is learned by minimizing the empirical risk as,

$$\min_f \left[\frac{1}{m+n}\sum_{i=1}^{m+n} \ell(y_i^{\bar{S}T}, f(x_i^{\bar{S}T})) + \lambda \cdot \Omega(f)\right] \tag{1}$$

where $\ell(y, \hat{y})$ is the loss calculated between true PM$_{2.5}$ value (y) and predicted value ($\hat{y}$) (here $f(x_i^{\bar{S}T})$), $\Omega(f)$ is a regularization term, and λ controls the trade-off between the empirical risk and model complexity. When a new feature is imputed, the empirical risk in (1) is transformed as,

$$\min_f \left[\frac{1}{m+n}\sum_{i=1}^{m+n} \widetilde{\ell}(y_i^{\bar{S}T}, \widetilde{f}(x_i^{\bar{S}T})) + \lambda \cdot \Omega(\widetilde{f})\right] \tag{2}$$

with the new trained regressor, $\widetilde{f}$ and loss function $\widetilde{\ell}$. *Hence, the new loss function allows obtaining a lower minimum.* The framework for *spatial* transfer learning via LDF contains 3 stages, as shown in Fig. 1, which we elaborate further.

4.1 Neighborhood Cloud Generation

The first stage (Fig. 1(a)) generates a neighborhood cloud of k similar data points for each sensor in the source and target regions. This cloud is training data for the two-stage autoencoder model, allowing each sensor to learn the spatial dependencies of its neighbors and semantic dependencies between the two domains. The similarity between data points (sensors) is calculated by minimizing the $||L||_2$ distance across geographical, topographical, and meteorological features (see supplementary).

4.2 Generating Latent Dependency Factor (LDF)

After generating the neighborhood cloud, the subsequent steps involve generating the LDF, imputed as a new feature into the original dataset. This feature is derived using a two-stage autoencoder model (Fig. 2(a)), where the input dataset (neighborhood cloud) utilizes features – topographical, meteorological, geographical, and PM$_{2.5}$ levels. We believe these predictors influence the PM$_{2.5}$ levels at the objective location (centroid of the cluster). *E.g.*, given a sensor location, l_i, in the target region, the predictors such as the *wind-direction*, *elevation*, *population*, and more, for the surrounding sensors can influence the PM$_{2.5}$ levels at l_i (spatial autocorrelation). Additionally, the sensor location, l_i, can be semantically correlated to another location, l_j, in the source region, influencing the PM$_{2.5}$ levels at l_i in the combined dataset. In Fig. 2(a), each sensor has $(p+1)$ features with p features and a label. We first calculate the weight for each feature. This is achieved by finding the similarity (inverse distance) between the feature of the objective location and neighboring sensors. This allows sensors with influential features to be given more importance. Following the weighing, the features from m sensors are stacked together with the objective location to generate the input data of size $(m+1) \cdot (p+1)$. The PM$_{2.5}$ for the objective location is voided by setting it to 0. This high-dimensional data is summarized into the LDF, using the two-stage autoencoder model shown in Fig. 2(b).

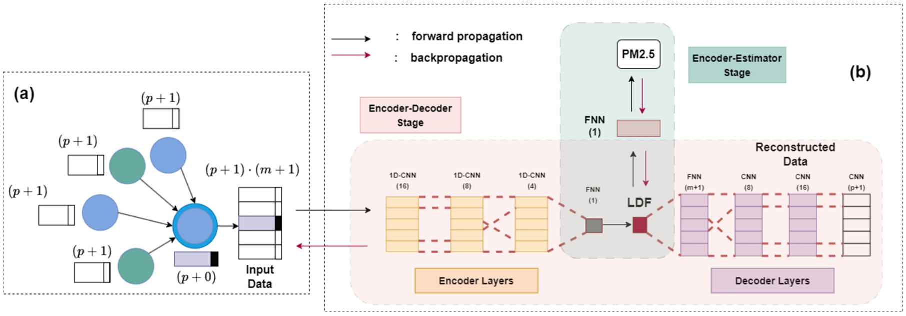


Fig. 2. Two-stage autoencoder model for generating LDF.

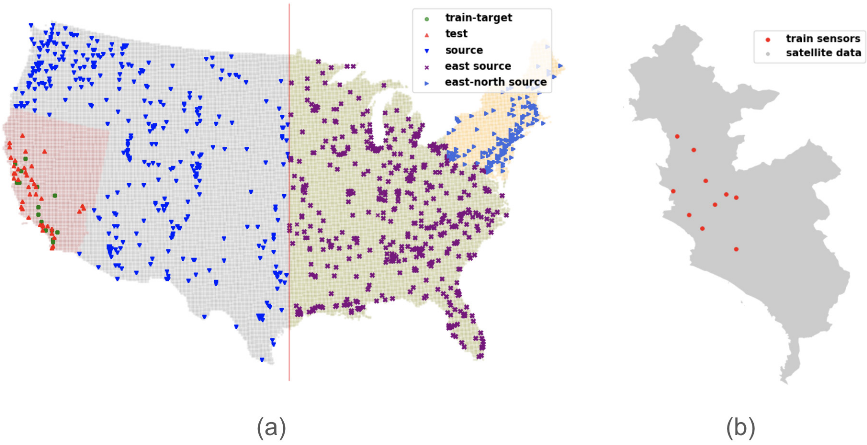
Encoder-Decoder Stage. The *encoder-decoder* stage of the two-stage autoencoder model is similar to the standard autoencoder model, where the encoder first summarizes the input data to generate a latent value. The decoder employs backpropagation to train the autoencoder. The encoder and the decoder have three 1D-CNN layers with varying filter sizes, as shown in Fig. 2(b). For the encoder, the kernel size of the first 2 CNN layers is chosen as 1 to achieve individual attention for each sensor and amplify the effectiveness of information summarization [19]. The third CNN layer has a kernel size 3 to retain the condensed pattern from multiple stations. Finally, the information is summed up using an FNN layer, which outputs the latent value, *i.e.*, the LDF value.

Encoder-Estimator Stage. Since the input data consists of multiple features, we increase the attention on $PM_{2.5}$ labels using the *encoder-estimator* stage. The estimator layer takes the encoded LDF value as input. It has a single FNN layer with a single weight and bias set. It utilizes back-propagation and $PM_{2.5}$ value of the objective location to train the encoder-decoder model and consequently optimize the LDF generation process. The autoencoder stages alternate training over the epochs. We also explore extending LDF to include Aerosol Optical Depth (AOD) [29] feature, which we call LDF-A and which measures the aerial density of aerosols such as smoke, dust, and PM particles, in the *encoder-esitmator* stage.

4.3 Transfer Learning and Multivariate Regression

In Fig. 1(c), we employ Instance Transfer Learning (ITL) to mitigate discrepancies between source and target domain samples [13]. This involves reweighing the source domain samples to align them closer to the target domain. The reweighed source data is combined with the target data, creating a unified dataset reflecting both domains' characteristics.

This combined dataset is subsequently used to train a multivariate regressor for predicting $PM_{2.5}$ values. The choice of regressor can range as *polynomial-function* based, *decision-tree* based, or *ensemble* model. We employ an *ensemble* regressor for our framework, given their high prediction accuracy [9].



(a) (b)

Fig. 3. (a) US PM$_{2.5}$ ground sensors. The points in the pink target region represent sample training (green) and testing (red) sensors. The green and yellow regions represent the eastern and north-eastern source regions, respectively. (b) PM$_{2.5}$ sensors in Lima, Peru. Red points represent sensors used for training, and the grey area represents satellite data for testing. (Color figure online)

5 Evaluation

5.1 Datasets

We employ existing PM$_{2.5}$ datasets of two distinct regions: the US [27] and the Lima, Peru [36]. In comparison to other datasets [5], these corpora draw from diverse sources (EPA, NLDAS-2, and NED for the US and SENAMHI and JHU for Lima) and encompass a wide array of heterogeneous features such as *wind patterns*, *atmospheric pressure*, *humidity levels*, *potential energy* and more.

United States Dataset. As the US region has abundant PM$_{2.5}$ sensors, we select this dataset to simulate a transfer learning scenario within its geographical boundaries. The US dataset has daily averaged PM$_{2.5}$ levels for 2011 using 1081 sensors, as shown in Fig. 3, with over 249k samples and 77 features. Although the sample size should be 1081×365, some sensors were inactive on certain days (daily average active sensors: ~ 682). This contributes to missing temporal points in the dataset, which limits the application of time-series forecasting methodologies. We follow the prior work [27] and use Layerwise Relevance Propagation [2] to extract 27 meteorological, topographical, and geographical features. As illustrated in Fig. 3(a), we select two source regions, the eastern US (highlighted green; marker: $\mathbf{x}$) and north-eastern US (highlighted yellow; marker: ►) and a target region, California-Nevada (highlighted pink). Prior works [4] show that the California-Nevada region has a diverse landscape compared to the remaining US, thereby simulating a TL scenario with distribution shift and low spatial correlation among the two domains.

We sample the 128 target region sensors into sets of 5, 7, 9, and 11 sensors to have fewer samples. The remaining sensors are used for testing. For cross-validation (CV), we use 20 random samples per sensor. We extrapolate the active sensors per day and generate a neighborhood cloud for each sensor that includes both source and target sensors. Next, the clustered data is used to generate the LDF which is fed to the transfer models. Our reported R^2 and RMSE values represent averages across the 20 CVs. The features are normalized before model training. For qualitative analysis (Sect. 5.5), we use $\sim$ 19.5 million unlabeled satellite data samples from the California-Nevada region.

Lima Dataset. Given the dearth of sensors in the Lima data, it is a use-case of real-world transfer learning, where the source data is the complete US dataset (249k+ samples, 27 features). Lima region has 10 $PM_{2.5}$ sensors, as shown in Fig. 3(b), with 2419 samples and 21 features for the year 2016. Lima and the US datasets have only 14 common features (see supplementary material). For the qualitative analysis, the Lima satellite data contains 5959 samples covering the entire Lima region, as shown in Fig. 3(b) (highlighted grey). We use all 10 sensors and the US dataset to construct the neighborhood cloud data such that each *day of the year (doy)* between the two datasets (e.g., day 17 in Lima matched with day 17 in the US) are extrapolated to generate the clusters. However, the matching of *doy* is not on the same year or season between the two domains to have a real-world transfer condition with minimal alignment.

5.2 Prediction Models

Machine Learning (ML) Models. We select two popular ML models, **Random Forest Regressor (RF)** [41] and **Gradient Boosting Regressor (GBR)** [40], trained on only the target region data and tested on the remaining test data. The RF and GBR have parameters varied as *n-estimators:* {100, 400, 1000}, *max-depth*: {4, 8, inf} with *max-leaf-node*: {4, 8, inf} for RF and *learning-rate*: {0.1, 0.5, 1.0} for GBR, to get the best fit.

Transfer Learning (TL) Models. We select competitive ITL models [16,31,42] for the regression task and train them on target and source region data.

1. **Nearest Neighbor Weighing (NNW)**: The NNW [42] reweighs the source samples by creating a Voronoi tessellation for each sample and counts the number of target samples falling inside it. The model parameters are varied as: *neighbors*: {6, 8, 10} and *n-estimator*: Decision Tree Regressor with *depth*: {6, 8, inf} to get the best fit.
2. **Kullback-Leibler Importance Estimation Procedure (KLIEP)**: The KLIEP reweighs the source samples to minimize the KL divergence between the source and target domains [31]. The model parameters are varied as: *kernel*: {rbf, poly}, *gamma*: {0.1, 0.5, 1.0}, and *n-estimator*: Decision Tree Regressor with *depth*: {6, 8, inf} to get the best fit.

3. **Kernel Mean Matching (KMM)**: The KMM reweighs the source samples such that means of source and target samples in reproducible kernel Hilbert space is minimized [16]. The model parameters are varied as: *kernel*: {rbf, poly}, *gamma*: {0.1, 0.5, 1.0}, and *n-estimator*: Decision Tree Regressor with *depth*: {6, 8, inf} to get the best fit.
4. **Fully-connected Neural Network (FNN)**: The FNN transfer model, although not an ITL model, is utilized to validate the performance of non-ITL models on the $PM_{2.5}$ data. It uses 3 fully connected layers: *nodes*: 128, *activation-function*: Relu, and 1 final layer with a single node and a linear activation function. It was trained on LDF-imputed source data and transferred by fine-tuning over LDF-imputed target data.

The TL models are trained on data sans LDF, LDF, and LDF-A-imputed data. We use the GBR model as the multivariate regressor to predict $PM_{2.5}$, with parameters varied as: *estimators:* {100, 400, 1000}, *max-depth*: {4, 8, inf}, *max-leaf-node*: {4, 8, inf}, and *learning-rate*: {0.1, 0.5, 1.0} to get the best fit. The source code, datasets, and final hyperparameter values are available at: https://github.com/YongbeeIngkle/spatial-transfer-learning.git.

5.3 Optimal *k* for Neighborhood Cloud

In Fig. 5(a), we use the eastern US as source data and vary the size of the neighborhood cloud (k) for the NNW [LDF] model as {4, 8, 12, 16}. Our choice of k mimicked optimizing parameters, ceasing at 16 due to high computational costs. We observe that $k = 4$ has the worst performance, while for the remaining values, there is no observable difference for sensors ≥ 9. For sensors ≤ 9, $k = 12$ has the most optimal performance. Hence, we chose $k = 12$ to optimize the computation and generalizability of the model.

5.4 Results and Analysis

In Table 1 and Table 2, we compare the performance of various models with the eastern US and the north-eastern US as source datasets, respectively.

Eastern US as Source Data. First, we compare the ML and TL sans LDF models. In Table 1, we observe that NNW, KLIEP, and KMM have a positive transfer (improved accuracy), with NNW having the best performance. We observe an unpredictable performance for the FNN transfer model, validating that non-ITL models are less suited for such transfer problems. Next, we illustrate the impact of the *Latent Dependency Factor* (LDF) on TL models. We observe an improvement in estimation accuracy for NNW, KLIEP, and KMM (for both LDF and LDF-A), where NNW [LDF] is the best-performing model. For the FNN model, LDF has no notable effect as it caters to only ITL models. The high performance of NNW is due to the Voronoi tesselation neighborhood it uses for reweighing source samples. This allows it to capture similar data points in its neighbor, a spatially preferred reweighing for the $PM_{2.5}$ data.

Table 1. Source: Eastern US (best highlighted; second-best underlined)

Model	**Sensors**							
	5		**7**		**9**		**11**	
	R^2	**RMSE**	**R^2**	**RMSE**	**R^2**	**RMSE**	**R^2**	**RMSE**
RF	−0.082	8.855	0.002	8.565	0.066	8.387	0.071	8.311
GBR	−0.061	8.684	0.064	8.210	0.177	7.857	0.157	7.891
NNW	0.236	7.563	0.263	7.447	0.280	7.406	0.296	7.288
KLIEP	0.155	7.960	0.192	7.801	0.200	7.811	0.222	7.666
KMM	0.197	7.757	0.226	7.634	0.242	7.601	0.258	7.479
FNN	−0.064	8.818	−0.350	9.715	0.009	8.629	−0.039	8.765
NNW [LDF]	**0.247**	**7.494**	**0.336**	**7.061**	**0.378**	**6.874**	**0.378**	**6.838**
NNW [LDF-A]	0.225	7.596	0.298	7.230	0.359	6.973	0.359	6.924
KLIEP [LDF]	0.202	7.724	0.278	7.370	0.325	7.173	0.336	7.073
KLIEP [LDF-A]	0.232	7.584	0.267	7.427	0.319	7.201	0.330	7.100
KMM [LDF]	0.210	7.671	0.302	7.236	0.353	7.013	0.352	6.971
KMM [LDF-A]	0.196	7.723	0.295	7.277	0.330	7.134	0.333	7.067
FNN [LDF]	−0.255	9.532	−0.141	9.082	0.072	8.374	0.087	8.236
FNN [LDF-A]	−0.150	9.146	−0.105	8.990	0.091	8.275	0.078	8.287

North-Eastern US as Source Data. In Table 2, we observe a positive transfer for NNW and KLIEP models, with NNW having the best performance. KMM shows a negative transfer [28] due to the high marginal distribution present between the target and source datasets [17]; unable to be minimized in reproducing kernel Hilbert space (RKHS) [16]. Like earlier, the FNN transfer model has an unpredictable performance. When the LDF is introduced, we observe an improvement in estimation accuracy for NNW and KLIEP models. NNW [LDF] and NNW [LDF-A] are the best-performing models. KMM [LDF-A] shows improvement for more sensors ($\geq$11). As expected, the FNN models using LDF and LDF-A show no improvement.

5.5 Qualitative Analysis

While improving prediction accuracy is crucial, visualizing $PM_{2.5}$ patterns on geo-maps is also valuable. We visualize $PM_{2.5}$ estimations for the California-Nevada region and the Lima, Peru region in Fig. 4(a) and Fig. 4(b), respectively. For this analysis, we need a ground truth against which all the models can be compared. We use the GBR model, trained on all 128 monitors (249k+ samples) and estimated on the unlabeled satellite data ($\sim$ 19.5 M samples), and use its predicted geo-map as the assumed ground truth for verification. We use 9 sensors and the eastern US as source data for transfer models (NNW, NNW[LDF], NNW[LDF-A]). For Lima, we use all 10 sensors and the eastern US as the source

Table 2. Source: North Eastern US (best highlighted; second-best: underlined)

	Sensors							
	5		**7**		**9**		**11**	
Model	**R^2**	**RMSE**	**R^2**	**RMSE**	**R^2**	**RMSE**	**R^2**	**RMSE**
RF	−0.082	8.855	0.002	8.565	0.066	8.387	0.071	8.311
GBR	−0.061	8.684	0.064	8.210	0.177	7.857	0.157	7.891
NNW	0.199	7.732	0.294	7.286	0.301	7.297	0.298	7.257
KLIEP	0.098	8.180	0.219	7.650	0.263	7.494	0.270	7.408
KMM	−0.142	9.053	−0.070	8.809	0.232	7.640	0.246	7.526
FNN	0.022	8.448	−0.006	8.598	0.091	8.266	0.078	8.307
NNW [LDF]	**0.225**	**7.592**	0.317	7.157	0.376	6.886	**0.392**	**6.751**
NNW [LDF-A]	0.201	7.702	**0.320**	**7.122**	**0.378**	**6.873**	0.374	6.847
KLIEP [LDF]	0.164	7.889	0.275	7.363	0.353	7.011	0.360	6.924
KLIEP [LDF-A]	0.170	7.860	0.270	7.396	0.342	7.068	0.348	6.991
KMM [LDF]	−0.265	9.409	0.009	8.468	0.188	7.749	0.257	7.389
KMM [LDF-A]	−0.152	9.042	−0.029	8.566	0.172	7.845	0.288	7.260
FNN [LDF]	0.036	8.429	−0.052	8.761	0.131	8.061	0.237	7.566
FNN [LDF-A]	−0.060	8.774	0.045	8.390	0.159	7.983	0.207	7.708

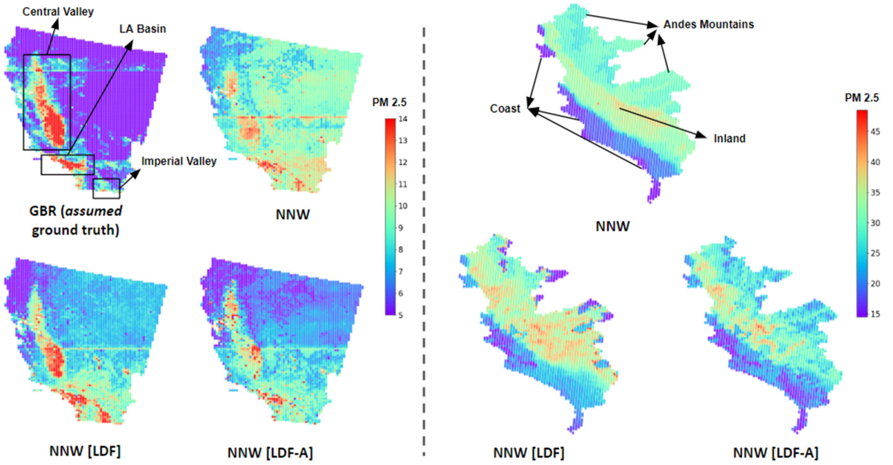
data for the three transfer models. The true labels for Lima were unavailable, and domain experts (environmental scientists) were consulted for analysis.

Due to the scarcity of target domain data, this qualitative analysis aims to observe if transfer models successfully capture glaring PM$_{2.5}$ estimation patterns.

California-Nevada Region. In Fig. 4(a), we observe that the NNW [LDF] model has the most accurate PM$_{2.5}$ estimation in the hotspots (solid boxes in the GBR map). It accurately captures patterns in the *Central Valley* and the *Los Angeles Basin* but overestimates in the *Imperial Valley.* NNW [LDF-A] has the second-best performance but has a patchy estimation in the *Central Valley.* For NNW, we observe obscure patterns that are patchy and underestimated in the *Central Valley* and highly overestimated in the *Imperial Valley.*

Lima Region. In Fig. 4(b), all models exhibit lower PM$_{2.5}$ levels near the *coast* and higher levels moving *inland*, a pattern validated by domain experts. However, NNW [LDF] has a clearer concentration gradient of *inland* PM$_{2.5}$ compared to the other models. Near the *Andes mountain ranges*, the PM$_{2.5}$ is the lowest, which the NNW[LDF] model accurately captures but slightly and highly overestimated by the NNW [LDF-A] and NNW models, respectively. These observations confirm the improvement of prediction by LDF-based TL models.

Additionally, we performed a 60:40 train-test split on the Lima sensors and trained NNW and NNW [LDF] TL models using 3-fold cross-validation, with



(a) California-Nevada

(b) Lima

Fig. 4. (a) Annual mean $PM_{2.5}$ prediction for *California-Nevada*, trained using GBR and NNW with and without LDF features (9 sensors). (b) Annual mean $PM_{2.5}$ prediction for Lima region trained using NNW models.

the complete US as the source data. The results for [R^2, RMSE] for NNW and NNW[LDF], respectively, were [0.476, 9.852] and [0.558, 9.091]. Hence, NNW [LDF] outperforms NNW, validating the quantitative analysis.

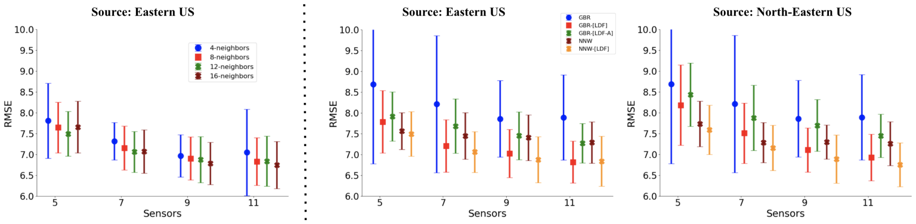
5.6 Ablation Study

For the ablation study, we use GBR instead of ITL models to validate the performance of non-transfer models using LDF-imputed data. Figure 5(b) shows the comparison between GBR [LDF], GBR [LDF-A], GBR (target only), NNW, and NNW [LDF]. For both the eastern US and the north-eastern US as source data, GBR [LDF] is the second-best performing model. Though it doesn't outperform NNW [LDF], the improved predictions highlight LDF's effectiveness.

The performance of FNN [LDF] and FNN [LDF-A] in Table 1 and Table 2 further tests LDF with non-ITL models, confirming that LDF is effective with ITL and multivariate regression models but not other transfer models.

6 Discussion

While the evaluation results show the improvement using the LDF, we further analyze the correlation between LDF and PM2.5, as shown in Table 3, where LDF demonstrates the highest correlation with the dependent variable, indicating strong predictive power and feature importance [15]. This experiment uses an LDF-imputed dataset of 10 target sensors and eastern US source data.



(a) Optimal 'K' for neighborhood cloud **(b) Ablation Study**

Fig. 5. (a) Comparing performance of NNW [LDF] model when neighborhood cloud uses k = {4, 8, 12, 16} neighbors. (b) Ablation study comparing the performance of GBR, GBR [LDF], GBR [LDF-A], NNW, and NNW [LDF] models.

Table 3. Most correlated features (5) to $PM_{2.5}$ variable.

Method	**LDF**	**Pressfc**	**Dswrfsfc**	**Elev**	**Ugrd10m**
Corr Coeff	0.754	0.208	0.181	0.179	0.156

Deployment in Lima: Despite the lack of ground labels for deploying the LDF-based NNW model in Lima, Peru, it is important to address the pressing issue: Lima is the second most polluted city in the Americas [32] and suffers from a scarcity of sensors [36] (Peru is a developing country). Our model provides a groundbreaking outcome in $PM_{2.5}$ estimation for Lima and serves as a vital first step toward implementing similar models in other *data-poor* regions.

6.1 Limitations and Future Work

While our methodology improves $PM_{2.5}$ estimation, further exploration, and alternate improvements are still needed, which we outline below.

Experiments with Alternate Datasets. Previous experiments with the US and Lima data are comprehensive but do not include datasets lacking spatial and semantic dependencies [5]. This was done primarily to ensure accurate and comprehensive data for modeling and estimation. Future plans include expanding our study to incorporate such datasets.

Capturing Temporal Trends. The LDF feature captures spatial and semantic dependencies but lacks focus on temporal trends in the data due to missing temporal points. In the future, we aim to extend this technique to time-series data, aiming for prediction rather than forecasting [39].

Extending to Alternate Domains. While our focus lies in $PM_{2.5}$ estimation, testing the LDF on alternate domains like wildfire estimation and weather forecasting is useful due to the presence of similar spatial patterns. Future studies should explore these applications and develop new LDF features accordingly.

7 Conclusion

This paper addresses the problem of *spatial* transfer learning for estimating $PM_{2.5}$ levels, emphasizing transfer between regions with low autocorrelation and predicting at unseen test locations. We aim to improve *instance transfer learning* (ITL) models, which often overlook spatial and semantic dependencies in the data. We introduce the *Latent Dependency Factor* (LDF) to capture these dependencies, integrating it as a new feature in both source and target datasets. Our experiments on US and Peru datasets demonstrate LDF's effectiveness in improving PM2.5 estimation. Furthermore, qualitative analysis of these datasets confirms that the LDF captures larger PM2.5 patterns missed by regular transfer models. While more future work remains in this space, we believe our approach of achieving *spatial* transfer learning using *Latent Dependency Factor* is a promising and novel solution for this highly complex domain.

References

1. Ayers, G., Keywood, M., Gras, J.: TEOM vs. manual gravimetric methods for determination of PM2.5 aerosol mass concentrations. Atmos. Environ. **33**(22), 3717–3721 (1999)
2. Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.R., Samek, W.: On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. PLoS ONE **10**(7), e0130140 (2015)
3. Bi, J., Belle, J.H., Wang, Y., Lyapustin, A.I., Wildani, A., Liu, Y.: Impacts of snow and cloud covers on satellite-derived PM2. 5 levels. Remote Sens. Environ. **221**, 665–674 (2019)
4. Bi, J., Wildani, A., Chang, H.H., Liu, Y.: Incorporating low-cost sensor measurements into high-resolution PM2. 5 modeling at a large spatial scale. Environ. Sci. Technol. **54**(4), 2152–2162 (2020)
5. Chen, S.: Beijing PM2.5. UCI Machine Learning Repository (2017). https://doi.org/10.24432/C5JS49
6. Daumé III, H.: Frustratingly easy domain adaptation. ACL **2007**, 256 (2007)
7. Department of Energy and Environmental Protection: Deep forecasts unhealthy levels of PM2.5 wednesday for the entire state from Canadian wildfire smoke (2023)
8. Dey, S., Di Girolamo, L., van Donkelaar, A., Tripathi, S., Gupta, T., Mohan, M.: Variability of outdoor fine particulate (PM2.5) concentration in the Indian subcontinent: a remote sensing approach. Remote Sens. Environ. **127**, 153–161 (2012)
9. Dong, X., Yu, Z., Cao, W., Shi, Y., Ma, Q.: A survey on ensemble learning. Front. Comput. Sci. **14**, 241–258 (2020)
10. Duan, L., Xu, D., Tsang, I.: Learning with augmented features for heterogeneous domain adaptation. In: Proceedings of the 29th International Conference on Machine Learning (ICML), pp. 667–674 (2012)
11. El Haddad, I., Marchand, N., Wortham, H., et al.: Primary sources of PM 2.5 organic aerosol in an industrial Mediterranean city, Marseille. Atmos. Chem. Phys. **11**(5), 2039–2058 (2011)
12. Fong, I.H., Li, T., Fong, S., Wong, R.K., Tallon-Ballesteros, A.J.: Predicting concentration levels of air pollutants by transfer learning and recurrent neural network. Knowl.-Based Syst. **192**, 105622 (2020)

13. Garcke, J., Vanck, T.: Importance weighted inductive transfer learning for regression. In: Calders, T., Esposito, F., Hüllermeier, E., Meo, R. (eds.) ECML PKDD 2014. LNCS (LNAI), vol. 8724, pp. 466–481. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-44848-9_30
14. Gupta, S., Bi, J., Liu, Y., Wildani, A.: Boosting for regression transfer via importance sampling. Int. J. Data Sci. Anal. (2023)
15. Hall, M.A.: Correlation-based feature selection for machine learning. Ph.D. thesis, The University of Waikato (1999)
16. Huang, J., Gretton, A., Borgwardt, K., Schölkopf, B., Smola, A.: Correcting sample selection bias by unlabeled data. In: Advances in Neural Information Processing Systems, vol. 19 (2006)
17. Ito, K., Xue, N., Thurston, G.: Spatial variation of PM2.5 chemical species and source-apportioned mass concentrations in New York City. Atmos. Environ. **38**(31), 5269–5282 (2004)
18. Jaipuria, N., et al.: Deflating dataset bias using synthetic data augmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pp. 772–773 (2020)
19. Kingma, D.P., Dhariwal, P.: Glow: generative flow with invertible 1 × 1 convolutions. In: Advances in Neural Information Processing Systems, vol. 31 (2018)
20. Kinney, P.L., Aggarwal, M., Northridge, M.E., Janssen, N.A., Shepard, P.: Airborne concentrations of PM (2.5) and diesel exhaust particles on Harlem sidewalks: a community-based pilot study. Environ. Health Perspect. **108**(3), 213–218 (2000)
21. Kumar, A., Naughton, J., Patel, J.M., Zhu, X.: To join or not to join? Thinking twice about joins before feature selection. In: Proceedings of the 2016 International Conference on Management of Data, pp. 19–34 (2016)
22. Li, M., Chen, X., Li, X., Ma, B., Vitányi, P.M.: The similarity metric. IEEE Trans. Inf. Theory **50**(12), 3250–3264 (2004)
23. Liu, J., Chai, C., Luo, Y., Lou, Y., Feng, J., Tang, N.: Feature augmentation with reinforcement learning. In: 2022 IEEE 38th International Conference on Data Engineering (ICDE), pp. 3360–3372. IEEE (2022)
24. Ma, J., Li, Z., Cheng, J.C., Ding, Y., Lin, C., Xu, Z.: Air quality prediction at new stations using spatially transferred bi-directional long short-term memory network. Sci. Total Environ. **705**, 135771 (2020)
25. Pan, S.J., Shen, D., Yang, Q., Kwok, J.T.: Transferring localization models across space. In: AAAI, pp. 1383–1388 (2008)
26. Pan, S.J., Yang, Q.: A survey on transfer learning. IEEE Trans. Knowl. Data Eng. **22**(10), 1345–1359 (2009)
27. Park, Y., Kwon, B., Heo, J., Hu, X., Liu, Y., Moon, T.: Estimating PM2. 5 concentration of the conterminous united states via interpretable convolutional neural networks. Environ. Pollut. **256**, 113395 (2020)
28. Rosenstein, M.T., Marx, Z., Kaelbling, L.P., Dietterich, T.G.: To transfer or not to transfer. In: NIPS 2005 Workshop on Transfer Learning, vol. 898 (2005)
29. Sato, M., Hansen, J.E., McCormick, M.P., Pollack, J.B.: Stratospheric aerosol optical depths, 1850–1990. J. Geophys. Res. Atmos. **98**(D12), 22987–22994 (1993)
30. Sharma, S., Chandra, M., Kota, S.H.: Health effects associated with PM 2.5: a systematic review. Curr. Pollut. Rep. **6**, 345–367 (2020)
31. Sugiyama, M., Nakajima, S., Kashima, H., Buenau, P., Kawanabe, M.: Direct importance estimation with model selection and its application to covariate shift adaptation. In: Advances in Neural Information Processing Systems, vol. 20 (2007)

32. Tapia, V., Steenland, K., Vu, B., Liu, Y., Vásquez, V., Gonzales, G.F.: PM2.5 exposure on daily cardio-respiratory mortality in Lima, Peru, from 2010 to 2016. Environ. Health **19**, 1–7 (2020)
33. Vasiliev, I.R.: Visualization of spatial dependence: an elementary view of spatial autocorrelation. In: Practical Handbook of Spatial Statistics, pp. 17–30. CRC Press (2020)
34. Veyseh, A.P.B., Van Nguyen, M., Min, B., Nguyen, T.H.: Augmenting open-domain event detection with synthetic data from GPT-2. In: Oliver, N., Pérez-Cruz, F., Kramer, S., Read, J., Lozano, J.A. (eds.) Machine Learning and Knowledge Discovery in Databases. Research Track: European Conference, ECML PKDD 2021, Bilbao, Spain, 13–17 September 2021, Proceedings, Part III 21, pp. 644–660. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-86523-8_39
35. Vobeckỳ, A., Hurych, D., Uřičář, M., Pérez, P., Sivic, J.: Artificial dummies for urban dataset augmentation. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, pp. 2692–2700 (2021)
36. Vu, B.N., et al.: Developing an advanced PM2.5 exposure model in Lima, Peru. Remote Sens. **11**(6), 641 (2019)
37. Yadav, K., Arora, V., Kumar, M., Tripathi, S.N., Motghare, V.M., Rajput, K.A.: Few-shot calibration of low-cost air pollution ($PM_{2.5}$) sensors using meta learning. IEEE Sens. Lett. **6**(5), 1–4 (2022)
38. Yao, B., Ling, G., Liu, F., Ge, M.F.: Multi-source variational mode transfer learning for enhanced PM2.5 concentration forecasting at data-limited monitoring stations. Expert Syst. Appl. **238**, 121714 (2024)
39. Zheng, V.W., Xiang, E.W., Yang, Q., Shen, D.: Transferring localization models over time. In: AAAI, vol. 2008, pp. 1421–1426 (2008)
40. Zhang, T., He, W., Zheng, H., Cui, Y., Song, H., Fu, S.: Satellite-based ground PM2. 5 estimation using a gradient boosting decision tree. Chemosphere **268**, 128801 (2021)
41. Hu, X., et al.: Estimating $PM_{2.5}$ concentrations in the conterminous United States using the random forest approach. Environ. Sci. Technol. **51**(12), 6936–6944 (2017)
42. Loog, M.: Nearest neighbor-based importance weighting. In: 2012 IEEE International Workshop on Machine Learning for Signal Processing, pp. 1–6 (2012)

Leveraging Foundation Models for Multi-modal Federated Learning with Incomplete Modality

Liwei Che[1,2], Jiaqi Wang[2], Xinyue Liu[3], and Fenglong Ma[2](✉)

[1] Rutgers University, Piscataway, NJ 08854, USA
`lw.che@rutgers.edu`
[2] Pennsylvania State University, University Park, PA 16802, USA
{`jawang,fenglong`}`@psu.edu`
[3] Dalian University of Technology, Dalian, Liaoning 116621, China
`xyliu@dlut.edu.cn`

Abstract. Federated learning (FL) has obtained tremendous progress in providing collaborative training solutions for distributed data silos with privacy guarantees. However, few existing works explore a more realistic scenario where the clients hold multiple data modalities. In this paper, we aim to solve a novel challenge in multi-modal federated learning (MFL) – modality missing – the clients may lose part of the modalities in their local data sets. To tackle the problems, we propose a novel multi-modal federated learning method, **Fed**erated **M**ulti-modal contrasti**V**e training with **P**re-trained completion (FedMVP), which integrates the large-scale pre-trained models to enhance the federated training. In the proposed FedMVP framework, each client deploys a large-scale pre-trained model with frozen parameters for modality completion and representation knowledge transfer, enabling efficient and robust local training. On the server side, we utilize generated data to uniformly measure the representation similarity among the uploaded client models and construct a graph perspective to aggregate them according to their importance in the system. We demonstrate that the model achieves superior performance over two real-world image-text classification datasets and is robust to the performance degradation caused by missing modality.

Keywords: Federated Learning · Multi-modal Learning

1 Introduction

Federated learning (FL) has emerged as a promising paradigm for training machine learning models on decentralized data [1,23,35–38]. In many realistic scenarios, the multi-modal data are collected among distributed data silos and

---

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-70378-2_25.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
A. Bifet et al. (Eds.): ECML PKDD 2024, LNAI 14949, pp. 401–417, 2024.
https://doi.org/10.1007/978-3-031-70378-2_25

stored in a privacy-sensitive manner, such as the examination and diagnostic records of patients in different hospitals and the multimedia data generated on mobile devices. However, most existing federated learning works focus on single modality scenarios (*e.g.*, image or text) with limited capacity for data with heterogeneous formats and properties. Regarding the fast development of multimedia technology and distributed systems, developing a robust and efficient FL framework for multi-modal machine learning tasks is significant.

To date, several early attempts for multimodal federated learning (MFL) [2] have been proposed [3,6,18,42,44–46,48]. Some of these approaches [3,44,45] consider scenarios where the federated system contains both uni-modal and multi-modal clients. However, most of these works assume that all modalities are available to all clients, which is a strong assumption that may not hold in real-world situations. For example, content posted on social media often combines images and text, but users may also publish posts containing only images or text. This modality missing problem poses a substantial challenge as it can severely impact the model's learning ability and performance.

In this paper, we aim to address this general and realistic problem of **modality missing**, where clients share the same modality combinations, but some multi-modal instances lack part of the modality data. For example, a client holds 1000 image-text pairs, while 200 of them only have image data, and 300 instances have only text data. A few existing works [20,22] focus on the modality incompleteness problem. However, they either only consider text missing in the vision-language learning task or deal with sensor signals that are similar in format. We believe that an advanced MFL framework should be robust to modality incomplete training data and maintain satisfactory performance.

To resolve those challenges, we proposed a multi-modal federated learning framework, namely **Fed**erated **M**ulti-modal contrasti**V**e training with **P**re-trained completion (FedMVP), which uses frozen pre-trained models as the teachers to support the learnable multi-modal joint encoder module for efficient multi-modal representation learning and to generate informative synthetic data. To enhance the model resilience to the performance degradation caused by modality missing, we utilize the cross-modal generation ability of the recently proposed pre-trained models [14,15,27] to complete the missing modalities. To further improve the representation learning performance, we proposed an efficient knowledge-transferring method to transfer the representation knowledge from the pre-trained large models to our multi-modal joint learning module. This knowledge-transferring method can alleviate the conflict between the massive data and computing costs requirements for training and fine-tuning of pre-trained large models and the limited resources of federated learning clients. The proposed framework is competent in integrating various pre-trained models with affordable communication costs. As shown in Table 1, compared to the most costly baseline FedViLT, the FedMVP reduces the communication cost by **26.7×** and computation FLOPS by **15.5×**. The pre-trained foundation models will play as the frozen data encoders to transform the original data into high-quality representations, which play an important role in the contrastive-manner training process for the multi-modal joint encoder module.

Table 1. Comparison between FedMVP and baselines in terms of #FLOPS (Floating Point Operations Per Second) and #transmitted parameters per round.

Method	#FLOPS	#Parameters
FedViT [8]	22.6G	86.4M
FedBERT [7]	38.1G	110.1M
FedCLIP [27]	60.7G	197.2M
FedViLT [21]	55.9G	298.6M
MMFed [42]	1.4G	4.49M
FedMVP	3.6G	11.2M

We summarize our **contributions** as follows: (1) We proposed a novel MFL framework that integrates pre-trained large-scale models to conduct efficient multi-modal representation learning and is robust to the modality missing challenge. Our proposed method shows superior performance on two multi-modal classification benchmarks under both IID and non-IID settings. (2) To efficiently transfer the learnable representation knowledge from the pre-trained model to the multi-modal joint module under the resource-limited scenario, we proposed a Multi-modal Contrastive Matching (MCM) loss and a Representation Aligned Margin (RAM) loss, which effectively improve the model performance with severe modality missing up to 80%. (3) Instead of aggregating the models based on the data distribution or the model architecture, we propose a novel aggregation algorithm for the MFL server aggregation based on the representation abilities among the client models.

2 Related Work

Multi-modal Federated Learning (MFL). MFL is still in its early stages of development. Some of the most existing works [18,42,48] focus on exploring task-specific approaches with complete modalities. In [42], the authors propose a multi-modal federated learning framework for multi-modal activity recognition with a local co-attention module to fuse multi-modal features. [5] gives a detailed analysis of the convergence problem of MFL with late fusion methods under the Non-IID setting. [3,44,45] adapt modality-wise encoders to tackle the MFL system with both uni-modal and multi-modal clients. However, few of them explore the scenario where multi-modal data are incomplete, which may cause significant performance degradation.

Modality Missing in Multi-modal Learning. As a widely existing challenge in the realistic scenario, handling modality missing has drawn the attention of the multi-modal learning community. Some early works [25,30,39] build their methods based on conditional VAE to capture the multi-modal distribution for the cross-modal generation. [33] as one of the recent works utilizes cross-modal fusion to improve the model robustness for modality missing in testing. [29] proposes a contrastive framework for learning both paired and unpaired data. In [22], the authors leverage Bayesian meta-learning to reconstruct pseudo text input from image input to resolve the missing modality issue. Instead of training a generative model from scratch, we utilize the large-scaled pre-trained model [14,15] and prompt augmentation to achieve effective cross-modal generation for completing the missing data pairs.

Vision-language Pre-training. Represented by CLIP [27] and ALIGN [12], the large-scale Vision and Language Pre-training (VLP) models have demonstrated their surprising performance in many downstream vision-language learning tasks [10] and strong adaptability to new scenarios. A few works have taken the first steps towards incorporating federated learning with pre-training techniques. In [32], the authors propose a splitting learning-based framework for training large-scale models like BERT in federated learning systems. PromptFL [9] allows the clients to train shared soft prompts collaboratively using CLIP [27] to provide strong adaptation capability to distributed users tasks. [4,19,40,41] are trying to explore the efficient methods for lightweight and fast adaptation of pre-trained models. [31] proposes FedPCL to transfer shared knowledge among the clients based on prototype contrastive learning. In this work, instead of fine-tuning the large-scale pre-trained models or splitting the model into multiple modules, we conduct effective knowledge transferring to enhance the representation learning performance of a lightweight local module.

Multi-modal Contrastive Learning. Contrastive learning is widely used in the self-supervised learning field, where the learned representations will be assigned to positive and negative samples based on the class belongings. As for its application in multi-modal learning [16,17,47], instead of using spatial or temporal transforming to a single instance, the positive pairs are defined as the samples with the same ID or time window. In [47], the authors propose CrossCLR to improve the quality of learned joint embedding from multi-modal data with a novel contrastive loss, which utilizes both inter-modality and intra-modality alignment. [26] extends the multi-modal contrastive learning to efficiently align the cross-modal representations. Inspired by the predecessors, we adopt a multi-modal contrastive loss to improve the quality of the learned multi-modal joint representations based on the modality-specific representation encoded by the frozen pre-trained models.

3 Methodology

To explore multi-modal data in federated systems, we propose **FedMVP** for MFL with the robustness of modality missing during training. As illustrated in Fig. 1, the proposed FedMVP contains four main modules for effective MFL,

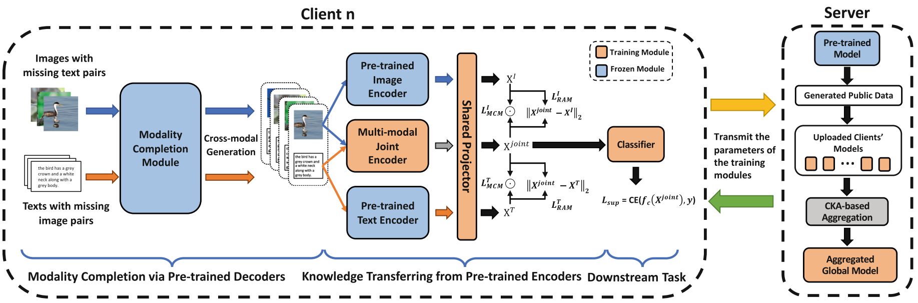


Fig. 1. The overview of the proposed FedMVP framework.

including *Modality Completion Module*, *Multi-modal Joint Learning Module*, *Knowledge Transferring via Contrastive Training*, *CKA-based Aggregation*.

3.1 Problem Formulation

Multi-modal Federated Learning. In an MFL system, there exist N clients aiming to collaboratively train a global model w_G for multi-modal representation learning. For client n, its local data set $\mathcal{D}_n = \{(X_i, y_i)\}_{i=1}^{|\mathcal{D}_n|}$ contains $|\mathcal{D}_n|$ image-text pairs denoted as $X_i = \{x_i^I, x_i^T\}$, i.e., the i-th image data x_i^I and text data x_i^T. y_i is the corresponding label. A data instance is denoted as $X_i = \{x_i^I\}$ or $X_i = \{x_i^T\}$ if modality missing happens. Each local model w_n performs on the local task $F_n(\cdot; w_n) : \mathbb{R}^n \rightarrow \mathbb{R}^d$ and collaborates with other clients for the global task $F_G(\cdot; w_G) : \mathbb{R}^{d_G} \rightarrow \mathbb{R}^d$. Formally, the global objective of MFL for the image-text classification problem is defined as

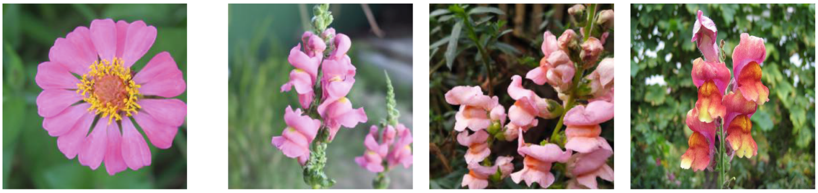
$$\min\ L_G(F_G(\cdot; w_G)) = \min \sum_{n=1}^{N} \gamma_n L_n(F_n(\mathcal{D}_n; w_n)) \tag{1}$$

where γ_n is the aggregation weights, and L_n is the local loss function.

3.2 Local Data Preprocessing

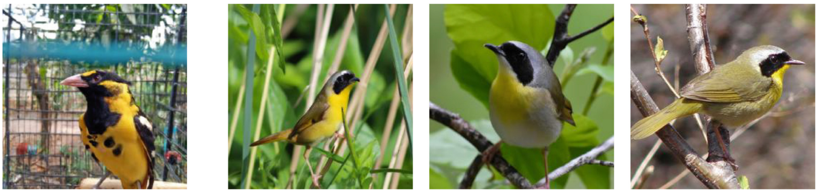
A pre-trained foundation model is deployed on both the server side and client side, which consists of an image encoder $f_E^I(\cdot)$ and a text encoder $f_E^T(\cdot)$ for representation extraction, an image decoder $f_D^I(\cdot)$ and a text decoder $f_D^T(\cdot)$ for the cross-modal generation. Notably, all the parameters of the pre-trained models are frozen and will not be transmitted between the server and clients. We will explain the pre-trained model we used below, as well as the details of the local training process.

Modality Completion Module. To solve the performance drop problem caused by modality missing, the modality completion module utilizes the cross-modal generation ability of the pre-trained model to complete the missing



(a) original text (b) label only (c) prompt (d) original image

Fig. 2. Examples of the generated "snapdragon" images.



(a) original text (b) label only (c) prompt (d) original image

Fig. 3. Examples of the generated "yellowthroat" images.

part of multi-modal data. We use DALLE2 [28] for text-to-image generation, and BLIP2 [14] for image-to-text generation. Inspired by [27], we use designed prompts to improve the generation quality of the modality completion module.

Prompt Augmented Text-to-Image Generation. Given an image-text pair X_i with only text data x_i^T, the modality completion module could generate an image from a text prompt. To avoid the semantic ambiguities caused by synonyms and polysemy in the text data and label name. Instead of directly using text data x_i^T as the input, we adopt a coarse-to-fine prompt to augment the generation. The prompt template is *"A photo of {fine-grained label}, a kind of {class label}, {text description}"*, which helps the pre-trained models to better understand the characteristics of the generation target and improve the semantic correlation between the text prompt and generated image. Figures 2 and 3 show examples with different inputs to generate the classes "snapdragon" and "yellow throat" on the Oxford Flower and CUB-200 datasets, where our designed prompt gives high-quality fake images that are close to the original ones.

Accordingly, we obtain the image generation prompt based on the original text data, and the process is denoted as $p^T(x_i^T)$. The augmented prompt $p^T(x_i^T)$ will firstly be decomposed by text encoder $f_E^T(\cdot)$, then passed to image decoder f_D^I for generating the synthetic image $\hat{x}_i^I$, i.e.$\hat{x}_i^I = f_D^I(f_E^T(p^T(x_i^T)))$.

Prompt Augmented Image-to-text Generation. For the image-to-text generation, considering the original text data contains detailed descriptions of the image pair, the direct image captioning result may not be able to cover the fine-grained text details. Therefore, we adapt both the visual question answering (VQA) and image captioning functions of the pre-trained model to generate text pairs $\hat{x}_i^T$

for the image input. Specifically, with a given image input x_i^I, the modality completion module first performs the VQA task over three serial question prompts to get fine-grained descriptions of the image. For instance, given prompt input *"What is the color of the petals?"* for a flower image with the pink pedal, the response answer could be *"Pink"*. After obtaining the answers to the three question prompts, we combine them with the image captioning outcome as the final synthetic text, e.g., *"A photo of {flower}, with {pink} petals and {white} pistils,{there is a pink flower with a yellow center in the middle of the picture}"*. We show examples of image-to-text generation in Table 2.

To better understand the model design and avoid notation confusion, we use completed image-text pair $X_i = \{x_i^I, x_i^T\}$ in the following sections to illustrate how data is processed in FedMVP.

Table 2. Image-to-text completion examples from CUB-200 and Oxford Flower.

Type	Text	Original Image
Original Text	this flower is yellow in color, and has petals that are layered.	
Synthetic Text	A flower with yellow petals and yellow pistil. yellow flower with water droplets on it in a garden	
Original Text	this bird is yellow and black in color, and has a stubby black beak.	
Synthetic Text	A bird with black wings and yellow belly. yellow and black bird perched on a cattails plant in a marsh	

Modality-specific Representations. The foundation models are believed to have extraordinary representation extraction ability since they are trained with millions of data instances. Thus, we obtain the image-specific embedding and text-specific embedding via the pre-trained encoders. Specifically, we use the pre-trained Vision Transformer(ViT) [8] with the patch size of 16×16 as the image-specific encoder to generate high-quality embedding from image input. The image-specific embedding $\mathbf{X}^I$ is encoded via the pre-trained image encoder $f_E^I(\cdot)$ and then mapping to the multi-modal latent space via a shared projection head $f_{shared}(\cdot)$, i.e., $\mathbf{X}^I = f_{shared}(f_E^I(\mathbf{x}^I)) \in \mathbb{R}^{d_{latent}}$. Similarly, we get the text-specific embedding $\mathbf{X}^T$ from the pre-trained BERT model [7], where $\mathbf{X}^T = f_{shared}(f_E^T(\mathbf{x}^T)) \in \mathbb{R}^{d_{latent}}$.

3.3 Local Training

Multi-modal Joint Learning Module. The multi-modal joint learning module contains a joint encoder $f_E^{Joint}(\cdot)$ designed to efficiently fuse the image-text information into a complete view. It consists of a cross-modal fusion layer and follows attention-based embedding layers.

Pre-processing. Given an image-text pair $\{\mathbf{x}^I, \mathbf{x}^T\}$ as input, we use a non-overlapped patch embedding layer and the pre-trained text encoder $f_E^T(\cdot)$ to get the patch sequence $\mathbf{I}_{com}$ and text embedding $\mathbf{T}_{com}$, both belongs to the common dimension d_{com}.

Cross-Modal Fusion. After the positional embedding operation, both the image and text embeddings are fed into the cross-modal fusion layer, which contains a vision-to-language attention module and a language-to-vision attention module. Both modules are based on the cross-modal attention [33], which can effectively fuse the representation between the two input modality embeddings. We take the image-to-text embedding $\mathbf{X}^{I\rightarrow T}$ to show the cross-modal attention:

$$\mathbf{X}^{I\rightarrow T} = CM_{I\rightarrow T}(\mathbf{I}_{com}, \mathbf{T}_{com}) = softmax(\frac{W_{Q_I}\mathbf{I}_{com}W_{K_T}^{\mathsf{T}}\mathbf{T}_{com}^{\mathsf{T}}}{\sqrt{d_{com}}})W_{V_T}. \tag{2}$$

Similarly, we can get text-to-image embedding $\mathbf{X}^{T\rightarrow I}$. The obtained $\mathbf{X}^{I\rightarrow T}$ and $\mathbf{X}^{T\rightarrow I}$ will be concatenated together and projected to the latent space as the final joint embedding via the shared projection head $f_{shared}(\cdot)$ and a self-attention layer as follows:

$$\mathbf{X}^{joint} = f_{shared}(SelfAttention(\mathbf{X}^{I\rightarrow T} \oplus \mathbf{X}^{T\rightarrow I})). \tag{3}$$

We now obtain the image-specific embedding $\mathbf{X}^I$, text-specific embedding $\mathbf{X}^T$, and joint embedding $\mathbf{X}^{joint}$ in the same latent space $\mathbb{R}^{d_{latent}}$.

Knowledge Transferring from Pre-trained Model. The training data of large-scale models in the pre-training stage is neither available nor affordable for distributed silos to process, making the fine-tuning and traditional knowledge distillation [11] of large-scale models impractical under the MFL scenario. In order to transfer the rich representation knowledge from the pre-trained model, we propose *Multi-modal Contrastive Matching (MCM) Loss* and *Representation Aligned Marginal (RAM) Loss* to improve the representation learning performance of the joint encoding module.

Multi-modal Contrastive Matching Loss. To obtain a high-quality joint representation, we utilize the idea of contrastive learning to closer the joint embedding with its corresponding modality-specific embedding and distance it from the embedding of the other categories in the latent space. Let $s_c(x_i, x_j)$ represent the cosine similarity between two embedding, x_i and x_j, and $\tau \in (0, 1]$ be the temperature hyperparameter. The corresponding scaled similarity is defined as: $sim(x_i, x_j) = \exp(\frac{s_c(x_i,x_j)}{\tau})$.

Given a batch of embedding $\mathcal{B} = \{\mathbf{X}_i^T, \mathbf{X}_i^I, \mathbf{X}_i^{joint}\}_{i=1}^{|\mathcal{B}|}$, the positive pair for the contrastive learning is defined as the joint embedding with its corresponding modality-specific embedding, i.e., $(\mathbf{X}_i^T, \mathbf{X}_i^{joint})$ and $(\mathbf{X}_i^I, \mathbf{X}_i^{joint})$. The other ways of pairing will be treated as negative pairs, denoted as:

$$\Omega_i^m = \sum_{i \neq j} (sim(\mathbf{X}_i^M, \mathbf{X}_j^M) + sim(\mathbf{X}_i^M, \mathbf{X}_j^{joint}) + sim(\mathbf{X}_i^{joint}, \mathbf{X}_j^{joint})), \tag{4}$$

where $M \in \{I, T\}$ indicates the modality type. We define the multi-modal contrastive matching (MCM) loss of all data embedding as follows:

$$L_{MCM}(\mathcal{B}) = -\frac{1}{|\mathcal{B}|} \sum_{i=1}^{|\mathcal{B}|} \log \left(\frac{sim(\mathbf{X}_i^T, \mathbf{X}_i^{joint})}{\Omega_i^T} + \frac{sim(\mathbf{X}_i^I, \mathbf{X}_i^{joint})}{\Omega_i^I} \right). \tag{5}$$

Representation Aligned Margin Loss. We propose the *Representation Aligned Margin* (RAM) loss to further enrich the joint representation via pre-trained knowledge to close the semantic gap between the joint embedding and the modality-specific embeddings. We use the classification loss derived from the embeddings to evaluate its representation quality. For the i-th data sample, the supervised classification loss of one of its corresponding embeddings is denoted as $L_{sup}^M(i) = CE(f_c(\mathbf{X}_i^M), y_i)$.

Intuitively, embeddings with lower cross-entropy losses contain more informative features from the raw data. With an embedding batch $\mathcal{B}$, the RAM loss aligns joint embedding with image and text embedding separately, if the modality-specific embedding has better representation. Thus, the RAM loss is defined as:

$$L_{RAM}(\mathcal{B}) = \frac{1}{|\mathcal{B}|} \sum_{i=1}^{|\mathcal{B}|} \left(L_{RAM}^I(i) + L_{RAM}^T(i) \right), \tag{6}$$

$$L_{RAM}^M(I) = \begin{cases} \|\mathbf{X}_i^{joint} - \mathbf{X}_i^M\|_2, & \text{if } L_{sup}^M(i) < L_{sup}^{joint}(i) \\ 0, & \text{otherwise} \end{cases}, \tag{7}$$

where $\mathbf{X}_i^M$ and $\mathbf{X}_i^{joint}$ are all derived from the i-th sample in the batch, and $|\mathcal{B}|$ is the batch size. The L2 norm is denoted by $\|\cdot\|_2$.

Classification Loss. A two-layer linear classifier $f_C(\cdot)$ will serve as the classifier using only joint embedding as input. The supervised classification loss L_{sup} of client n can be obtained:

$$L_{sup}(\mathcal{B}) = \frac{1}{|\mathcal{B}|} \sum_{i=1}^{|\mathcal{B}|} CE\left(f_C\left(\mathbf{X}_i^{joint}; \boldsymbol{\omega}_n \right), y_i \right), \tag{8}$$

where $f_C(\cdot)$ denotes the classifier model of client n, $CE(\cdot)$ is the cross-entropy loss function, and y_i is the corresponding label of i-th joint embedding $\mathbf{X}_i^{joint}$.

Total Loss. The final local training loss of client k in FedMVP is:

$$L_{local}(\mathcal{D}_k) = L_{sup}(\mathcal{D}_k) + L_{MCM}(\mathcal{D}_k) + L_{RAM}(\mathcal{D}_k), \tag{9}$$

At each communication round, each client will upload the parameters of the multi-modal joint learning module and classifier to the server for further global aggregation.

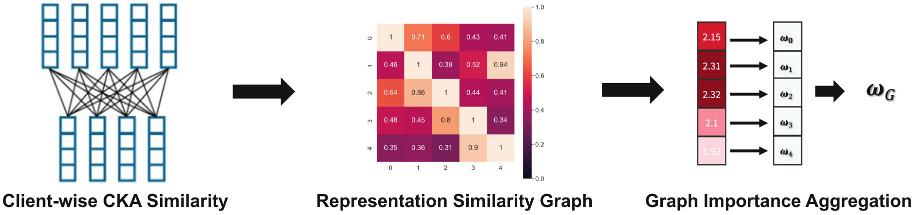


Fig. 4. CKA-based Server Aggregation

3.4 Server Aggregation

Previous works tend to aggregate based on the modality type held by the clients [3,43], share public dataset [44], or model structure [45], which may lead to data privacy leakage and lacking uniformity. To better enhance the representational ability of the global model, we propose a server aggregation method based on the similarity of model output representations.

At the beginning of the aggregation phase, the server-side pre-trained model will automatically generate m synthetic data pairs X_m, where the data amount m is equal to the number of classes of the dataset. Given an uploaded client model, its output representations with generated data are defined as:

$$\mathbf{X}_\omega = [F_\omega(X_1), \ldots, F_\omega(X_m)]^T \in \mathbb{R}^{m \times d_{out}}. \tag{10}$$

To measure the similarity of the model representations among the clients, we utilize the centered kernel alignment (CKA) metric [13] based on the output representations from upload models, which is defined as follows:

$$s_{ij}(\omega_i, \omega_j) = \frac{Cov(\mathbf{X}_{\omega_i}, \mathbf{X}_{\omega_j})}{\sqrt{Cov(\mathbf{X}_{\omega_i}, \mathbf{X}_{\omega_i}) Cov(\mathbf{X}_{\omega_j}, \mathbf{X}_{\omega_j})}}, \tag{11}$$

where $Cov(X, Y) = (m-1)^2 tr(XX^T H_m YY^T H_m)$, H_m is the centering matrix, $tr(\cdot)$ denotes the matrix trace, m represents the number of input represents.

With the calculated representation similarity scores, the server constructs a representation similarity graph to illustrate the relationship among clients, as shown in Fig. 4. The importance of each client in the representation similarity graph is determined by the sum of its similarity score with all the other clients.

$$\gamma_i^t = softmax([s_1, \ldots, s_i, \ldots, s_K]), \tag{12}$$

where K is the number of clients who participate in the t-th aggregation, $s_i = \sum_{j=1}^{K-1} s_{ij}$ is the collection of the graph importance of all K clients. Finally, the global model is weighted and aggregated based on the clients' graph importance γ_i^t as follows:

$$w_G^t = \sum_{i=1}^{K} \gamma_i^t w_i^t. \tag{13}$$

4 Experiments

4.1 Experiment Setting

Datasets. We evaluate the proposed FedMVP on two multi-modal fine-grained categorization datasets, *The Caltech-UCSD Birds-200-2011 (CUB-200) dataset* [34] and *Oxford Flower* [24]. Both contain paired image-text data, and each image has 10 related descriptive text. CUB-200 has 200 bird classes with 10610 training image-text instances and 1178 for testing. Oxford Flower has 102 flower classes, a training size of 7370, and a testing size of 819.

Table 3. Evaluating the impact of incomplete modality on CUB-200 and Oxford Flower datasets under IID setting. β indicates the missing ratio of the training set.

Methods	CUB-200			Oxford Flower		
	$\beta = 0.3$	$\beta = 0.5$	$\beta = 0.8$	$\beta = 0.3$	$\beta = 0.5$	$\beta = 0.8$
FedViT	74.71%	67.12%	60.33%	92.15%	84.52%	76.64%
FedBERT	66.76%	58.98%	52.54%	74.23%	70.72%	67.81%
FedCLIP	75.73%	69.68%	63.41%	91.12%	86.32%	78.55%
FedViLT	76.29%	70.28%	64.11%	92.67%	88.31%	81.52%
MMFed	63.15%	57.48%	51.60%	72.91%	69.43%	64.05%
FedMVP(Ours)	**77.89%**	**74.46%**	**70.31%**	**93.19%**	**91.28%**	**89.32%**

Data Distribution Setting. For **Independent Identically Distribution (IID)** setting, we equally distribute the training data to 10 clients with random selection. Each client will hold the same quantity of local data with a balanced category distribution. To simulate the **non-IID** scenario in federated systems, we divide the training data set into C shards according to the data set categories, i.e., 200 shards for CUB-200-2011 dataset and 102 shards for the Oxford Flower dataset. With fixed 10 clients, the data shards are randomly and equally distributed to clients.

Modality Missing Setting. We set $\beta \in [0, 1]$ as the missing ratio. For example, given a constant $\beta = 0.3$, 30% randomly selected image-text pairs will lose either image or text data in equal chances. We select $\beta = 0.3, 0.5, 0.8$ to conduct our experiments, and the number of missing images and texts is the same.

Training Setting. With fixed 10 clients, the total communication round is 200. In each communication round, the clients will perform 10 epochs for local training with their own local datasets and the server will randomly select 70% of clients for aggregation. We choose AdamW as the optimization function with a scheduler-controlled learning rate $2e-5$. We adopt the warm-up scheduler and cosine annealing scheduler for the training process as well.

Baselines. Since the existing approaches for addressing modality missing in multi-modal federated learning are relatively limited, we choose **FedViT**, **FedBERT** as the uni-modal baseline and **FedCLIP**, **FedViLT**, **MMFed** as the multi-modal baseline. FedViT [8], FedBERT [7], FedCLIP [27] and FedViLT [21] are using large-scale foundation models pre-trained with millions of data as the local models. These large models are fine-tuned on the local data and upload all the parameters to the server for aggregation. MMFed [42] is a federated multi-modal learning method without leveraging foundation models. FedViLT [21] is designed specifically for modality missing. *Please refer to Appendix for details of the implementation.*

Table 4. Evaluating the impact of incomplete modality on CUB-200 and Oxford Flower datasets under the non-IID setting. β indicates the missing ratio of the training set.

Methods	CUB-200			Oxford Flower		
	$\beta = 0.3$	$\beta = 0.5$	$\beta = 0.8$	$\beta = 0.3$	$\beta = 0.5$	$\beta = 0.8$
FedViT	67.05%	61.17%	50.39%	86.25%	78.30%	70.03%
FedBERT	59.31%	51.14%	43.67%	68.43%	62.01%	57.16%
FedCLIP	67.63%	61.72%	56.78%	85.01%	80.13%	72.91%
FedViLT	69.19%	65.26%	58.34%	86.96%	81.63%	73.32%
MMFed	57.55%	51.12%	42.14%	65.90%	59.26%	52.79%
FedMVP(Ours)	**72.62%**	**69.73%**	**66.44%**	**88.54%**	**84.78%**	**82.47%**

Table 5. Evaluating the robustness of the methods over different test sets. *image only* and *text only* indicate the test set only contains either image or text. All the methods are trained over train set WITHOUT modality missing.

Methods	CUB-200			Oxford Flower		
	image only	text only	complete	image only	text only	complete
FedCLIP	56.47%	47.30%	79.73%	64.11%	53.59%	94.12%
FedViLT	64.55%	52.08%	**82.29%**	76.71%	60.91%	**96.67%**
MMFed	7.94%	13.07%	65.28%	26.37%	40.90%	74.89%
FedMVP(Ours)	**70.39%**	**64.44%**	80.79%	**80.82%**	**73.50%**	94.27%

4.2 Empirical Results

Results of the IID Setting. Table 3 shows the superior performance of FedMVP across different missing ratios under the IID setting on both CUB-200 and Oxford Flower datasets. Observably, all models exhibit a decline in accuracy with an increase in the missing ratio (β). FedMVP outperforms baseline methods consistently and demonstrates exceptional resilience to performance degradation due to missing modalities. For instance, on the CUB-200 dataset, FedMVP's accuracy margin over the next best-performing model, FedViLT, widens from about 1.6% at $\beta = 0.3$ to 6.2% at $\beta = 0.8$. A similar trend is observed on the Oxford Flower dataset, with the margin increasing from 0.52% to 7.8%. The rate of performance degradation of FedMVP is notably slower than the other models. Specifically, as β increases from 0.3 to 0.8, the accuracy of FedMVP drops by merely 7.58% and 3.87% on the CUB-200 and Oxford Flower datasets, respectively. In contrast, FedViT witnesses larger drops of 14.38% and 15.51%.

Results of the Non-IID Setting. The non-IID experimental results, presented in Table 4, all methods experience a significant decrease in accuracy compared to the IID setting, including FedMVP. The proposed FedMVP consistently outperforms the other methods across the settings. FedMVP has minimal performance degradation caused by non-IID compared to the baseline methods, with no more than 5% drop on CUB-200 and no more than 7% on Oxford Flower. Despite the increasing missing ratio from $\beta = 0.3$ to $\beta = 0.8$, FedMVP maintains a substantial lead in accuracy on both datasets. For instance, even with $\beta = 0.8$, FedMVP achieves an accuracy of 66.44% and 82.47% on the CUB-200 and Oxford Flower datasets, respectively, confirming its robustness to modality incompleteness under non-IID settings. Notably, the performance margin between FedMVP and baseline is further widened compared to the IID setting. For instance, on the Oxford Flower dataset, as $\beta = 0.8$, the accuracy of FedMVP is 29.68% higher than MMFed compared to 25.27% under IID.

Results of Single-modality Testing. Shown in Table 5, all methods experience significant performance drops when tested with only one modality (image or text). FedMVP shows the best resilience, achieving the highest accuracy in both image-only and text-only scenarios across datasets. FedViLT [21] performs best with complete data since it has 26.7× more parameters than FedMVP and is pre-trained over millions of pre-training data. It holds second place in single-modality tests. FedCLIP's performance is limited by local dataset size but benefits from separate ViT and BERT encodings. MMFed suffers the most due to its co-attention mechanism and performs better in text-only testing due to its integrated BERT. In summary, FedMVP demonstrates robustness in both training and testing under missing modalities.

Ablation Study. The results in Table 6 show that all the modules in the FedMVP model significantly contribute to its performance. Experimental results show that MCM loss and RAM loss can effectively improve the quality of the representation generated by the multi-modal joint encoder and enhance the final performance of the model by transferring pre-trained knowledge through representation learning. The modality completion module can supplement the data by providing additional training information using the transferable knowledge of the pre-trained model. Furthermore, the experimental results suggest that CKA similarity can effectively measure the importance of the representation learned by each client's local model and can improve aggregation performance compared to traditional average aggregation.

Table 6. Ablation study on both CUB-200 and Oxford Flower datasets with $\beta = 0.3$ under non-IID setting; wo/MCM denoting MCM Loss excluded; wo/RAM excludes RAM loss; wo/Completion refers to training without modality completion module; wo/CKA indicates server aggregation as FedAvg.

Model	CUB-200	Oxford Flower
FedMVP	72.62%	88.54%
-wo/MCM	66.87%	81.44%
-wo/RAM	68.25%	83.60%
-wo/Completion	67.49%	81.87%
-wo/CKA	70.11%	85.01%

5 Conclusion

In conclusion, we proposed the FedMVP framework to tackle modality missing, a widely existing real-world challenge, where part of the multi-modal data is incomplete and unaligned. Our framework utilizes large-scale pre-trained models with frozen parameters for modality completion and representation knowledge transfer at each client. It provides a solution for integrating large-scale pre-trained models to empower the federated system with robustness towards modality incompleteness. The experiments on the real-world image-text pair benchmark demonstrated the effectiveness of our proposed method. The proposed FedMVP framework shows great potential in addressing the missing modality and unified representation learning challenges of multi-modal federated learning. We hope this work can provide inspiration for future research in this field.

Acknowledgments. This study was partially supported by the National Science Foundation under Grant No. 2238275, 2333790, and 2348541.

References

1. Che, L., Long, Z., Wang, J., Wang, Y., Xiao, H., Ma, F.: Fedtrinet: a pseudo labeling method with three players for federated semi-supervised learning. In: 2021 IEEE Big Data, pp. 715–724 (2021)

2. Che, L., Wang, J., Zhou, Y., Ma, F.: Multimodal federated learning: A survey. Sensors **23**(15) (2023)
3. Chen, J., Zhang, A.: Fedmsplit: Correlation-adaptive federated multi-task learning across multimodal split networks. In: ACM SIGKDD, pp. 87-96 (2022)
4. Chen, J., Xu, W., Guo, S., Wang, J., Zhang, J., Wang, H.: Fedtune: a deep dive into efficient federated fine-tuning with pre-trained transformers (2022)
5. Chen, S., Li, B.: Towards optimal multi-modal federated learning on non-iid data with hierarchical gradient blending. In: IEEE INFOCOM 2022-IEEE Conference on Computer Communications, pp. 1469–1478. IEEE (2022)
6. Cobbinah, B.M., et al.: Reducing variations in multi-center alzheimer's disease classification with convolutional adversarial autoencoder. Med. Image Anal. **82**, 102585 (2022)
7. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv:1810.04805 (2018)
8. Dosovitskiy, A., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
9. Guo, T., Guo, S., Wang, J., Xu, W.: Promptfl: Let federated participants cooperatively learn prompts instead of models–federated learning in age of foundation model. arXiv preprint arXiv:2208.11625 (2022)
10. He, X., Peng, Y.: Fine-grained visual-textual representation learning. IEEE Trans. Circuits Syst. Video Technol. **30**(2), 520–531 (2019)
11. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network (2015)
12. Jia, C., et al.: Scaling up visual and vision-language representation learning with noisy text supervision. In: International Conference on Machine Learning, pp. 4904–4916. PMLR (2021)
13. Kornblith, S., Norouzi, M., Lee, H., Hinton, G.: Similarity of neural network representations revisited. In: International Conference on Machine Learning, pp. 3519–3529. PMLR (2019)
14. Li, J., Li, D., Savarese, S., Hoi, S.: Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. arXiv preprint arXiv:2301.12597 (2023)
15. Li, J., Li, D., Xiong, C., Hoi, S.: Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In: International Conference on Machine Learning, pp. 12888–12900. PMLR (2022)
16. Li, W., et al.: Unimo: towards unified-modal understanding and generation via cross-modal contrastive learning. arXiv preprint arXiv:2012.15409 (2020)
17. Liang, W., Zhang, Y., Kwon, Y., Yeung, S., Zou, J.: Mind the gap: Understanding the modality gap in multi-modal contrastive representation learning. arXiv preprint arXiv:2203.02053 (2022)
18. Liu, F., Wu, X., Ge, S., Fan, W., Zou, Y.: Federated learning for vision-and-language grounding problems. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 11572–11579 (2020)
19. Lu, W., Hu, X., Wang, J., Xie, X.: Fedclip: Fast generalization and personalization for clip in federated learning. arXiv preprint arXiv:2302.13485 (2023)
20. Ma, M., Ren, J., Zhao, L., Testuggine, D., Peng, X.: Are multimodal transformers robust to missing modality? In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 18177–18186 (June 2022)
21. Ma, M., Ren, J., Zhao, L., Testuggine, D., Peng, X.: Are multimodal transformers robust to missing modality? In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 18177–18186 (2022)

22. Ma, M., Ren, J., Zhao, L., Tulyakov, S., Wu, C., Peng, X.: Smil: multimodal learning with severely missing modality. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, pp. 2302–2310 (2021)
23. McMahan, B., Moore, E., Ramage, D., Hampson, S., y Arcas, B.A.: Communication-efficient learning of deep networks from decentralized data. In: Artificial Intelligence and Statistics, pp. 1273–1282. PMLR (2017)
24. Nilsback, M.E., Zisserman, A.: Automated flower classification over a large number of classes. In: Indian Conference on Computer Vision, Graphics and Image Processing (Dec 2008)
25. Pandey, G., Dukkipati, A.: Variational methods for conditional multimodal deep learning. In: 2017 International Joint Conference on Neural Networks (IJCNN), pp. 308–315. IEEE (2017)
26. Poklukar, P., Vasco, M., Yin, H., Melo, F.S., Paiva, A., Kragic, D.: Geometric multimodal contrastive representation learning. In: International Conference on Machine Learning, pp. 17782–17800. PMLR (2022)
27. Radford, A., et al.: Learning transferable visual models from natural language supervision. In: International conference on machine learning, pp. 8748–8763. PMLR (2021)
28. Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., Chen, M.: Hierarchical text-conditional image generation with clip latents. arXiv:2204.06125 (2022)
29. Shi, Y., Paige, B., Torr, P.H., Siddharth, N.: Relating by contrasting: A data-efficient framework for multimodal generative models. arXiv preprint arXiv:2007.01179 (2020)
30. Suzuki, M., Nakayama, K., Matsuo, Y.: Joint multimodal learning with deep generative models. arXiv preprint arXiv:1611.01891 (2016)
31. Tan, Y., Long, G., Ma, J., Liu, L., Zhou, T., Jiang, J.: Federated learning from pre-trained models: A contrastive learning approach. arXiv:2209.10083 (2022)
32. Tian, Y., Wan, Y., Lyu, L., Yao, D., Jin, H., Sun, L.: Fedbert: When federated learning meets pre-training. ACM Trans. Intell. Syst. Technol. **13**(4) (2022)
33. Tsai, Y.H.H., Bai, S., Liang, P.P., Kolter, J.Z., Morency, L.P., Salakhutdinov, R.: Multimodal transformer for unaligned multimodal language sequences. In: Proceedings of the Conference. Association for Computational Linguistics. Meeting, vol. 2019, p. 6558. NIH Public Access (2019)
34. Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S.: Caltech-ucsd birds-200-2011 (cub-200-2011). Tech. rep. (2011)
35. Wang, J., Chen, Y., Wu, Y., Das, M., Yang, H., Ma, F.: Rethinking personalized federated learning with clustering-based dynamic graph propagation. In: Pacific-Asia Conference on Knowledge Discovery and Data Mining, pp. 155–167 (2024)
36. Wang, J., Qian, C., Cui, S., Glass, L., Ma, F.: Towards federated covid-19 vaccine side effect prediction. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pp. 437–452. Springer (2022). https://doi.org/10.1007/978-3-031-26422-1_27
37. Wang, J., et al.: Towards personalized federated learning via heterogeneous model reassembly. Adv. Neural Inform. Process. Syst. **36** (2024)
38. Wang, J., Zeng, S., Long, Z., Wang, Y., Xiao, H., Ma, F.: Knowledge-enhanced semi-supervised federated learning for aggregating heterogeneous lightweight clients in iot. In: Proceedings of the 2023 SIAM International Conference on Data Mining (SDM), pp. 496–504. SIAM (2023)
39. Wu, M., Goodman, N.: Multimodal generative models for scalable weakly-supervised learning. Adv. Neural Inform. Process. Syst. **31** (2018)

40. Wu, X., Huang, F., Hu, Z., Huang, H.: Faster adaptive federated learning. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 37(9), pp. 10379–10387 (2023)
41. Wu, X., Lin, W.Y., Willmott, D., Condessa, F., Huang, Y., Li, Z., Ganesh, M.R.: Leveraging foundation models to improve lightweight clients in federated learning (2023)
42. Xiong, B., Yang, X., Qi, F., Xu, C.: A unified framework for multi-modal federated learning. Neurocomputing **480**, 110–118 (2022)
43. Yang, X., Xiong, B., Huang, Y., Xu, C.: Cross-modal federated human activity recognition via modality-agnostic and modality-specific representation learning (2022)
44. Yu, Q., Liu, Y., Wang, Y., Xu, K., Liu, J.: Multimodal federated learning via contrastive representation ensemble. In: ICLR (2023)
45. Zhao, Y., Barnaghi, P., Haddadi, H.: Multimodal federated learning on iot data. In: 2022 IEEE/ACM Seventh International Conference on Internet-of-Things Design and Implementation (IoTDI), pp. 43–54. IEEE (2022)
46. Zhou, Y., Wu, J., Wang, H., He, J.: Adversarial robustness through bias variance decomposition: a new perspective for federated learning. In: CIKM, pp. 2753–2762. ACM (2022)
47. Zolfaghari, M., Zhu, Y., Gehler, P., Brox, T.: Crossclr: Cross-modal contrastive learning for multi-modal video representations. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 1450–1459 (2021)
48. Zong, L., Xie, Q., Zhou, J., Wu, P., Zhang, X., Xu, B.: Fedcmr: federated cross-modal retrieval. In: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 1672–1676 (2021)

Contrastive Learning Enhanced Diffusion Model for Improving Tropical Cyclone Intensity Estimation with Test-Time Adaptation

Ziheng Zhou, Haojia Zuo, Ying Zhao(✉), and Wenguang Chen

Tsinghua University, Beijing, China
{zhouzihe18,zuohj19}@mails.tsinghua.edu.cn, {yingz,cwg}@tsinghua.edu.cn

Abstract. Tropical cyclone (TC) intensity estimation from satellite images is the very first and critical step of making TC forecasts, whose SOTA performance is achieved by methods built upon CNN based regression models. Unlike discriminative models trained for specific tasks, generative models on the other hand learns to comprehend data in a more sophisticated way through generation. In this paper, we explore the potential of using generative models to further improve the regression task of TC intensity estimation, distinguished from precedents that aim at classification tasks. Our proposed method ConDiff-RTTA optimizes a TC regression model during test time, by back-propagating the loss of a diffusion model conditioned on the regression outputs. More importantly, by enhancing the diffusion model's training process with our proposed contrastive loss, the diffusion model is more likely to align diffusion losses with prediction errors of the regression model. This enhancement leads to a better understanding of incorrect conditions which facilitates the adaptation of the regression model. We evaluate our proposed method on a benchmark dataset TCIR, where TCs of the latest two years are used as testing cases. Experimental results show that our proposed method ConDiff-RTTA improves the regression model in overall performance, especially on high intensity tropical cyclones. Our code is publicly available at https://github.com/maxmaxcu/ConDiff-RTTA/.

Keywords: Tropical cyclone · Intensity estimation · Diffusion Models · Test-time adaptation · Regression · Contrastive learning

1 Introduction

Tropical cyclones (TC) are among the most catastrophic weather events that can cause injuries and deaths as well as huge economic losses. Tropical cyclone monitoring and forecasting are among the most concerned missions for meteorologists and weather service centers worldwide. The very first and critical step of making TC forecasts is intensity estimation, which is defined as the maximum sustained

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
A. Bifet et al. (Eds.): ECML PKDD 2024, LNAI 14949, pp. 418–434, 2024.
https://doi.org/10.1007/978-3-031-70378-2_26

surface wind speed near the TC center (measured in knot, 1 kt $\approx$ 0.51 ms^{-1}). Since tropical cyclones usually occur on the open ocean, satellite images are mostly used for estimating the intensity. Traditional methods, such as Dvorak [8], DAV [22], and ADT [21] are based on cloud patterns recognized from satellite images. Recently, many efforts have been made in developing Neural Network (NN) based models [1–3,6,10,24,29,31] for the task of TC intensity estimation, which has become a promising direction to achieve more accurate estimations. All of these models are discriminative models that are inspired by the ability of automatically learning useful features from satellite images with various network architectures, data pre-processing methods or physics guided feature extractions. Backbone models of these works are often CNN based regression models, which predict numerical TC intensities directly.

Alternative to discriminative models, generative models are trained on a harder task, forcing them to learn a deeper and more sophisticated comprehension of the data so as to synthesize new samples, thereby improving their potential for discriminative tasks especially under limited data [11,20]. Inspired by the recent advancements of diffusion models that show promising ability in synthesizing high quality images following class or text conditions [7,12–14,26–28], there emerge a number of studies [4,5,17,18,23] that aim to unleash the potential of diffusion models on discriminative tasks. Among them, Diffusion-TTA [23] uses a pre-trained conditional diffusion model to tune an image classifier during test time and observes improvements on accuracy over the original classifier. The two models are attached in a way that, the classifier output serves as the condition to the diffusion model, such that the classifier can be adapted by back-propagating the diffusion loss.

It is natural to utilize diffusion models in a similar way on regression tasks to achieve improved TC intensity estimations. Unlike classification tasks, where the predicted attributes are categorical, regression tasks predict ordinal and numerical attributes, and face additional challenges. To successfully tune a regressor in a gradient descent manner, it should hold that given a biased prediction, the gradient of the diffusion loss on the condition points to the direction toward the ground truth, considering the ordinal nature of attributes to be regressed on. Existing studies like Diffusion-TTA focus on classification tasks and have not yet inspected into this problem. Furthermore, the level of diffusion loss should be connected to the degree in which the condition is biased, so as to encourage the expected gradient. However conditional diffusion models are typically trained with only correct conditions, lacking penalties for the incorrect ones let alone such "distance awareness", which could result in sub-optimal results.

In this paper, we propose a method driven by a contrastive learning enhanced diffusion model that meets the aforementioned challenges and can better resolve the tropical cyclone intensity estimation task. The main contributions of this paper are the following:

1. We propose a test-time adaptation method **Diff-RTTA** to improve performances of regression models utilizing diffusion models, and observe favorable loss characteristics that lead adaptations towards more accurate predictions.

2. We propose the **ConDiff-RTTA** method to enhance the diffusion model with contrastive learning such that it is aware of the distance between true and false conditions, which further optimizes the model to be more aligned with the regression task.
3. We conduct experiments on a benchmark dataset of TC intensity estimation and observe performance gains with our method, especially on high intensity tropical cyclones.

The rest of this paper is organized as follows. We first give a brief overview of related work in Sect. 2. Then, we introduce the preliminary knowledge on both diffusion models and test-time adaptation with diffusion models, and propose our constractive learning enhanced diffusion models in Sect. 3. Experimental results of our proposed method on TC benchmark dataset TCIR are shown in Sect. 4. Finally, we make concluding remarks in Sect. 5.

2 Related Work

Neural Network Models for TC Intensity Estimation. Neural network based models for the TC intensity estimation problems fall into two categories in terms of their outputs, i.e., classification models and regression models. Classification models, e.g. [10,24], output TC categories or TC intensity ranges instead of the numerical intensity value, whose performance is inferior to that of regression models [1–3,6,29,31] in terms of estimation accuracy in RMSE or MAE. For regression models, recent works mostly focus on physics guided methods, through using extra data or features as inputs [2,31,32], or designing loss functions with TC knowledge [29,31]. Some works also focus on the network design [1,2], suggesting that the neural network should not be too deep and need to exclude dropout layers.

Diffusion Generative Models for Discriminative Tasks. There have been continuing attempts in aiming to unleash the potential of generative models on discriminative tasks, dated back to early studies [11,20,25]. With recent advancements in diffusion models, a number of works [4,5,17] face this challenge by sharing the idea that, a mildly noised image should be denoised by a diffusion model with the best effect when given the correct condition. In this light, they transform either class-conditional or text-to-image diffusion models to image classifiers by enumerating through classes and converting their corresponding diffusion losses to class probabilities. Diffusion models can also be seen as teacher models to optimize dedicated discriminative student models. DreamTeacher [18] distills knowledge from generative models pre-trained on large datasets onto a discriminative backbone, which is later trained on small downstream datasets. Diffusion-TTA [23] back-propagates the diffusion loss to a classifier, allowing test-time adaptation to improve the classification accuracy of the discriminative model. Our work is more similar to the latter than the former as we target at TC intensity estimation, an area where data are limited due to the fact that satellite images of TCs are only available since past few decades and have to be fully used by both generative and discriminative models, not allowing for an up- and down-stream split.

Contrastive Learning to Capture Data Divergence. By contrasting positive samples with negative ones, contrastive learning serve as a powerful tool to capture various forms of divergence in the data. Such divergence could be data mismatching, label differences, or even the precise distances between values. For instance, SupCon [15] projects data to positions in the embedding space according to their class labels, and Rank-N-Contrast [30] further extends the idea to continuous label values, making embeddings repel each other in a degree of their label distances. CoDi [16] aims to generate tabular data entries which consist of both continuous and discrete parts by two co-evolving diffusion models, and penalizes mismatching between the two parts by utilizing contrastive learning. Inspired by these works, we use contrastive learning to make our diffusion model not only able to capture image-condition mismatching, but also be "distance aware" of correct and biased conditions.

3 Methodology

3.1 Preliminaries

Diffusion Models. For an image x sampled from the real data distribution $x \sim q(x)$, a diffusion model learns to approximate the data distribution by gradually adding noise to x in the diffusion process and predict the noise in the reverse process. Conditional diffusion models further learns the distribution $q(x|c)$, where c is the condition input corresponding to the image x. The diffusion process, where a sequence of noise are added to the original input image x (now denoted as x_0) generating a noised image sequence $x_1, x_2, ..., x_T$, is formally defined [12] as:

$$\begin{aligned} q(x_{1:T}|x_0) &:= \prod_{t=1}^{T} q(x_t|x_{t-1}), \\ q(x_t|x_{t-1}) &:= N(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t \mathbf{I}), \end{aligned} \tag{1}$$

where $\beta_1, ...\beta_T$, is a variance schedule that controls the level of the noise. We can further sample x_t from x_0 using

$$x_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, \epsilon \sim \mathcal{N}(0, \mathbf{I}), \tag{2}$$

where $\alpha_t := 1 - \beta_t$ and $\bar{\alpha}_t := \prod_{s=1}^{t} \alpha_s$

A diffusion denoising network $\epsilon_\phi(x_t, t)$ learns to predict the noise with noisy image x_t and the noise level t as inputs. For conditional diffusion models that takes c as an input condition during the reverse process, the diffusion loss for training is defined as:

$$\mathcal{L}_{\text{diff}}(\phi; \mathcal{D}) = \frac{1}{|\mathcal{D}|} \sum_{(x^i, c^i) \in \mathcal{D}} \| \epsilon_\phi(\sqrt{\bar{\alpha}_t}x^i + \sqrt{1-\bar{\alpha}_t}\epsilon, c^i, t) - \epsilon \|^2 \tag{3}$$

where $\mathcal{D} = \{(x^i, c^i)\}_{i=1}^N$ is a training batch of N images with their corresponding conditions (labels).

Note that for the sake of simplicity, the above formulations are from DDPM [12], an origin of diffusion models. In our work we follow the framework of EDM [14] which includes altered design choices that boost the generative ability.

Test-Time Adaptation with Diffusion Models. Test-time adaptation refers to a procedure in which a pre-trained model is adapted on *unlabeled* test data [19]. Without labels, what is helpful to the adapted model can be another model that contains better knowledge about the test data. Diffusion-TTA [23] tackles this by using a pre-trained diffusion model, and tune an image classifier in an iterative manner. First, the classifier does inference on an image to provide an initial guess of class probabilities, from which a class condition is synthesized as weighted mixing of class embeddings. Then, a noise batch of different strengths is added onto the image, as inputs into the diffusion model along with the synthesized condition to compute the conditional diffusion loss. Last, loss gradients are back-propagated to the classifier, updating it to produce new class probabilities for the next iteration. After a specified number of iterations, the classifier is optimized on the image sample to produce a more accurate classification result with the help of the diffusion model, yielding better performance on the test set.

3.2 Conditional Diffusion Model for a Regression Task

Existing works that use diffusion models in discriminative tasks are limited to using categorical conditions such as one-hot class labels or text embedding during the training and inference of diffusion models. This raises a direct question that whether regression tasks can benefit from conditional diffusion models as well. In our TC intensity estimation task, the intensity is a numerical number with its range from 10 kt to 180 kt. Given the continuous nature of labels in regression tasks, it is infeasible to build a generative regressor by enumerating through labels as conditions and infer the target from corresponding conditional diffusion losses as in [4,17]. Therefore we build our method on top of Diffusion-TTA which is gradient based.

Towards this goal, we migrate Diffusion-TTA to TC intensity estimation in a simple yet effective fashion. We follow the process of Diffusion-TTA and make modifications to take the TC intensity value as the condition, instead of class text embedding as in Diffusion-TTA. First we train a conditional diffusion model on an open dataset of TCs (will be described in Sect. 4.1), where the intensity condition is passed through a linear layer, projected to an embedding vector and taken by the diffusion model. Then we take a CNN-based TC intensity regression model [31] and conduct TTA on it in an instance-wise manner. We denote this method as **Diff-RTTA**, whose overall architecture and pseudo code are shown in Fig. 1 and Algorithm 1, respectively.

We see improvements on Diff-RTTA over the regression model (reported in Sect. 4.4), but in this phase of study what we mainly want to inspect is the reason

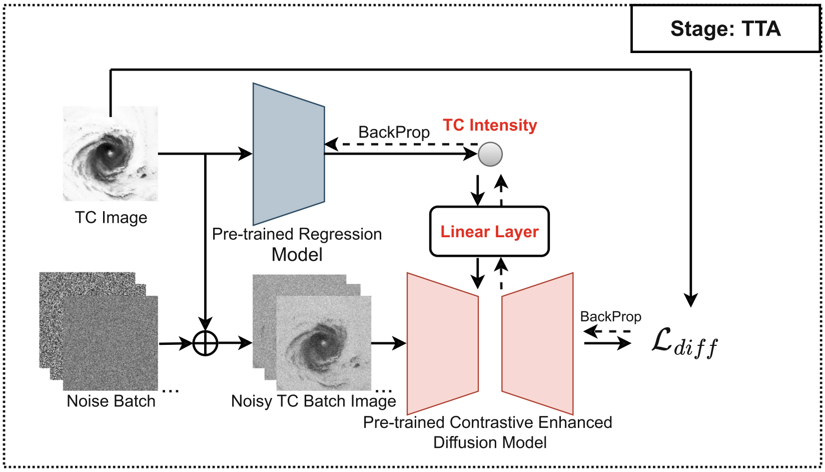
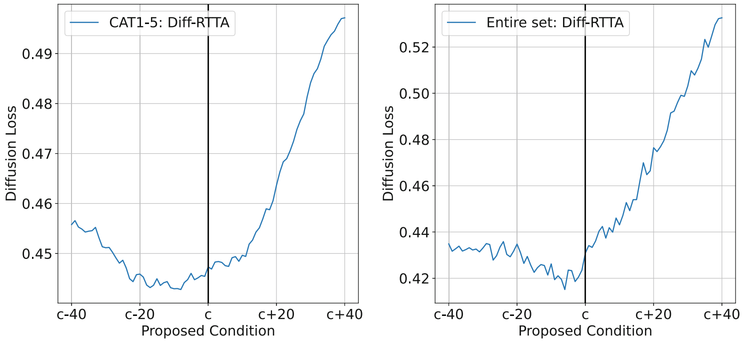


Fig. 1. Overall Architecture for Test-time Adaptation

Algorithm 1 Test-time Adaptation

```
Require: Test image $x$, regression model weights $\theta$, diffusion model weights $\phi$, adaptation steps $N$
for $s \in [1, N]$ do
    Do inference on regression model to get prediction $\hat{c} \leftarrow f_\theta(x)$
    Project $\hat{c}$ to embedding $\boldsymbol{e}_{\hat{c}}$ by the linear layer
    Sample noise strength batch $\boldsymbol{t}$ following settings of Diffusion-TTA
    Sample noise batch $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I})$
    repeat $x$ to build batch $\boldsymbol{x}$
    Compute $\mathcal{L}_{\text{diff}} = \| \epsilon_\phi(\boldsymbol{x} + \boldsymbol{t} \odot \boldsymbol{\epsilon}, \boldsymbol{e}_{\hat{c}}, \boldsymbol{t}) - \boldsymbol{\epsilon} \|^2$
    Take gradient descent step on $\nabla_\theta \mathcal{L}_{\text{diff}}$ to update $\theta$
end for
return $f_\theta(x)$
```

why a diffusion model can indeed benefit regression tasks. To demonstrate it, for every TC image we enumerate the intensity condition as an integer from 10 kt to 180 kt, and collect diffusion losses over the enumeration. It is expected that the diffusion loss should be minimal at the correct intensity of the TC image. Figure 2 shows the loss enumerations on test set for TCs of CAT1-CAT5 categories (well be defined in Sect. 4.1) and the entire set. It can be observed that the loss curves tend to be U-shaped with the valley near the correct condition (denoted by c). With the U-shaped loss curves, it is made possible that a biased proposed value of intensity could be optimized towards the ground truth intensity by steps of gradient descent, whereas enumeration on possible conditions is far more costly for continuous values.



(a) CAT1-CAT5 (b) Entire Set

Fig. 2. Diffusion loss enumerations over conditions by Diff-RTTA: For each TC image, diffusion losses are calculated on each condition enumerated from range [c-40,c+40], where c is the true condition of the corresponding TC image. The average of diffusion losses on each condition offset value of all TC images from CAT1 to CAT5 categories and from the entire test set are shown in (a) and (b), respectively.

3.3 Contrastive Learning Enhanced Diffusion Model

Observations on Diff-RTTA indicate that, by following the vanilla training procedure, a diffusion model conditioned on numerical values can exhibit our expected characteristic: the loss enumeration curve is U-shaped around the true condition. In other words, the diffusion model denoises noisy images the best around the true condition, and behaves worse when the proposed condition is farther away. Nevertheless, we suppose this favorable characteristic can be even strengthened, since the vanilla training way of the conditional diffusion model assumes the conditions are always correct, thus paying no attention on the relation between true and false conditions and their distances. It is reasonable because such knowledge can hardly be of use for pure generation, but it comes to importance in the context of our study. We expect that explicitly relating diffusion loss to condition distances can point the gradient more to the correct direction, and reduce the bias between the loss minimum point and correct condition.

Similar ideas can be seen in contrastive learning literature such as [30]. This motivates us to explore supervised contrastive learning for the enhancement. Contrastive learning works by contrasting similar samples (positive samples) with dissimilar ones (negative samples). In the TC estimation scenario, for a TC image x with its true condition c, a positive-negative pair is defined as:

$$\begin{aligned} \text{Positive} : [\text{aug}(x), c_{\text{pos}}], \\ \text{Negative} : [\text{aug}(x), c_{\text{neg}}], \end{aligned} \tag{4}$$

where $\text{aug}(\cdot)$ is a data augmentation function, $c_{\text{pos}} := c$ and c_{neg} is a false condition not equal to c_{pos}. Concerning the ordinal nature of our conditions and the local gradient we pursue, we sample the negative condition in a neighborhood of the positive, which also serves as a harder negative compared to some arbitrarily positioned one. There are also common observations that regression models tend to exhibit larger estimation errors on TCs of high intensities [1,2,29,31,32], therefore we enlarge the sampling neighborhood for high intensities to cover the potential error bar with negative samples. The sampling strategy is defined as

$$c_{\text{neg}} = c_{\text{pos}} + \text{rand}(-(\log c_{\text{pos}})^2, (\log c_{\text{pos}})^2), \tag{5}$$

where $\text{rand}(a, b)$ draws a random number from a uniform distribution on the interval (a, b).

With the defined positive-negative pair, we propose a contrastive loss term in the form of a triplet loss, which is formally defined as

$$\mathcal{L}_{\text{con}} = \max\left(\mathcal{L}_{\text{diffpos}} - \mathcal{L}_{\text{diffneg}} + \text{Margin}(c_{\text{pos}}, c_{\text{neg}}), 0\right), \tag{6}$$

where $\mathcal{L}_{\text{diffpos}}$ is the diffusion loss for the positive sample and $\mathcal{L}_{\text{diffneg}}$ is the diffusion loss for the negative sample. In the standard triplet loss, margin is defined as a constant to keep the positive away from the negative in a certain degree. Here, we propose margin as a distance aware function so that it adjusts the margin between positive and negative losses according to the distance between the corresponding conditions. With a larger distance, the negative loss should exceed the positive loss to a greater extent. The Margin function is defined as

$$\text{Margin}(c_{\text{pos}}, c_{\text{neg}}) = \log\left(1 + D(c_{\text{pos}}, c_{\text{neg}})\right) * \mathcal{L}_{\text{diffpos}}, \tag{7}$$

where $D(c_{\text{pos}}, c_{\text{neg}})$ is the distance between true and false conditions, and $\mathcal{L}_{\text{diffpos}}$ here only provides the value without contributing a gradient. We choose the current form to let the margin shrink when c_{neg} gets close to c_{pos}. The margin is also proportional to $\mathcal{L}_{\text{diffpos}}$ because the loss scale differs through conditions and the margin should be adjusted in a relative manner. With this distance aware margin, the diffusion model learns to increase the diffusion loss under a false condition adaptive to the condition distance and the loss scale.

We propose the following contrastive learning enhanced diffusion loss for continuous training on the previously trained diffusion model,

$$\mathcal{L}_{\text{ConDiff}} = \mathcal{L}_{\text{diff}} + \lambda \mathcal{L}_{\text{con}} \tag{8}$$

where λ is the weight for the contrastive loss, which is a hyper parameter. The training procedure is modified from the standard procedure of training a conditional diffusion model, where in each iteration the batch is doubled to construct the negative half whose conditions are sampled according to Eq. 5, and the doubled batch is fed into the model to update it via $\mathcal{L}_{\text{ConDiff}}$. The contrastive learning enhanced diffusion model is then used in the TTA stage. We denote this improved method as **ConDiff-RTTA**.

The pipeline for the contrastive enhanced diffusion model training phase is shown in Fig. 3. The overall pseudo code for training is shown in Algorithm 2.

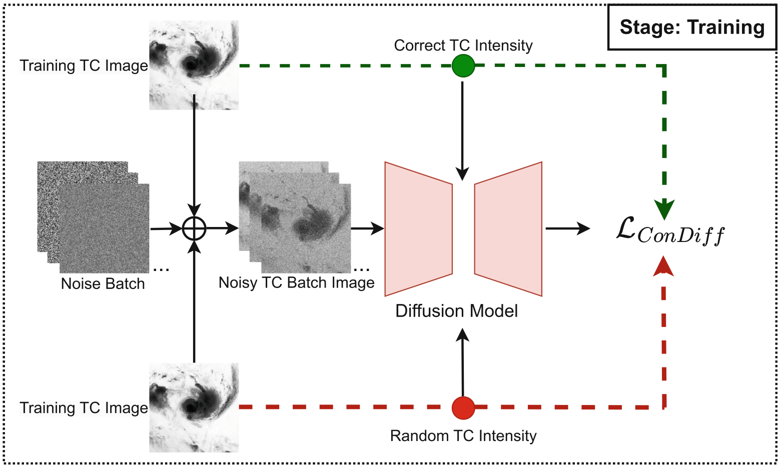


Fig. 3. Overall Architecture for Constructive Enhanced Diffusion Model Training

```
Algorithm 2 Training of Contrastive Learning Enhanced Diffusion Model
Require: training set $\mathcal{D}_{\text{train}}$, diffusion model weights $\phi$
while not converged do
    Sample training image-intensity batch $(\boldsymbol{x}_0, \boldsymbol{c}_{\text{pos}}) \sim \mathcal{D}_{\text{train}}$
    Sample noise strength batch $\boldsymbol{t}$ following settings of EDM
    Sample noise batch $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I})$
    $\boldsymbol{x}_t \leftarrow \boldsymbol{x}_0 + \boldsymbol{t} \odot \boldsymbol{\epsilon}$
    Synthesize false conditions $\boldsymbol{c}_{\text{neg}}$ according to Eq. 5
    Compute $\mathcal{L}_{\text{ConDiff}}$ according to Eq. 8
    Take gradient descent step on $\nabla_\phi \mathcal{L}_{\text{ConDiff}}$ to update $\phi$
end while
```

4 Experiments

4.1 Dataset

We use a publicly available benchmark dataset, the Tropical Cyclone Dataset for Image Intensity Regression (TCIR)[1] [1]. TCIR contains TCs in the North Eastern Pacific, the North Western Pacific, and the Atlantic Ocean. The satellite observations in TCIR are derived from two open datasets, GridSat and CMORPH. The best track intensities (IBTrACS) are derived from the Joint Typhoon Warning Center (JTWC) and the Atlantic Hurricane Database (HURDAT2).

As shown in Table 1, we classify TCs according to the Saffir-Simpson Hurricane Wind Scale, which consists of 7 classes, with higher classes representing higher maximum sustained winds. We use a total of 36566 image frames from TCs in 2003-2013 as training data, 3245 from TCs in 2014 as validation data,

---

[1] Available at https://www.csie.ntu.edu.tw/~htlin/program/TCIR/.

and 7570 frames from TCs in 2015-2016 as testing data. Each frame has 201×201 pixels and a total of 4 channels per pixel, i.e., infrared (IR), water vapor (WV), visible channel (VIS), and passive microwave rain-rate (PMW). In our experiments, we use the IR channel, and normalize it to have zero mean and unit standard deviation, and resize it to 65×65 pixels as the input.

Table 1. Number of Frames in TCIR from [1]

Category	Training	Validation	Testing
TD ($33 \leq$kt)	13766	1154	2353
TS ($34 \sim 63$ kt)	13850	1194	3048
CAT1 ($64 \sim 82$ kt)	3793	388	787
CAT2 ($83 \sim 95$ kt)	1909	178	490
CAT3 ($96 \sim 112$ kt)	1381	129	418
CAT4 ($113 \sim 136$ kt)	1558	147	394
CAT5 (≥ 137 kt)	309	55	80
Total	36566	3245	7570

4.2 Models and Metrics

Regression Model. To achieve SOTA performance on TC intensity estimation, it is needed to include physics-guided features in the regression network, and conduct special post-processing such as sliding windows and rotation ensembles [1,2,29,31,32]. These techniques are along different dimensions compared to our method, and will require a lot of extra efforts and computational resources. Therefore we set our goal to explore the ability of diffusion models on improving the intensity estimation performance of CNN based backbone models. We use ResNet-18 [9] as the backbone for feature extraction and train it on the TCIR training set with L2 loss. This regression model achieves comparable performance to backbone models in [1,31] on the TCIR validation set and test set. We refer to this model as Regression or Reg Model in the experiments.

Diffusion Models. For diffusion models, we use the implementation framework of EDM [14] and the U-Net backbone model from [28]. **Diff-RTTA**: This is a diffusion model trained with our modification for the regression task as discussed in Sect. 3.2. The trained diffusion model is then used to adapt the Reg Model during test time. **ConDiff-RTTA**: We fine-tune the above diffusion model with our proposed $\mathcal{L}_{\text{ConDiff}}$ loss. Same as in Diff-RTTA, $\mathcal{L}_{\text{diff}}$ is used during TTA stage.

Evaluation Metrics. We report the TC intensity estimation accuracy of various models in terms of Root Mean Square Error (RMSE) and Mean Absolute Error (MAE).

4.3 Implementation Details

Models are trained on 8 RTX 4090 GPUs. We train a total of 21M TC images randomly sampled from the training dataset with batch size 256 for the Diff-RTTA and continue to train 3M TC images with batch size 128 for the ConDiff-RTTA, with the rest of training settings following the default of EDM. For test-time adaptation, the noise batch size is 200 with 10 adaptation steps and Adam optimizer is used with a learning rate of 5×10^{-5}.

4.4 Overall Performance

Diff-RTTA as Regression Model. To get a better understanding on using diffusion model alone as a regression model, we test the performance of Diff-RTTA model without using the pre-trained Reg Model but instead with 50 kt as the initial conditional inputs for all the test TC images. 50 kt is the mean value of the TC intensities from training set and has an overall RMSE of 30.39 on the entire test set. The performance of using 50 kt as initial conditions with Diff-RTTA is shown in Table 2 labelled as Diff-RTTA (50). Even with the initial condition of 50 kt, the overall RMSE of Diff-RTTA improves to 14.83, showing its ability as a regression model. In Fig. 4, Diff-RTTA (50) results for each TC image are ordered by the true conditions (from IBTrACS) from left to right. We can see that the predicted intensities are spread along the true conditions, which indicates the important fact that the correct adaptation directions are likely to be found using the diffusion loss as the feedback.

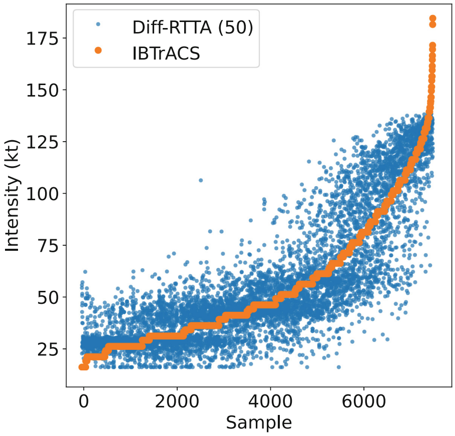


Fig. 4. Diff-RTTA (50) results for each TC image

Comparisons to Baselines. The performances of Reg Model, Diff-RTTA and ConDiff-RTTA are shown in Table 2. Diff-RTTA shows an improvement of 0.33

on RMSE over Reg Model, from 11.22 to 10.89. ConDiff-RTTA further improves the overall performance to 10.76. Although ConDiff-RTTA achieves mildly better results over Diff-RTTA in overall performance, a detailed inspection reveals that improvements on each TC category are made differently, as shown in Fig. 5. ConDiff-RTTA shows more significant improvements over Diff-RTTA as the TC intensity becomes higher, roughly between 0.6 to 1.0 compared to Reg Model on CAT1-5, in which the most destructive TCs reside. We attribute this observation to the stronger contrastive effect on high intensities due to larger contrastive margins and wider negative sampling windows, which we design deliberately to enhance the regression model's performance on strong TCs.

Table 2. RMSE and MAE results on TCIR test set

	TD		TS		CAT1-5		Overall	
Method	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
Regression	6.76	4.95	9.89	7.70	15.76	12.16	11.22	8.17
Diff-RTTA (50)	9.25	6.91	12.49	9.28	21.21	17.12	14.83	10.84
Diff-RTTA	**6.49**	**4.69**	9.62	7.49	15.31	11.89	10.89	7.93
ConDiff-RTTA	6.66	4.80	**9.58**	**7.43**	**14.97**	**11.57**	**10.76**	**7.85**

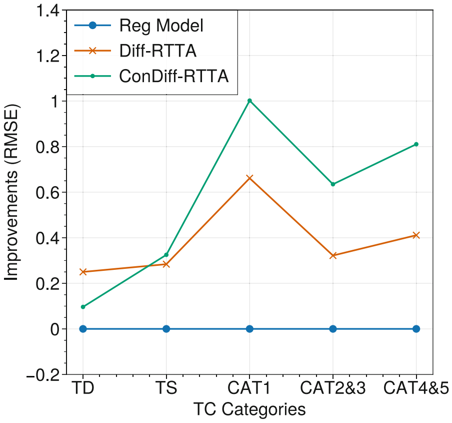
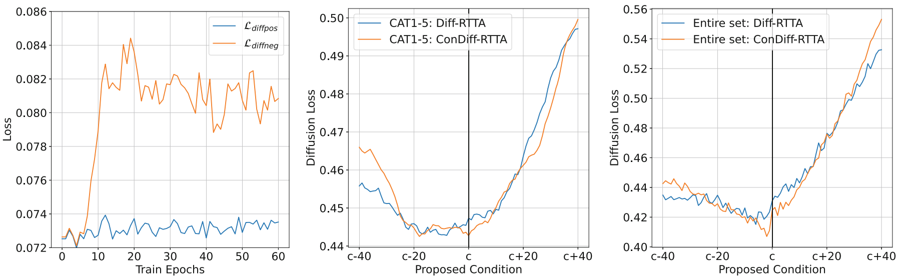


Fig. 5. Improvements on different categories over baseline Reg Model

4.5 Diffusion Loss Analysis

We show the training curves of $\mathcal{L}_{\text{diffpos}}$ and $\mathcal{L}_{\text{diffneg}}$ in Fig. 6 (a). $\mathcal{L}_{\text{diffpos}}$, the diffusion loss given true conditions, remains at a low level while $\mathcal{L}_{\text{diffneg}}$, the diffusion loss given false conditions, increases significantly during training. Diffusion loss enumerations of the diffusion model trained in Diff-RTTA and that trained in ConDiff-RTTA are shown in Fig. 6 (b) and (c), on CAT1 to CAT5 TCs and the entire test set respectively. We can see that on both figures, the enumeration curves (yellow) with ConDiff-RTTA are sharper than the curves (blue) with Diff-RTTA. The valley of the curve with ConDiff-RTTA also shifts more towards the center (true condition c) compared to Diff-RTTA. These figures comply with our intention to still learn $p(x|c)$ as well as impose stronger constraints on false condition scenarios.



(a) Training loss curves (b) CAT1-CAT5 (c) Entire Set

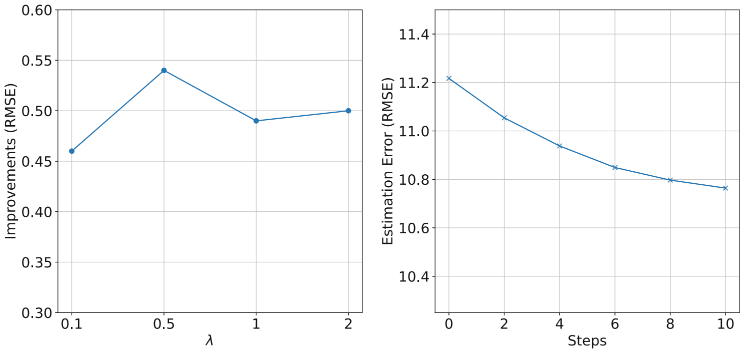
Fig. 6. (a) Training loss curves and diffusion loss enumerations over conditions on (b) CAT1-CAT5 and (c) the entire test set

4.6 Parameter Study

A parameter study is conducted using validation set on the hyper parameter λ, which is the weight of our proposed contrastive loss. Figure 7 (a) shows the overall improvements over baseline Reg Model with different λ values of 0.1, 0.5, 1.0, 2.0. It shows that the overall performance improves even with a small λ value. $\lambda = 0.5$ is selected according to our parameter study for reporting ConDiff-RTTA results. We also perform another parameter study w.r.t. the number of adaptation steps during test-time adaptation and the results are shown in Fig. 7 (b). It shows that by extending the adaptation steps, the overall RMSE keeps decreasing. As more adaptation steps lead to more running time, we stop the adaptation step at 10.

4.7 Case Study

We select from our test set the Super Typhoon Meranti, one of the most disastrous typhoons of this century for our case study. Meranti impacted South Eastern Asia and Southern China areas in September 2016, causing numerous deaths and injuries along with massive economic loss. It was recognized as a CAT5 typhoon during its peak times.



(a) Hyper Parameter λ (b) Adaptation Steps

Fig. 7. Parameter study on (a) Hyper Parameter λ and (b) Adaptation Steps

Figure 8 shows the best track intensities (from IBTrACS) and model intensity estimations of Meranti throughout its lifetime. The regression model underestimates the peak intensities, which is likely due to the rareness of violent typhoons in the nature and therefore in the TCIR dataset. As a comparison, Our proposed method ConDiff-RTTA revises the estimations upward such that they are closer to IBTrACS values. This case demonstrates that with the assist of our contrastive learning enhanced diffusion model, over-fitting in the regression model can be mitigated, resulting in a more accurate discriminative estimation on rare data.

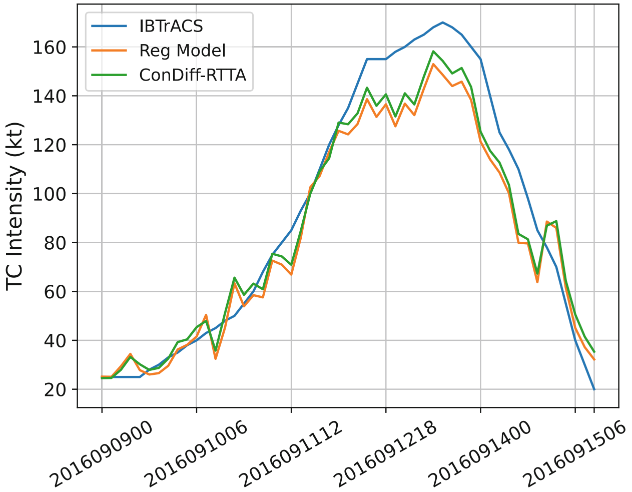


Fig. 8. The intensities of Super Typhoon Meranti over its lifetime

5 Conclusion

In this paper, we propose a new method ConDiff-RTTA to improve TC intensity estimation performance. We find that TC regression network can be optimized during test time by a diffusion model conditioned on ordinal intensity numbers instead of categorical labels as in previous works. Furthermore, we enhance the diffusion model by training in a contrastive learning approach in order to improve the alignment between diffusion losses and prediction errors of the regression model. Experimental results show that the diffusion model pre-trained from TC satellite images improves TC estimation performance, and ConDiff-RTTA achieves further overall performance gains, especially significant on high intensity TCs.

Acknowledgments. This work was supported by the National Key Research and Development Program of China (2022YFC3004102, 2017YFA0604502), Science and Technology Project of Qinghai Province (No.2023-QY-208), and High performance computing center of Qinghai University.

References

1. Chen, B., Chen, B.F., Lin, H.T.: Rotation-blended cnns on a new open dataset for tropical cyclone image-to-intensity regression. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 90–99 (2018)
2. Chen, B.F., Chen, B., Lin, H.T., Elsberry, R.L.: Estimating tropical cyclone intensity by satellite imagery utilizing convolutional neural networks. Weather Forecast. **34**(2), 447–465 (2019)
3. Chen, Z., Yu, X.: A novel tensor network for tropical cyclone intensity estimation. IEEE Trans. Geosci. Remote Sens. **59**(4), 3226–3243 (2020)
4. Chen, H., Dong, Y., Wang, Z., Yang, X., Duan, C., Su, H., Zhu, J.: Robust classification via a single diffusion model. arXiv preprint arXiv:2305.15241 (2023)
5. Clark, K., Jaini, P.: Text-to-image diffusion models are zero shot classifiers. Adv. Neural Inform. Process. Syst. **36** (2024)
6. Combinido, J.S., Mendoza, J.R., Aborot, J.: A convolutional neural network approach for estimating tropical cyclone intensity using satellite-based infrared images. In: 2018 24th International Conference on Pattern Recognition (ICPR), pp. 1474–1480. IEEE (2018)
7. Dhariwal, P., Nichol, A.: Diffusion models beat gans on image synthesis. Adv. Neural. Inf. Process. Syst. **34**, 8780–8794 (2021)
8. Dvorak, V.F.: Tropical cyclone intensity analysis and forecasting from satellite imagery. Mon. Weather Rev. **103**(5), 420–430 (1975)
9. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
10. Higa, M., et al.: Domain knowledge integration into deep learning for typhoon intensity classification. Sci. Rep. **11**(1), 1–10 (2021)
11. Hinton, G.E.: To recognize shapes, first learn to generate images. Prog. Brain Res. **165**, 535–547 (2007)

12. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. Adv. Neural. Inf. Process. Syst. **33**, 6840–6851 (2020)
13. Ho, J., Salimans, T.: Classifier-free diffusion guidance. arXiv preprint arXiv:2207.12598 (2022)
14. Karras, T., Aittala, M., Aila, T., Laine, S.: Elucidating the design space of diffusion-based generative models. Adv. Neural. Inf. Process. Syst. **35**, 26565–26577 (2022)
15. Khosla, P., et al.: Supervised contrastive learning. Adv. Neural. Inf. Process. Syst. **33**, 18661–18673 (2020)
16. Lee, C., Kim, J., Park, N.: Codi: co-evolving contrastive diffusion models for mixed-type tabular synthesis. In: International Conference on Machine Learning, pp. 18940–18956. PMLR (2023)
17. Li, A.C., Prabhudesai, M., Duggal, S., Brown, E., Pathak, D.: Your diffusion model is secretly a zero-shot classifier. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 2206–2217 (2023)
18. Li, D., et al.: Dreamteacher: pretraining image backbones with deep generative models. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 16698–16708 (2023)
19. Liang, J., He, R., Tan, T.: A comprehensive survey on test-time adaptation under distribution shifts. arXiv preprint arXiv:2303.15361 (2023)
20. Ng, A., Jordan, M.: On discriminative vs. generative classifiers: a comparison of logistic regression and naive bayes. Adv. Neural Inform. Process. Syst. **14** (2001)
21. Olander, T.L., Velden, C.S.: The advanced dvorak technique: continued development of an objective scheme to estimate tropical cyclone intensity using geostationary infrared satellite imagery. Weather Forecast. **22**(2), 287–298 (2007)
22. Piñeros, M.F., Ritchie, E.A., Tyo, J.S.: Estimating tropical cyclone intensity from infrared image data. Weather Forecast. **26**(5), 690–698 (2011)
23. Prabhudesai, M., Ke, T.W., Li, A., Pathak, D., Fragkiadaki, K.: Test-time adaptation of discriminative models via diffusion generative feedback. Adv. Neural Inform. Process. Syst. **36** (2024)
24. Pradhan, R., Aygun, R.S., Maskey, M., Ramachandran, R., Cecil, D.J.: Tropical cyclone intensity estimation using a deep convolutional neural network. IEEE Trans. Image Process. **27**(2), 692–702 (2017)
25. Raina, R., Shen, Y., Mccallum, A., Ng, A.: Classification with hybrid generative/discriminative models. Adv. Neural Inform. Process. Syst. **16** (2003)
26. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10684–10695 (2022)
27. Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models. arXiv preprint arXiv:2010.02502 (2020)
28. Song, Y., Sohl-Dickstein, J., Kingma, D.P., Kumar, A., Ermon, S., Poole, B.: Score-based generative modeling through stochastic differential equations. In: International Conference on Learning Representations (2021). https://openreview.net/forum?id=PxTIG12RRHS
29. Tian, W., Zhou, X., Huang, W., Zhang, Y., Zhang, P., Hao, S.: Tropical cyclone intensity estimation using multi-dimensional convolutional neural network from multi-channel satellite imagery. IEEE Geosci. Remote Sensing Lett. (2021)
30. Zha, K., Cao, P., Son, J., Yang, Y., Katabi, D.: Rank-n-contrast: Learning continuous representations for regression. In: Thirty-seventh Conference on Neural Information Processing Systems (2023)

31. Zhou, Z., Zhao, Y., Qing, Y., Jiang, W., Wu, Y., Chen, W.: A physics-guided nn-based approach for tropical cyclone intensity estimation. In: Proceedings of the 2023 SIAM International Conference on Data Mining (SDM), pp. 388–396. SIAM (2023)
32. Zhuo, J.Y., Tan, Z.M.: Physics-augmented deep learning to improve tropical cyclone intensity and size estimation from satellite imagery. Mon. Weather Rev. **149**(7), 2097–2113 (2021)

BESTMVQA: A Benchmark Evaluation System for Medical Visual Question Answering

Xiaojie Hong[1], Zixin Song[1], Liangzhi Li[2](✉), Xiaoli Wang[1](✉), and Feiyan Liu[1]

[1] School of Informatics, Xiamen University, Xiamen, China
{xjhong,zxsong,feiyanliu}@stu.xmu.edu.cn, xlwang@xmu.edu.cn
[2] Meetyou AI Lab, Xiamen, China
liliangzhi@xiaoyouzi.com

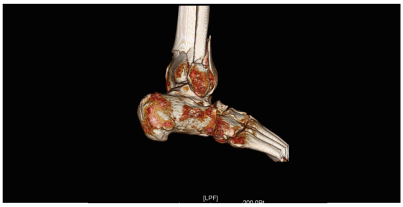
Abstract. Medical Visual Question Answering (Med-VQA) is a task that answers a natural language question with a medical image. Existing VQA techniques can be directly applied to solving the task. However, they often suffer from (*i*) the data insufficient problem, which makes it difficult to train the state of the arts (SOTAs) for domain-specific tasks, and (*ii*) the reproducibility problem, that existing models have not been thoroughly evaluated in a unified experimental setup. To address the issues, we develop a Benchmark Evaluation SysTem for Medical Visual Question Answering, denoted by BESTMVQA. Given clinical data, our system provides a useful tool for users to automatically build Med-VQA datasets. Users can conveniently select a wide spectrum of models from our library to perform a comprehensive evaluation study. With simple configurations, our system can automatically train and evaluate the selected models over a benchmark dataset, and reports the comprehensive results for users to develop new techniques or perform medical practice. Limitations of existing work are overcome (*i*) by the data generation tool, which automatically constructs new datasets from unstructured clinical data, and (*ii*) by evaluating SOTAs on benchmark datasets in a unified experimental setup. The demonstration video of our system can be found at https://youtu.be/QkEeFlu1x4A, and the source code is shared on https://github.com/emmali808/BESTMVQA.

Keywords: Medical Visual Question Answering · Benchmark Evaluation System · Comprehensive Experimental Study

1 Introduction

Medical visual question answering is a challenging task in healthcare industry, which answers a natural language question with a medical image. Figure 1 shows an example of the Med-VQA data. It may aid doctors in interpreting medical images for diagnoses with responses to close-ended questions, or help patients

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
A. Bifet et al. (Eds.): ECML PKDD 2024, LNAI 14949, pp. 435–451, 2024.
https://doi.org/10.1007/978-3-031-70378-2_27



Question	Is this image abnormal?
Answer	Yes.
Question Type	Abnormality
Answer Type	CLOSED
Image Organ	LEG

Fig. 1. An example of Med-VQA

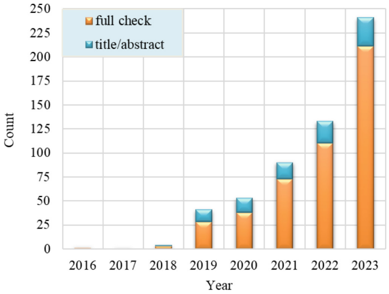


Fig. 2. Publications on Med-VQA since 2016

with urgent needs get timely feedback on open-ended questions [13]. It is a challenging problem which processes multi-modal information. Different from general VQA, Med-VQA requires substantial prior domain-specific knowledge to thoroughly understand the contents and semantics of medical visual questions.

Many exiting techniques contribute to solving this task (e.g., [9]). However, they generally suffer from the data insufficient problem. They need to be trained on well-annotated large datasets, to learn enough domain-specific knowledge for understanding medical visual questions. Several works focus on constructing Med-VQA datasets [2,11,12,15,17]. However, these datasets seem to be a drop in the bucket. Other works employ data augmentation method to tackle the problem. VQAMix [9] has focused on generating Med-VQA training samples. However, it may incur noisy samples that affect the performance of models. Current work have adopted transfer learning to pre-train a visual encoder on external medical image-text pairs to capture suitable visual representations for subsequent cross-modal reasoning [6,9,13]. They achieve success by performing pre-training using large-scale data unannotated data. However, they have not been thoroughly evaluated in benchmark settings.

To address the problems, we develop BESTMVQA, which is a benchmark evaluation system for Med-VQA. We first provide a data generation tool for users to automatically construct new datasets from self-collected clinical data. We implement a wide spectrum of SOTA models for Med-VQA in a model library. Accordingly, users can conveniently select a benchmark dataset and any model in model library for medical practice. Our system can automatically train the models and evaluate them over the selected dataset, and present a final comprehensive report to users. With our system, researchers can comprehensively study SOTA models and their applicability in Med-VQA. The impact of our contributions also can be inferred from Fig. 2, which shows the significant increase in Med-VQA publications since 2016. We provide a unified evaluation system for users to (*i*) reveal the applicability of SOTA models to benchmark datasets, (*ii*) conduct a comprehensive study of the available alternatives to develop new Med-VQA techniques, and (*iii*) perform various medical practice.

2 Research Scope and Task Description

The research scope is tailored to two types of readers: (*i*) Researchers who require Med-VQA techniques to perform downstream tasks; (*ii*) Contributors in the research community of Med-VQA who need to thoroughly evaluate the SOTAs.

Medical visual question answering is a domain-specific task that inputs a medical image and a related question, outputting an answer in natural language. It requires extensive domain knowledge, adding complexity beyond general VQA tasks. The lack of well-annotated large-scale datasets makes it hard to learn enough medical knowledge. To address the challenge, current work typically pre-train a visual encoder on large unlabeled medical image-text pairs.

In Fig. 3, Med-VQA models consist of four main components: vision encoder, text encoder, feature fusion, and answer prediction, which together process the image and question inputs to predict answers.

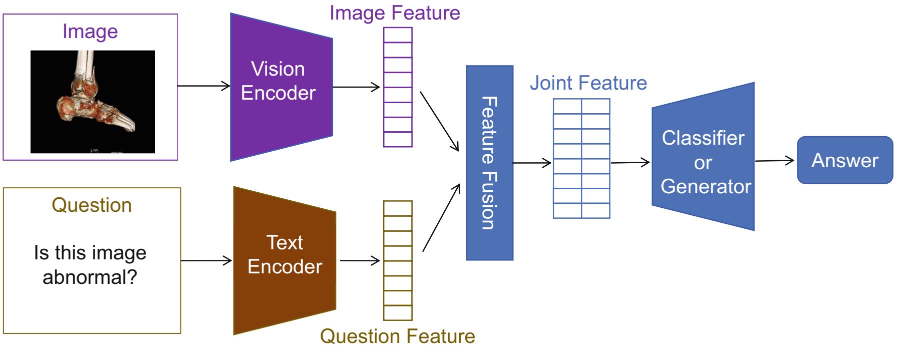


Fig. 3. The architecture of mainstream Med-VQA models

3 Related Work

Med-VQA is a challenging task that combines natural language processing and computer vision. Early work employing traditional machine learning algorithms suffers from poor performance due to significant differences between visual and textual features [26]. Inspired by the success of deep learning in information systems, deep learning models for Med-VQA are reported to have performance gains over traditional models [23]. They can be classified into four categories: joint embedding, encoder-decoder, attention-based, and large language models (LLMs). Table 1 shows the statistics of SOTAs we reproduced.

The joint embedding models combine visual and textual embeddings into a final representation. We implement some representative models such as MEVF [19] and CR [32]. MEVF uses MAML [7] and CDAE [18] to initialize

Table 1. The statistics of considered models, including the parameter size (Params), the training time (Training Time), supporting pre-training or not (Support PT), supporting fine-tuning or not (Support FT) and model category (Model Category). The left value of Training Time represents the smallest training time over all datasets, while the right value is the largest one.

Baseline	Params	Training Time	Support PT	Support FT	Model Category
MEVF [19]	15M	0.03 h–0.3 h	×	✓	Joint Embedding
CR [32]	38M	0.04 h–0.4 h	×	✓	Joint Embedding
MMQ [4]	20M	0.5 h–3.0 h	✓	✓	Joint Embedding
VQAMix [9]	19M	0.6 h–6.0 h	×	✓	Joint Embedding
CMSA [8]	88M	1.0 h–4.2 h	×	✓	Attention-Based
MMBERT [14]	117M	1.7 h–13.3 h	✓	✓	Attention-Based
PTUnifier [3]	350M	3.0 h–13.0 h	✓	✓	Attention-Based
METER [5]	320M	2.5 h–18.0 h	✓	✓	Attention-Based
TCL [29]	580M	1.3 h–8.3 h	×	✓	Encoder-Decoder
MiniGPT-4 [34]	14110M	-	×	×	LLMs
LLaVA-Med [16]	6743M	-	×	×	LLMs

the model weights for visual feature extraction, while CR proposes question-conditioned reasoning and task-conditioned reasoning modules for textual feature extraction.

For encoder-decoder models, visual and textual features are extracted separately by encoders, and fused in a feature fusion layer. The decoder generates the answer based on the fused features. NLM [21], TCL [29], and MedVInT [33] are such representative models.

The third category employs attention mechanisms to capture representative visual and textual features. MMBERT [14] employ Transformer-style architecture to extract visual and textual features. CMSA [8] introduce a cross-modal self-attention module to selectively capture the long-range contextual relevance for more effective fusion of visual and textual features. MedFuseNet [22] excels in open-ended visual question answering on recent public datasets through a BERT-based multi-modal representation, coupled with an LSTM decoder. We have implemented four representative models, including MMBERT [14], CMSA [8], PTUnifier [3] and METER [5].

Recently, motivated by the achievements of ChatGPT [27] and GPT-4 [1], alongside the efficacious deployment of open-source, instruction-tuned large language models (LLMs) within the general domain, a myriad of biomedical-oriented LLM chatbots have emerged. Notable among these are ChatDoctor [31], Med-Alpaca [10], PMC-LLaMA [25], DoctorGLM [28], and Huatuo [24]. LLMs are trained on large amounts of textual data that can help interpret complex and detailed information in medical images. Our model library also provides two recent models for generating the linguistic representation of the question in Med-VQA: MiniGPT-4 [34] has multi-modal abilities by properly aligning visual features with advanced LLMs, and LLaVA-Med [16] performs multi-modal instruction-tuning by leveraging large-scale biomedical data.

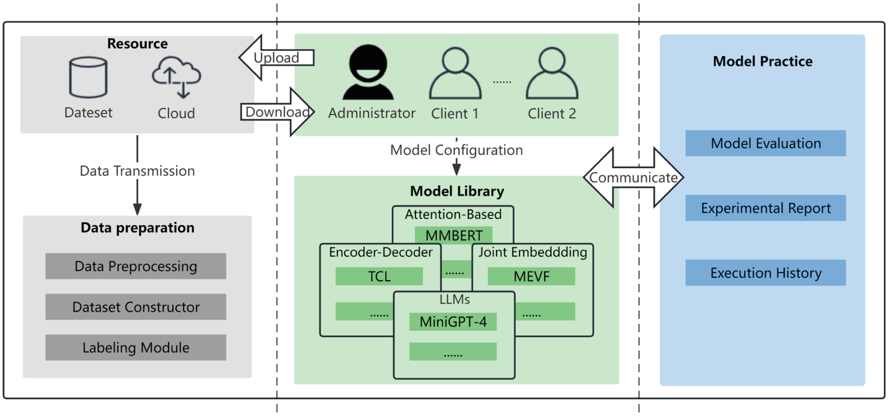


Fig. 4. System architecture of our BESTMVQA

4 System Overview

In Fig. 4, our BESTMVQA system has three components: data preparation, model library, and model practice. The data preparation component is developed based on a semi-automatic data generation tool. Users first upload self-collected clinical data. Then, medical images and relevant texts are extracted for medical concept discovery. We provide a human-in-the-loop framework to analyze and annotate medical concepts. To facilitate the effort, we first auto-label the medical concepts by employing the BioLinkBERT-BiLSTM-CRF [30]. Then, professionals can conveniently verify the medical concepts. After that, medical images, medical concepts and diagnosis texts are fed into a pre-trained language model for generating high-quality QA pairs. We employ a large-scale medical multi-modal corpus to pre-train and fine-tune an effective model, which can be easily incorporated into existing neural models for generating medical VQA pairs. our system provides a model library, to avoid duplication of efforts on implementing SOTAs for experimental evaluation. A wide spectrum of SOTAs have been implemented. The detailed statistics of the models can be seen in Sect. 3. Based our library, users can conveniently select a benchmark dataset and any number of SOTAs from our model library. Then, our system automatically performs extensive experiments to evaluate SOTAs over the benchmark dataset, and presents the final report to the user. From our report, the user can comprehensively study SOTAs and their applicability to Med-VQA. Users can also download the experimental reports and the source codes for further practice.

5 Empirical Study

Users can use our BESTMVQA system to systematically evaluate SOTAs on benchmark datasets for Med-VQA. To comprehensively evaluate the effectiveness of the models, we employ the metric of *accuracy* for open-ended, closed-ended, and overall questions. Five datasets are provided for users for model practice to investigate the applicability of models to diverse application scenarios.

Table 2. The statistics of datasets. NI, NQ and NA represent the number of images, questions and answers, respectively. MeanQL and MeanAL represent the length of questions and answers, respectively.

Dataset	NI	NQ	MeanQL	MeanAL	NA
VQA-RAD [15]	314	3515	6.49	1.61	557
MedVQA-2019 [2]	4200	15292	6.88	2.12	1749
SLAKE-EN [17]	642	7033	8.03	1.4	234
PathVQA [11]	4289	32795	6.33	1.79	4946
OVQA [12]	2000	19020	8.73	3.32	1065

5.1 Considered Models

We emphasize the utilization of "out-of-the-box" models, defining a model as "usable out of the box" if it meets the following criteria: (*i*) publicly available executable source code, (*ii*) well-defined default hyperparameters, (*iii*) no mandatory hyperparameter optimization, and (*iv*) absence of requirements for language model retraining and vocabulary adaptation. To ensure consistent evaluation and practical applicability, all models are expected to generate predictions in a standard format. Adhering to the criteria is essential for models that can help guarantee aligning with the concept of "out of the box".

Models are identified and classified as shown in Table 1, containing (*i*) those specifically tailored for Med-VQA, and (*ii*) the application of general VQA models to the medical domain.

5.2 Experimental Setup

Datasets. All models are evaluated using the following five datasets:

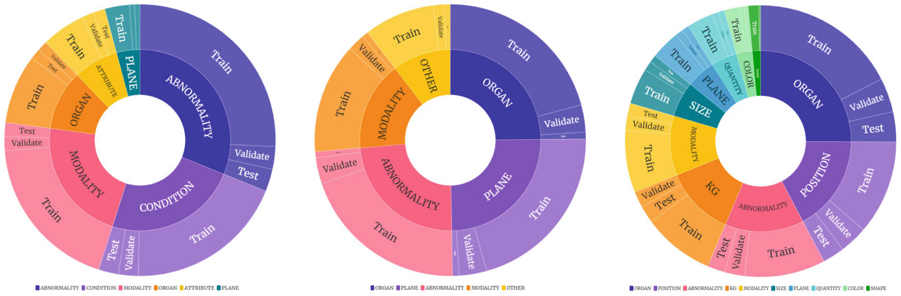
OVQA [12] has 2,001 images and 19,020 QA pairs, with each image linked to multiple QA pairs.

VQA-RAD [15] includes 314 images and 3,515 questions answered by clinical doctors, with 10 question types across the head, chest and abdomen.

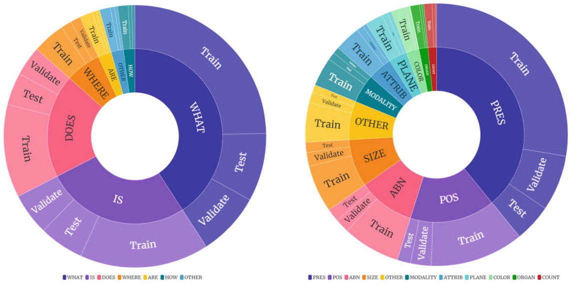
SLAKE [17] is a bilingual dataset annotated by experienced doctors, which is represented as SLAKE-EN in English.

Table 3. Default values for Batch Size, Learning Rate, and Epoch for each model

Baseline	Batch Size	Learning Rate	Epoch
MEVF+SAN	16	1.00E-03	20
MEVF+BAN	8	1.00E-03	20
CR	64	1.00E-03	40
MMQ	64	1.00E-03	60
VQAMix+SAN	8	1.00E-03	80
VQAMix+BAN	8	1.00E-03	80
CMSA	32	1.00E-03	60
MMBERT	16	1.00E-03	80
PTUnifier	8	1.00E-05	50
METER	32	1.00E-05	25
TCL	4	2.00E-05	20



(a) OVQA (b) MedVQA-2019 (c) SLAKE-EN



(d) PathVQA (e) VQA-RAD

Fig. 5. Distribution of question types per dataset

MedVQA-2019 [2] is a radiology dataset from the ImageClef challenge, which includes 642 images with over 7,000 QA pairs.

PathVQA [11] consists of 32,795 pairs generated from pathological images.

Datasets were chosen for their diversity in sample sizes (Table 2). For VQA-RAD and SLAKE, we have reorganized the datasets in a 70%-15%-15% ratio due to the lack of validation sets. As for the other datasets, We use the proportion of the corresponding data splits. The detailed statistics for data splits are shown in Table 4. The distribution of question types is illustrated in Fig. 5.

Table 4. The statistics of data splits. NI represents the number of images. MaxQL, MinQL and MeanQL represent the max, min and mean length of questions, respectively; NCF and NOF represent the number of close-ended and open-ended questions, respectively. MedVQA-2019 is not divided into open-ended and closed-ended questions.

Dataset	Sample	NI	MaxQL	MinQL	MeanQL	Vocabulary	NCF	NOF
VQA-RAD (train)	2451	314	21	3	6.43	1114	1443	1008
VQA-RAD (valid)	613	258	19	3	6.42	625	380	233
VQA-RAD (test)	451	203	22	3	6.89	538	272	179
Total	3515	314	22	3	6.49	1288	2095	1420
MedVQA-2019 (train)	12792	3200	11	4	6.88	98	-	-
MedVQA-2019 (valid)	2000	500	11	4	6.86	94	-	-
MedVQA-2019 (test)	500	500	11	4	6.86	93	-	-
Total	15292	4200	11	4	6.88	98	-	-
SLAKE-EN (train)	4777	546	21	4	7.98	301	1905	2872
SLAKE-EN (valid)	1195	484	18	4	8.12	265	460	735
SLAKE-EN (test)	1061	96	21	4	8.11	265	416	645
Total	7033	642	21	4	8.03	306	2781	4252
PathVQA (train)	19755	2599	37	2	6.35	4161	9868	9887
PathVQA (valid)	6279	832	37	2	6.24	2537	3156	3123
PathVQA (test)	6761	858	42	2	6.33	2608	3409	3352
Total	32795	4289	42	2	6.33	5095	16433	16362
OVQA (train)	15216	2000	95	4	8.63	958	8037	7179
OVQA (valid)	1902	1235	62	4	9.04	613	830	1072
OVQA (test)	1902	1234	67	4	9.26	533	832	1070
Total	19020	2000	95	4	8.73	1005	9699	9321

Implementation Details. For pre-training, we use a large-scale publicly available dataset called by ROCO [20]. It contains image-text pairs collected from PubMed (https://pubmed.ncbi.nlm.nih.gov/). We selected 87,952 non composite radiographic images with relevant captions. For fine-tuning, we follow the training, validation, and testing data splits according to Table 4. Five benchmark Med-VQA datasets were used to train and evaluate SOTAs. Questions are divided into closed-ended and open-ended. Closed-ended questions are usually answered with “yes/no” or other limited options. Open-ended questions have

Table 5. Experimental results for discriminative models on the test set of VQA-RAD, SLAKE-EN, PathVQA, and OVQA datasets, including the *Accuracy* (ACC) of three indicators: Closed-ended, Open-ended, and Overall.

Dataset	Baseline	Closed-ended (ACC)	Open-ended (ACC)	Overall (ACC)
VQA-RAD	MEVF+SAN	75.4	40.2	61.4
	MEVF+BAN	78.3	52.5	68.1
	CR	77.2	57.6	69.4
	MMQ	75.7	56.9	68.2
	VQAMix+SAN	79.4	57	70.5
	VQAMix+BAN	80.9	57.5	71.6
	CMSA	78.5	63.7	72.5
	MMBERT	74.3	46.9	63.4
	PTUnifier	**86.4**	**68.2**	**79.2**
	METER	78.3	57	69.8
	TCL	73.5	56.4	66.7
SLAKE-EN	MEVF+SAN	78.4	75.3	76.5
	MEVF+BAN	81	75.7	77.8
	CR	76.9	78.4	77.5
	MMQ	78.4	76.7	77.4
	VQAMix+SAN	77.9	77.7	77.8
	VQAMix+BAN	83.2	78.1	80.1
	CMSA	68.3	49.1	56.6
	MMBERT	43.3	1.9	18.1
	PTUnifier	**89.4**	**81.6**	**84.6**
	METER	87.3	79.2	82.4
	TCL	87.5	78.4	82
PathVQA	MEVF+SAN	83.4	13.1	48.5
	MEVF+BAN	83.8	16.4	50.3
	CR	84.9	15.9	50.5
	MMQ	83.2	14.3	48.9
	VQAMix+SAN	83.9	9.6	46.9
	VQAMix+BAN	84.3	12.7	48.6
	CMSA	83.7	16.1	50.2
	MMBERT	83.2	13	48.1
	PTUnifier	85.5	10.1	48.1
	METER	**89.9**	29.8	60
	TCL	88.1	**36.9**	**62.7**
OVQA	MEVF+SAN	74.2	52.3	61.9
	MEVF+BAN	76.6	50.5	61.9
	CR	76.6	36.9	54.3
	MMQ	79	53.2	64.5
	VQAMix+SAN	77.6	59.1	67.2
	VQAMix+BAN	79.3	57	66.8
	CMSA	79.7	45.6	60.5
	MMBERT	80.5	48.7	62.6
	PTUnifier	**84.9**	**60.5**	**71.3**
	METER	82.1	51.7	65.1
	TCL	82.6	60.4	70.1

no restrictive structure and can have multiple correct answers. All models are trained on dual graphics NVIDIA RTX V100 GPU. We use the AdamW opti-

Table 6. Experimental results for discriminative models on the test set of MedVQA-2019. Due to the fact that the MedVQA-2019 is not strictly divided into open-ended and closed-ended question types, the table only contains the values of Overall *Accuracy*

Dataset	Baseline	Overall(ACC)
MedVQA-2019	MEVF+SAN	50
	MEVF+BAN	47.4
	CR	46.8
	MMQ	50
	VQAMix+SAN	47.2
	VQAMix+BAN	49
	CMSA	47.4
	MMBERT	51.2
	PTUnifier	60.3
	METER	**73.9**
	TCL	63

Table 7. Experimental results for generative models on the test set of VQA-RAD, SLAKE-EN, PathVQA, OVQA and MedVQA-2019 datasets, including the *Accuracy* (ACC) of Closed-ended and the *Recall*, *METEOR* of Open-ended.

Dataset	Baseline	Closed-ended (ACC)	Open-ended	
			Recall	*METEOR*
VQA-RAD	MiniGPT-4	56.2	32.2	0.043
	LLaVA-Med	58.8	32.1	0.238
SLAKE-EN	MiniGPT-4	53.2	36.8	0.038
	LLaVA-Med	53.6	40.7	0.308
PathVQA	MiniGPT-4	53.4	12	0.018
	LLaVA-Med	57.9	11.8	0.026
OVQA	MiniGPT-4	53.1	33.4	0.066
	LLaVA-Med	66.8	39.1	0.237
MedVQA-2019	MiniGPT-4	-	23.2	0.019
	LLaVA-Med	-	25	0.055

mizer with the same preheating steps. See Table 3 for detailed parameter settings of models.

5.3 Evaluation Metrics

To quantitatively measure the performance of models, we use the *accuracy* as an evaluation metric, and compute it for closed-ended and open-ended questions for discriminative models, as they can be defined as a classification task. Let P_i

and L_i respectively denote the prediction and ground-truth label of sample i in the test set, and T represents the test set. The *accuracy* is calculated as follows:

$$accuracy = \frac{1}{|T|} \sum_{i \in T} l(P_i = L_i) \tag{1}$$

where l equals 1 only if $P_i = L_i$, otherwise 0.

For generative models such as MiniGPT-4 and LLaVA-Med, we report the *accuracy* for closed-ended questions as we leverage prompts to guide the model in answering these questions under a specified candidate set. For open-ended questions, we adopt *recall* to evaluate the ratio that ground-truth tokens appear in the generated sequences and $METEOR$ to assess the word order consistency between generated answer and ground-truth. The *recall* can be formalized as:

$$recall = \frac{TP}{TP + FN} \tag{2}$$

where TP is the number of ground-truth tokens that correctly predicted and FN stands for the number of ground-truth tokens that didn't appear in the predicted answer.

5.4 Results

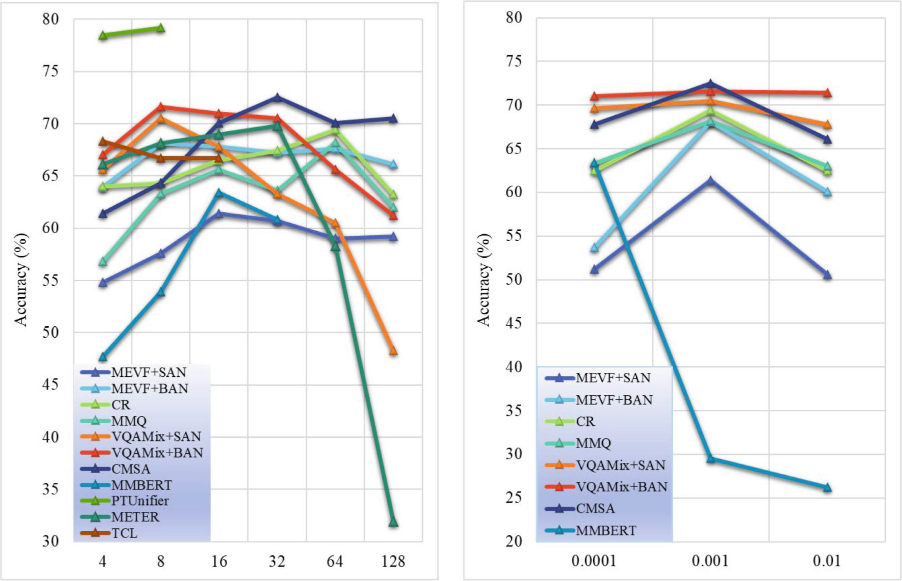
Tables 5, 7 and 6 show the *accuracy* achieved by all the considered models.

(*i*) In closed-ended questions, discriminative models (Table 5), are more applicable to Med-VQA, compared with LLMs (Table 7). This is because the generative models focus on simulating and generating data that requires broader language understanding and visual information processing capabilities. For simple closed-ended questions, they may suffer from the over-generation problem.

(*ii*) Among discriminative models, the PTUnifier which is pre-trained in the medical domain performs the best on VQA-RAD, SLAKE-EN and OVQA, but not so well on PathVQA and MedVQA-2019. As for the pre-trained models in general domain, TCL and METER achieve better performance on PathVQA and MedVQA-2019. The possible reason is that PathVQA is collected from a wide range of sources, including textbooks and literature, while MedVQA-2019 is artificially generated and cannot represent formal clinical data. PTUnifier adopts a visual language pre-training framework and unifies the fused encoder and dual encoder, thereby excelling on multi-modal tasks.

(*iii*) For generative models, MiniGPT-4 performs worst in terms of both the accuracy and the word order of generating answer on every dataset. Although utilizing massive amounts of data for training, it is still unable to effectively mine the domain-specific knowledge to answer a medical question, then over-generate lots of irrelevant text, and finally resulting in poor performance. In addition, the usage of inappropriate prompts may further degrade the model performance.

(*iv*) The performance of lightweight models such as MEVF, CR, MMQ, and CMSA is significantly inferior to complex models like PTUnifier, TCL, and METER. This is because models like PTUnifier have more parameters and adopt a deeper neural network structure, which is beneficial for learning the alignment between images and texts.



(a) Batch Size (b) Learning Rate

Fig. 6. Model performance varies with batch size and learning rate

5.5 Detailed Analysis

Figure 6 shows that the values of hyperparameters are determined based on the values set with the best performance on the validation dataset. The results of each model are obtained by changing the Batch Size (BZ) and Learning Rate (LR). Due to limited computing power, we only show parts of the results: (*i*) The results of MiniGPT-4 and LLaVA-Med are eliminated as they cannot be fine-tuned; (*ii*) We show part of results for PTUnifier in Fig. 6(a), as it requires more computing power for larger values of BZ; (*iii*) Similarly, we show part of results for PTUnifier, TCL, and METER with larger number of parameters in Fig. 6(b), as the value range of LR is not comparable to that of other models.

In Fig. 6(a), the performance of each model gradually increases when the BZ values increase, and then decrease after reaching a saddle point, due to the gradient calculation. However, when BZ is set to a large value, some models converge to local stationary points, such as METER and VQAMix-SAN. In Fig. 6(b), (*i*) with the increase of LR values, the performance of MMBERT shows a significant decline, and (*ii*) the performance of MEVF, CR, and CMSA first increase and then decrease with the increase of LR values.

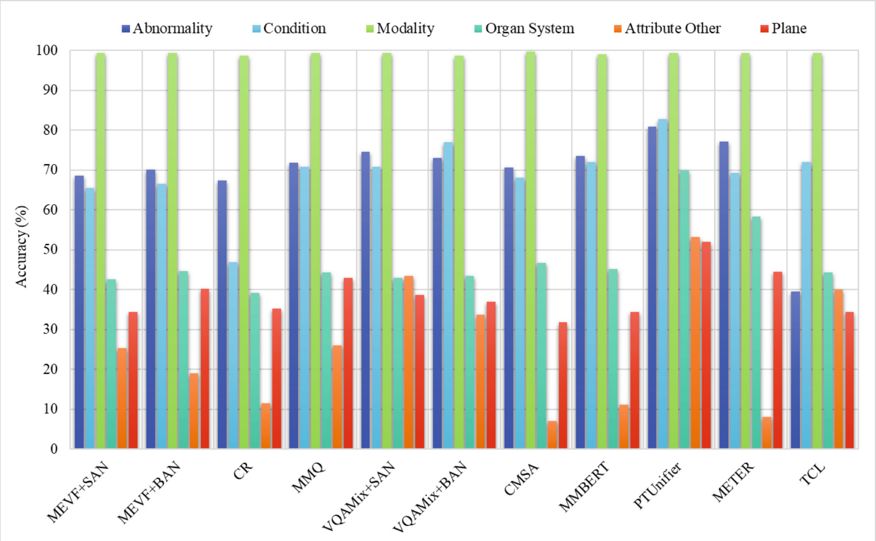
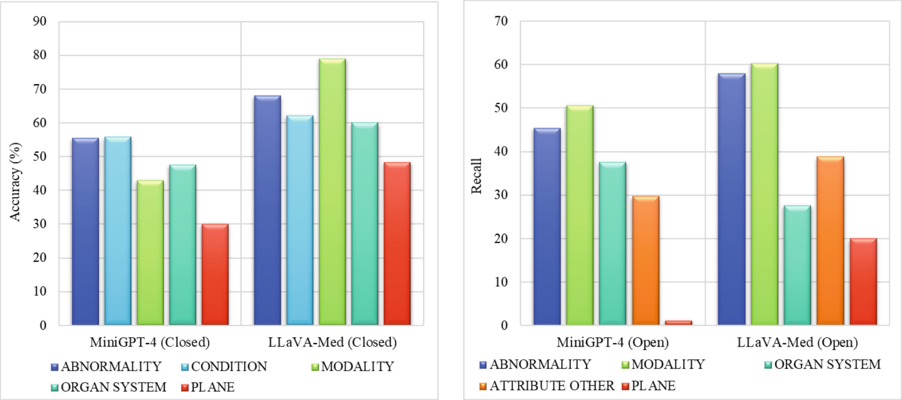


Fig. 7. The *Accuracy* of different question types for discriminative models in OVQA



(a) Closed-ended Question (b) Open-ended Question

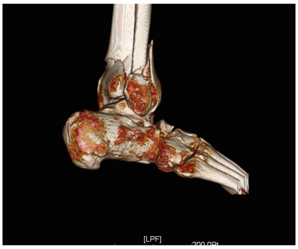
Fig. 8. The performance of different question types for LLMs in OVQA

Figures 7 and 8 show the results on various question types for discriminative and generative models over the OVQA dataset, respectively. In Fig. 7, we can derive that:(*i*) All discriminative models perform well on the *Modality* type of questions because MRI or CT image features are obvious, enabling the image encoder to effectively extract image features. (*ii*) All models have unsatisfactory performance on the *Attribute Other* type of questions, as descriptive questions are not suitable for label classification tasks. (*iii*) PTUnifier and VQAMix perform well on most types of questions. PTUnifier introduces visual and textual prompts for feature representation and improves the diversity of the prompts by constructing prompt pools, which enable different types of questions to select the appropriate prompts and enhance the image-text alignment in the fusion encoder. VQAMix incorporates a conditional label combination strategy for data augmentation, allowing for extracting more comprehensive image features.

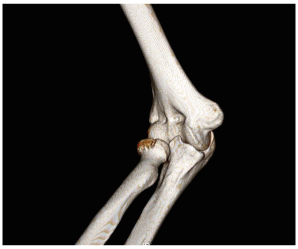
In Fig. 8, LLaVA-Med performs better than MiniGPT-4 on almost all types of questions, as it contains extensive domain-specific knowledge by pre-training and instruction tuning based on a large-scale biomedical dataset. Especially, LLaVA-Med greatly outperforms MiniGPT-4 on the *Plane* type of open-ended questions, as these specialized questions require models to fully capture the medical image features and exert domain knowledge to generate answers.

5.6 Qualitative Analysis

We provide a qualitative comparison of all models. Two examples from the OVQA dataset in Fig. 9 show that early discriminative models such as MEVF, CR, MMBERT, CMSA, and VQAMix, fail to answer Med-VQA questions, compared to the latest discriminative models such as TCL, METER, and PTUnifier. In Fig. 9, the Red Cross indicates that the prediction is wrong, and the green check indicates that the prediction is correct. The given question is to consult the abnormal position of orthopedic images. We observed that traditional models such as MEVF predict wrong abnormal positions. While TCL, and other



Q1:Is this image abnormal?
A1:Yes.
MiniGPT-4:It is difficult to determine whether the image is abnormal without more context. It appears to be a medical image, but the condition of the patient's foot is unclear. The image does not show any visible injuries or deformities. ✗
LLaVA-Med: No, the image is not abnormal. It shows a normal CT image of the tibial malleolus. ✗
MEVF+SAN:No. ✗
MEVF+BAN:No. ✗
CR:No. ✗
MMQ:No. ✗
VQAMix+SAN:No. ✗
VQAMix+BAN:No. ✗
CMSA:No. ✗
MMBERT:No. ✗
PTUnifier:Yes. ✓
METER:Yes. ✓
TCL:Yes. ✓



Q2:Are there signs of Comminuted fracture of the left distal radius, Avulsion fracture of the left ulnar styloid process, Fracture of the right radius head, or there's no abnormality?
A2:Comminuted fracture of left distal radius; Avulsion fracture of left ulnar styloid process; Fracture of right radial head.
MiniGPT-4:The distal radius is the lower and larger of the two bones that make up the forearm. It is located at the end of the upper arm bone (humerus) and forms the wrist joint with the carpals (the two bones in the lower arm). The distal radius is a curved bone that forms part of the elbow joint and helps to stabilize the joint during movement. It is also involved in wrist movements such as flexion and extension. ✗
LLaVA-Med:The CT image does not show any signs of a comminuted fracture of the left distal radius, avulsion fracture of the left ulnar styloid process, or fracture of the right radius head. This suggests that the bones in the left forearm appear to be intact and without any significant fractures or damage. ✗
MEVF+SAN:comminuted fracture of left radius head, avulsion fracture of left ulna coronoid process.. ✗
MEVF+BAN:comminuted fracture of the left proximal humerus, avulsion fracture of the left lateral humerus, comminuted fracture of the left proximal ulna, and left distal radius fracture. ✗
CR:fracture of the fifth metacarpal bone of the right hand, with the broken end shifted at an angle. ✗
MMQ:comminuted fracture of left olecranon, fracture of left coronal process, dislocation of left radial head. ✗
VQAMix+SAN:comminuted fracture of left distal radius; avulsion fracture of left ulnar styloid process. ✗
VQAMix+BAN:comminuted fracture of left distal radius, left ulnar styloid fracture. ✗
CMSA:comminuted fracture of left distal radius; avulsion fracture of left ulnar styloid process. ✗
MMBERT:comminuted fracture of right distal radius, avulsion fracture of right ulnar styloid process. ✗
PTUnifier/METER/TCL:Comminuted fracture of left distal radius; Avulsion fracture of left ulnar styloid process; Fracture of right radial head. ✓

Fig. 9. Two testing examples selected from OVQA

advanced models can locate the abnormality to the correct position. This also indicates that the advanced VQA deep learning models with large parameters can not only correctly understand the image content, but also capture the region of interest related to the question, leading to predicting the correct answer.

6 Conclusion

Deep learning models for Med-VQA face unique challenges, necessitating urgent comprehensive empirical studies on SOTAs to advance techniques and medical practice. To address this, we implemented a benchmark evaluation system that compares user-selected models and reports detailed experimental results. Additionally, users can download datasets, reports, and source codes for further exploration. Our system provides a unified platform to facilitate diverse medical practices.

Acknowledgement. This work was done when Xiaojie Hong worked for the project in Meetyou AI Lab. Xiaoli Wang was supported by the Natural Science Foundation of Fujian Province of China (No. 2021J01003).

References

1. Achiam, J., et al.: GPT-4 technical report. arXiv preprint arXiv:2303.08774 (2023)
2. Ben Abacha, A., Hasan, S.A., Datla, V.V., Demner-Fushman, D., Müller, H.: VQA-med: Overview of the medical visual question answering task at imageclef 2019. In: CLEF, 9–12 September 2019 (2019)
3. Chen, Z., Diao, S., Wang, B., Li, G., Wan, X.: Towards unifying medical vision-and-language pre-training via soft prompts. arXiv preprint arXiv:2302.08958 (2023)
4. Do, T., Nguyen, B.X., Tjiputra, E., Tran, M., Tran, Q.D., Nguyen, A.: Multiple meta-model quantifying for medical visual question answering. In: de Bruijne, M., et al. (eds.) MICCAI 2021. LNCS, vol. 12905, pp. 64–74. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-87240-3_7
5. Dou, Z.Y., et al.: An empirical study of training end-to-end vision-and-language transformers. In: CVPR, pp. 18166–18176 (2022)
6. Eslami, S., de Melo, G., Meinel, C.: Does clip benefit visual question answering in the medical domain as much as it does in the general domain? arXiv preprint arXiv:2112.13906 (2021)
7. Finn, C., Abbeel, P., Levine, S.: Model-agnostic meta-learning for fast adaptation of deep networks. In: ICML, pp. 1126–1135. PMLR (2017)
8. Gong, H., Chen, G., Liu, S., Yu, Y., Li, G.: Cross-modal self-attention with multi-task pre-training for medical visual question answering. In: ACM ICMR, pp. 456–460 (2021)
9. Gong, H., Chen, G., Mao, M., Li, Z., Li, G.: VQAMix: conditional triplet mixup for medical visual question answering. IEEE Trans. Med. Imaging **41**(11), 3332–3343 (2022)
10. Han, T., et al.: Medalpaca–an open-source collection of medical conversational AI models and training data. arXiv preprint arXiv:2304.08247 (2023)
11. He, X., et al.: Pathological visual question answering. arXiv preprint arXiv:2010.12435 (2020)

12. Huang, Y., Wang, X., Liu, F., Huang, G.: OVQA: a clinically generated visual question answering dataset. In: ACM SIGIR, pp. 2924–2938 (2022)
13. Huang, Y., Wang, X., Su, J.: An effective pre-trained visual encoder for medical visual question answering. In: Yang, X., et al. (eds.) ADMA. LNCS, vol. 14180, pp. 466–481. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-46677-9_32
14. Khare, Y., Bagal, V., Mathew, M., Devi, A., Priyakumar, U.D., Jawahar, C.: Mmbert: Multimodal BERT pretraining for improved medical VQA. In: ISBI, pp. 1033–1036. IEEE (2021)
15. Lau, J.J., Gayen, S., Ben Abacha, A., Demner-Fushman, D.: A dataset of clinically generated visual questions and answers about radiology images. Sci. Data **5**(1), 1–10 (2018)
16. Li, C., et al.: LLaVA-med: training a large language-and-vision assistant for biomedicine in one day. arXiv preprint arXiv:2306.00890 (2023)
17. Liu, B., Zhan, L.M., Xu, L., Ma, L., Yang, Y., Wu, X.M.: Slake: a semantically-labeled knowledge-enhanced dataset for medical visual question answering. In: ISBI, pp. 1650–1654. IEEE (2021)
18. Masci, J., Meier, U., Cireşan, D., Schmidhuber, J.: Stacked convolutional auto-encoders for hierarchical feature extraction. In: Honkela, T., Duch, W., Girolami, M., Kaski, S. (eds.) ICANN 2011. LNCS, vol. 6791, pp. 52–59. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-21735-7_7
19. Nguyen, B.D., Do, T.-T., Nguyen, B.X., Do, T., Tjiputra, E., Tran, Q.D.: Overcoming data limitation in medical visual question answering. In: Shen, D., et al. (eds.) MICCAI 2019. LNCS, vol. 11767, pp. 522–530. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-32251-9_57
20. Pelka, O., Koitka, S., Rückert, J., Nensa, F., Friedrich, C.M.: Radiology objects in COntext (ROCO): a multimodal image dataset. In: Stoyanov, D., et al. (eds.) LABELS/CVII/STENT -2018. LNCS, vol. 11043, pp. 180–189. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01364-6_20
21. Sarrouti, M.: NLM at VQA-med 2020: visual question answering and generation in the medical domain (2020)
22. Sharma, D., Purushotham, S., Reddy, C.K.: Medfusenet: an attention-based multimodal deep learning model for visual question answering in the medical domain. Sci. Rep. **11**(1), 19826 (2021)
23. Srivastava, Y., Murali, V., Dubey, S.R., Mukherjee, S.: Visual Question Answering using Deep Learning: A Survey and Performance Analysis, pp. 75–86 (2021)
24. Wang, H., et al.: Huatuo: tuning llama model with Chinese medical knowledge. arXiv preprint arXiv:2304.06975 (2023)
25. Wu, C., Zhang, X., Zhang, Y., Wang, Y., Xie, W.: PMC-llama: further finetuning llama on medical papers. arXiv preprint arXiv:2304.14454 (2023)
26. Wu, Q., Teney, D., Wang, P., Shen, C., Dick, A., Hengel, A.: Visual question answering: a survey of methods and datasets. Cornell University - arXiv, Cornell University - arXiv (2016)
27. Wu, T., et al.: A brief overview of ChatGPT: the history, status quo and potential future development. IEEE/CAA J. Automatica Sinica **10**(5), 1122–1136 (2023)
28. Xiong, H., et al.: Doctorglm: fine-tuning your Chinese doctor is not a herculean task. arXiv preprint arXiv:2304.01097 (2023)
29. Yang, J., et al.: Vision-language pre-training with triple contrastive learning. In: CVPR, pp. 15671–15680 (2022)
30. Yasunaga, M., Leskovec, J., Liang, P.: LinkBERT: pretraining language models with document links. In: ACL, pp. 8003–8016 (2022)

31. Yunxiang, L., Zihan, L., Kai, Z., Ruilong, D., You, Z.: Chatdoctor: a medical chat model fine-tuned on llama model using medical domain knowledge. arXiv preprint arXiv:2303.14070 (2023)
32. Zhan, L.M., Liu, B., Fan, L., Chen, J., Wu, X.M.: Medical visual question answering via conditional reasoning. In: ACM MM, pp. 2345–2354 (2020)
33. Zhang, X., et al.: PMC-VQA: visual instruction tuning for medical visual question answering. arXiv preprint arXiv:2305.10415 (2023)
34. Zhu, D., Chen, J., Shen, X., Li, X., Elhoseiny, M.: Minigpt-4: enhancing vision-language understanding with advanced large language models. arXiv preprint arXiv:2304.10592 (2023)

AeroINR: Meta-learning for Efficient Generation of Aerodynamic Geometries

Tom Bamford(✉), David Toal, and Andy Keane

University of Southampton, Southampton SO16 7QF, UK
{j.t.bamford,d.j.j.toal,andy.keane}@soton.ac.uk

Abstract. Effective optimisation of aerodynamic shapes requires high-quality parameterisation of candidate geometries. In recent years, the increasing availability and applicability of data - through increasing computational power, GPUs, cloud storage and AI - has motivated the development of data-driven approaches to the parameterisation problem, particularly those that can process the image-based data coming from scanned design parts. In this paper a novel approach to aerodynamic shape parameterisation is proposed, which leverages meta-learning in a generative deep learning framework. The solution put forward - AeroINR - aims to learn continuous neural representations as surrogates of the discrete field data used for shape representation in image-based applications. This approach transforms the learning problem to that of the surrogate model weight distribution of candidate geometries, rather than grid-based field values directly, which can reduce the number of variables describing each geometry by an order of magnitude or more. Benchmarking is carried out against three state-of-the-art deep-learning based aerofoil parameterisations, with AeroINR shown to outperform these models in two of the three metrics considered. Ablation study results show the robustness of this approach to generative framework and choice of discrete field representation.

Keywords: AI-Aided Design · Variational Auto-Encoders (VAE) · Hypernetworks · Implicit Neural Representations (INR) · Meta-Learning

1 Introduction

The requirement for engineering parts to be aerodynamically efficient is a common theme across many engineering projects, within specialities as diverse as aerospace [24,29], automative [22,44], energy [10,25], civil [2,27] and beyond [21,42]. In the context of aircraft design, aerodynamic considerations must be incorporated throughout the process. For example, in the design of the body itself, wings and other lift-generating components are optimised to increase lift whilst minimising drag and associated shock fronts [35], whereas non-lifting parts must

---

Supported by Rolls-Royce plc.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024

A. Bifet et al. (Eds.): ECML PKDD 2024, LNAI 14949, pp. 452–467, 2024.
https://doi.org/10.1007/978-3-031-70378-2_28

satisfy drag constraints as well as ensuring the desired functionality of a given part is achieved [36]. With regards to engine design, the profiles of turbomachinary blades such as compressors, turbines and fans are optimised for power efficiency, aiming to reduce pressure losses in the flow as it cascades through the engine stages.

In three dimensions, optimisation of essential aerodynamic parts such as blades and wings is typically carried out on a geometric parameterisation defined by the stitching together of two-dimensional cross-sections - aerofoils - in the third dimension [45,46]. Efficient parameterisation of the aerofoils - allowing sufficient flexibility in a compact space - is therefore a particularly important part of the aerodynamic design process, and has been extensively studied over the years [20,23,30,41].

Traditional parameterisations, such as the NACA 4-digit [5] and Class-Shape Transformation (CST) [23], specify a pre-defined functional form of the geometry and allow some variation around this. More recently, data-driven approaches built on neural networks have been investigated as an alternative approach to aerofoil parameterisation [7,8,26]. Thus far, their primary use has been as a dimensionality reduction algorithm, in which neural networks are used to find a more compact representation of traditional aerofoil design spaces, such as CST and Bezier curves. Correspondingly, the capability of data-driven approaches in increasing design space size, whilst limiting the growth in dimensionality, are yet to be fully understood. This has primarily been due to the difficulty in ensuring sufficiently smooth output designs for aerodynamic applications, but such an approach has been implemented successfully in SDF-GAN [3].

However, whilst SDF-GAN has been shown to be effective, scalability is still an issue due to the direct grid-based learning utilised. Recent work has shown that a more compact form of geometry representation is provided by Implicit Neural Representations (INRs), in which neural networks are used to encode individual geometries [11,31,32]. In this work INRs are utilised alongside a hypernetwork [19] to learn an efficient parameterisation of aerodynamic geometries through meta-learning. The following contributions are made:

- Propose the use of functional representations for aerodynamic geometry representation to achieve a continuous, grid-free parameterisation of the data
- Utilise meta-learning to learn a prior over the space of functions, yielding a geometric parameterisation suitable for use in an optimisation procedure
- Demonstrate applicability to aerofoil design and compare to state-of-the-art parameterisations

2 Existing Approaches and Related Work

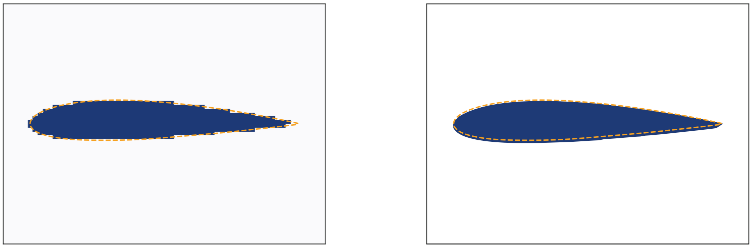
2.1 Traditional Parameterisations

Traditional approaches to aerofoil parameterisation - those that don't leverage deep learning frameworks - can be categorised into two distinct groups. Deformative approaches require a baseline geometry, with parameter variations modifying the shape relative to this baseline. This approach can often be useful since a designer will typically know the main properties of the geometry required, with only small scale variations necessary during the optimisation process. Parameterisations that fall into this group include Hicks-Henne and Free-Form Deformation (FFD) [20,37]. The alternative approach is to describe geometry explicitly, without reference to a pre-chosen design. This is known as a constructive parameterisation, and includes the well known NACA method, as well as more modern CST, PARSEC and orthogonal decomposition approaches [1,23,41,43]. In addition, shape descriptors built on piecewise polynomial curves such as B-splines and Bezier curves also fall into this category.

2.2 Deep Learning-Based Approaches

With regards to deep learning-based parameterisation approaches, one of the most successful is Bezier-GAN [7], which utilises an InfoGAN [9] framework to create a two-tier design space. In this setup, higher level latent variables are used to specify the class of shape required, and lower level noise variables vary the design within this class. Output designs are constrained to a Bezier curve to ensure smoothness of the geometry. An optimal transport variation has also been proposed, in which a Sinkhorn divergence-based loss is used in place of the discriminative network and aid stability in the training process; this is known as Entropic Bezier-GAN [6].

In an attempt to leverage the full benefits of a data-driven parameterisation, SDF-GAN has since been put forward which outputs designs directly through a grid-based SDF representation [3]. As such, the only constraints on the design space are the training data used and grid resolution, rather than an in-built mathematical form. In this case, the design space is specified by a single set of latent variables in an Adversarial Auto-Encoder [28] framework, which helps to drive the smoothness of the output geometries through the reconstruction loss term.



(a) Occupancy map

(b) SDF

Fig. 1. NACA 1410 reconstruction comparison for 64 × 64 grid resolution with occupancy map and SDF representations.

2.3 Grid-Based Geometry Representation

In contrast to pre-defined mathematical forms for geometry representation, grid-based representations - being essentially a direct visualisation - require very little user input, thus limiting designer bias in the resulting design space. In addition, the functional forms able to be represented are constrained only by the resolution of the grid.

Occupancy Maps. Occupancy maps define a binary field over the domain, with points inside the shape being assigned a value of 1, and all other points assigned 0. In practice, the mapped values are assigned to the cells of a discretised grid overlaying the domain. This is a memory intensive, brute force approach, due to the requirement for fine cells at the boundary and the uniform nature of the grid. Variations on this idea have been proposed to improve the efficiency, particularly via adaptive grids [14,15,17].

Signed Distance Fields (SDF). The binary values associated with occupancy maps give no information as to the distance of a given cell from the boundary, only direction. Signed Distance Fields remedy this by assigning to each cell a continuous value, signed to denote position relative to the boundary, with the magnitude denoting distance to that boundary at its closest point. The additional information is then used to interpolate the shape boundary during reconstruction, allowing sub-pixel resolution and ensuring a smoother surface. The simplicity and effectiveness of this representation has motivated its use for a variety of applications, including medical imaging [34], collision detection [18] and in engineering design [4,33].

2.4 Approaches for Learning Priors over Geometry Space

Given an appropriate geometry representation scheme, a data-driven parameterisation is constructed by leveraging generative deep learning to learn a prior over the space of geometries. In the case of Bezier-GAN and SDF-GAN, this is done by learning a prior over the representation space directly, in the form of grid values and Bezier parameters. The effectiveness of such an approach is however limited by its scalability. For Bezier-GAN, extending the output parameter space to more complex or three-dimensional geometries is challenging due to the lack of clear functional form that would be suited to such cases. On the other hand, whilst the representation utilised by SDF-GAN is naturally flexible enough to deal with a wider range of geometric datasets, practically such an approach is limited by the increased memory requirements as the number of grid cells grows.

An alternative approach is to utilise Implicit Neural Representations (INRs) for geometry encoding in a more compact space. This has the advantage of retaining the natural flexibility of image-based representations, whilst simultaneously ensuring scalability to more complex and three-dimensional geometries. INRs utilise overfitting to learn functional representations of geometries through a neural network encoding [11,31], thus converting a discrete sampling of a field (e.g. occupancy map or SDF) into a continuous function. Such a solution has two primary advantages: 1. the continuous nature of the learnt field ensures the approach is resolution-agnostic, allowing easy scalability, 2. the number of network parameters required to represent the geometry is significantly smaller than the number of cells in high resolution uniform grids of equivalent capability [11]. The problem of learning a prior over the space of geometries is thus converted into learning the distribution of neural network parameterisations. This type of framework has already been explored in a few works to date, including MetaSDF which uses the MAML algorithm [16] for generalisation and INR geometry instantiation via few-shot learning [39], and for scene representation via hypernetworks [40]. A more compact approach is provided by DeepSDF, which leverages an auto-decoding setup such that INR weights are shared across all samples, with instantiation specified instead by the input latent code [32].

3 Framework of AeroINR

In this work, a hypernetwork is used within a generative framework (AeroINR - Fig. 2) to learn a prior over the space of INR weight parameters. Meta-learning is utilised during the training process, such that the target network weights for the geometries in the dataset and generative network weights are learnt simultaneously. During inference, INR weights corresponding to the input latent code are output by the hypernetwork and reshaped for INR instantiation. Given a set of coordinates corresponding to a grid of the desired resolution, the INR is used to map these coordinates to SDF values (neural SDF), from which the geometry can be reconstructed.

As far as the authors are aware AeroINR provides novelty in both framework and application. In particular, whilst work has been carried out to learn a prior over neural SDF weights in both MetaSDF [39] and with Scene Representation Networks [40], in the former case the MAML algorithm was used for meta-learning as opposed to hypernetworks, whereas in the latter the application is not for generative learning. MAML-like algorithms were also used in [13], which argued that INRs should be considered as data (functa) rather than models. No previous work has been carried out on INR-based meta-learning for aerodynamic parameterisation.

3.1 Implementation

The implementation of the generative component of AeroINR follows that of SDF-GAN. As such the baseline network architectures are equivalent, with only the four hyper-parameter attributes found to be the most impactful in [3] re-optimised for the meta-learning application proposed here. The final implementation sets the learning rate to 0.001, uses a single decoder/hypernet layer, and sets the number of discriminator features to 32. Note that the decoder feature

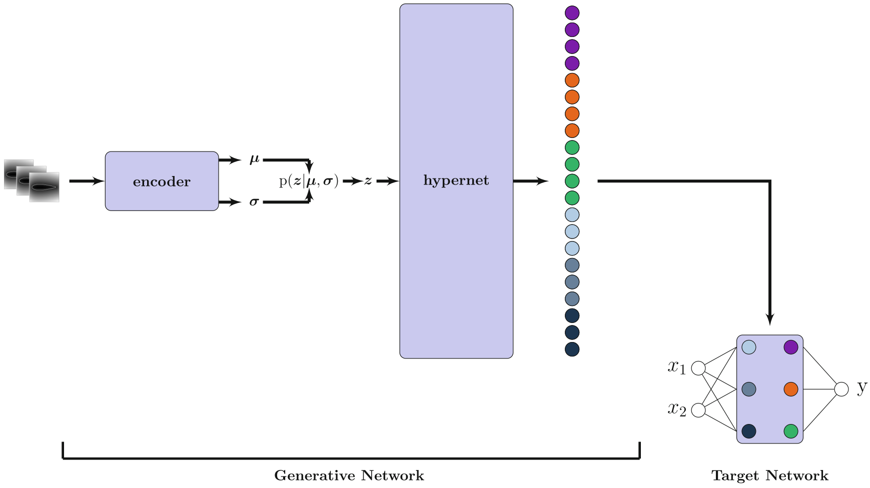


Fig. 2. Schematic of the generative architecture. The downstream target network weights are set to the output values of the hypernetwork, giving a unique neural representation for each sample. An additional discriminative network acts to smooth the structure of the latent space during training.

decrease rate is no longer applicable with only a single layer. For the INR architecture itself, the baseline network structure from SDF-GAN is also followed, giving 3 layers, 32 features, and a feature increase rate of 2.

The standard AAE loss function is used for optimisation during training, with the slight meta-learning modification that the reconstruction loss output is calculated from the downstream INR output after weight instantiation, rather than directly from the generative output. The final loss is given in Eq. 1, with the loss weighting γ set to 0.998 as for SDF-GAN.

$$\mathcal{L}_{\text{AAE}} = \mathcal{L}_{\text{rec}} + \gamma \cdot \mathcal{L}_{\text{adv}} \ , \tag{1}$$

The reconstruction term uses an MSE loss to calculate the difference between the raw aerofoil field data x_i and the INR reconstructed values $\hat{x}_i$,

$$\mathcal{L}_{\text{rec}} = \frac{1}{N}\sum_{i=1}^{N} \|x_i - \hat{x}_i\|^2 \ , \tag{2}$$

whilst the adversarial term uses the discriminator D to drive the latent coordinates output by the encoder Q to the smooth Gaussian distribution specified by the prior,

$$\mathcal{L}_{\text{adv}} = \min_{Q}\max_{D} \mathbf{E}_{z\sim p_z(z)}[\log D(z)] + \mathbf{E}_{x\sim p_{\text{data}}(x)}[\log(1 - D(Q(x)))] \ . \tag{3}$$

Training is carried out until convergence is reached, taken to be the case when the absolute change in all three loss components averaged over 1000 epochs is

Algorithm 1 AeroINR Meta-Learning Algorithm.

```
**Initialisation:**
Initialise encoder $q_{\theta^0}(z|x)$, hypernetwork $p_{\phi^0}(w|z)$, INR $f_w$; Convergence tolerance $\epsilon$; Batch size $B$; Learning rate $\beta$; Loss function weighting $\gamma$.
**Input:**
The grid data $\{(x^{(i)}, c^{(i)})\}_{i=1}^{N}$ representing aerofoil field values (SDF/Occupancy) and corresponding coordinates.
while not IsConverged($\epsilon$) do
    Sample a batch of aerofoils $\{(x^{(i)})\}_{i=1}^{B}$ from the dataset.
    for each aerofoil $j$ in the batch do
        Generate weights $w^{k,j} = p_\phi(z^j)$ corresponding to obtained representation $z^j = q_\theta(x^{(j)})$
        Instantiate INR weights and sample grid values: $\hat{x}^j = f_{w^{k,j}}(c^{(j)})$
        Compute loss: $L^j = L_{\text{AAE}}(x^{(j)}, \hat{x}^j, \gamma, \ldots)$
        Compute gradients: $\nabla L^j = \nabla_{f_w} L^j$
    Update hypernetwork: $p_\phi \leftarrow p_\phi - \beta \nabla_{p_\phi} \sum_j L^j$
    Increment epoch: $k \leftarrow k + 1$
```

less than 5% of their respective averages in the previous 1000 epochs. The full training procedure is given in Algorithm 1.

4 Experiments

4.1 Dataset

Dataset construction follows that in SDF-GAN [3]. Raw aerofoil coordinate data are used to construct the discrete field data for equivalent grid-based representation. Samples of 1500+ real aerofoils are first collected from the UIUC online database [38]. A 25-parameter CST [23] is fit to each aerofoil from which standardised sampling is carried out, giving 500 coordinates for each upper and lower aerofoil curve with a cosine sampling distribution. [30] showed such a fit to give an approximation accuracy to within a typical wind tunnel tolerance for over 99% of aerofoils within the UIUC dataset. This gives a total of 1231 real aerofoils in the dataset, after removal of those with unrealistic CST fits due to the small sample size of the original data.

Synthetic aerofoils are created to augment this dataset by sampling parameters of an SVD [43] of the UIUC set. The resultant weighted expansion allows sampling of coefficient values to create new designs. The mean and range of each coefficient is calculated across all UIUC aerofoils, a Gaussian distribution fit with the standard deviation set to a third of the range, and then random sampling of these distributions carried out. All synthetic aerofoils are evaluated by eye - ensuring a single smooth, continuous loop in each case - before being incorporated into the dataset. The dataset is split into train and validation sets, each containing 1000 samples. Only UIUC aerofoils are placed into the training set. Aerofoils are finally pre-processed into discrete grid representations of the relevant field (occupancy map/SDF). All grids are defined over a domain $x \in [-0.1, 1.1]$ and $y \in [-0.3, 0.3]$, which is large enough to contain all normalised aerofoils.

4.2 Benchmarks

AeroINR is compared as an aerofoil parameterisation tool against three state-of-the-art deep-learning-based competitors: Bezier-GAN, Entropic Bezier-GAN and SDF-GAN. Results for these models are taken from the original work.

Bezier-GAN [7]: GAN-based approach to parameterisation in which Bezier parameters are output by the generator to ensure smoothness. This is effectively a dimensionality reduction algorithm for the original Bezier curve parameterisation of aerofoils.

Entropic Bezier-GAN [6]: A variant of the Bezier-GAN approach. The training algorithm is modified such that the Sinkhorn divergence is utilised to minimise the difference between the generated and training distributions, rather than adversarial learning. This simplifies the architecture through removal of the discriminator and ensures a more stable learning process.

SDF-GAN [3]: Aerofoil parameterisation that outputs shapes directly as SDF grid representations, rather than a constrained mathematical form, via an AAE.

4.3 Metrics

For consistency with prior work, parameterisation capability is assessed through three metrics - MMD, relative diversity and latent space consistency - following [7]. MMD (Maximum Mean Discrepancy) is a well known measure of the difference between two distributions, used to quantify how well the learnt distribution matches that of the training data. Relative diversity measures the variability of the generated aerofoils relative to the training set, through comparison of the variance. Finally, latent space consistency quantifies the structure of the latent space through the correlation coefficient between distances of latent variables and distances of the corresponding aerofoil outputs. A more consistent latent space should aid the optimisation process.

To quantify the capabilities of the chosen AAE framework, an ablation study is carried out on a variety of model configurations. The unconstrained nature of the approach, alongside the auto-encoding functionality, means that measures of both design viability and reconstruction quality are required in this case. As such, three metrics are used to assess AeroINR ablation performance: feasibility, relative diversity and the L1 reconstruction error (evaluated on the validation set). Following [3], feasibility is assigned to those aerofoils for which the XFOIL aerodynamic solver [12] successfully finds convergent solutions, with a ratio of feasibility evaluated through a random sampling of 1000 aerofoils from the parameter space.

Note that aerodynamic evaluation is carried out both as a feasibility check on the output geometry, as well as for more precise lift-to-drag evaluations of aerofoils. The feasiblity check is carried out with a Mach number of 0.15 and Reynolds number of 6.0e+06. For lift-to-drag analysis, a smaller Mach number is used and geometric smoothing applied due to the increased sensitivity required for quantitative results with drag. In this case the Mach and Reynolds numbers are set to 0.01 and 1.8e+06 respectively, whilst two applications of smoothing via a Hanning filter are used before shape evaluation.

4.4 Experimental Setting

Experiments are carried out in parallel across a single, standard compute node on the IRIDIS HPC cluster at the University of Southampton, which consists of 40 CPUs, and 192 GB of available memory.

4.5 Results

Representation Capability of INRs. To assess the quality of aerofoil representation through INRs, a random subset of 100 UIUC aerofoils are used to train up equivalent neural field representations of the geometries. Since the aerofoils are now treated independently, as opposed to the generalised training associated with AeroINR, a slight modification to the neural architecture is required here. After some experimentation, an 8-layer INR is used with a constant value of 256 features in each layer. Each distinct INR is trained with a learning rate of 1e-04 and a batch size of 64. Transfer learning is implemented to speed up the training process, such that the first aerofoil is trained for 30 min, whilst all subsequent aerofoils in the dataset are trained for 10 min with INRs initialised to that of the previously trained aerofoil.

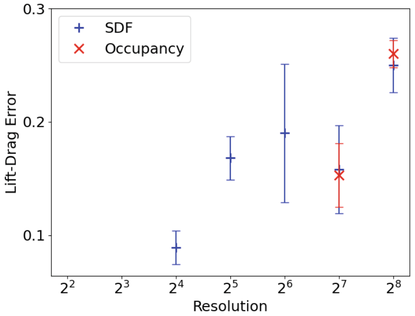


Fig. 3. Fractional lift-to-drag error for neural reconstructions across different sampling resolutions. Error bars correspond to standard deviations over three runs.

Figure 3 shows the average fractional lift-to-drag error of aerofoils reconstructed from their neural representation relative to the original training aerofoils. Results are shown for both type of field representation discussed (occupancy and SDF), as well are for varying training data resolution. Each configuration is run three times, for different weight initialisations and sampled aerofoils, to give the error bars shown. All reconstructions show aerodynamic performance within around 10–25% error relative to the original training aerofoils. As expected, the aerodynamic performance of the lower resolution reconstructions is closer to that of the original training data than that of higher resolutions, since overfitting is an easier problem with fewer data points. Such behaviour means a trade off between

Table 1. Metric comparison between AeroINR and alternative, state-of-the-art approaches.

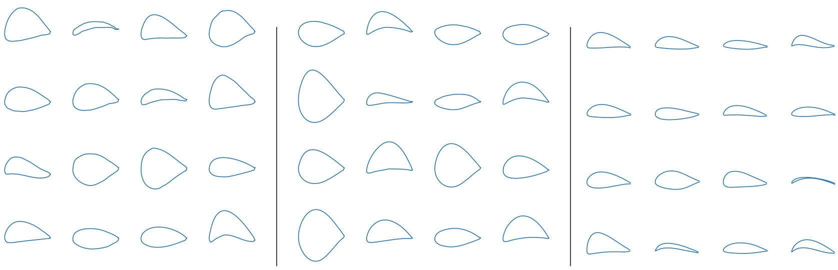
Model	MMD^2	Latent Space Consistency	Relative Diversity
Bezier-GAN	0.1856 ± 0.0007	0.9432 ± 0.0025	1.0455 ± 0.0144
Entropic Bezier-GAN	**0.0896 ± 0.0015**	0.9878 ± 0.0005	1.3264 ± 0.0196
SDF-GAN	0.3224 ± 0.0219	0.9640 ± 0.0030	2.5396 ± 0.2908
AeroINR	0.3494 ± 0.0151	**0.9996 ± 0.0001**	**2.6410 ± 0.1659**

grid and neural represention accuracy exists in this formulation. No significant difference is seen between the two representation fields for a given resolution.

AeroINR. The main results, both quantitative and qualitative, are presented in this section. Table 1 shows the performance of AeroINR relative to the state-of-the-art benchmark models for deep learning based aerofoil parameterisation described in Sect. 4.2. AeroINR outperforms all other approaches in two of the three metrics, with an inferior rank seen only for the MMD result. The relatively poor performance of both SDF-based approaches - AeroINR and SDF-GAN - in this metric is however somewhat unsurprising, given the output of the Bezier-GAN approaches is the constrained functional form of a Bezier curve. In addition, it is arguable as to whether superior performance in this metric is particularly desirable for a generative approach to parameterisation, in which the aim is not only to successfully parameterise shapes already known (i.e. the training data distribution), but also to generalise this such that designs from outside the original distribution may be synthesised.

Example aerofoils output with the proposed model are shown in both Figs. 4 and 5, which display the generative and reconstruction capabilities of AeroINR respectively. Figure 4 depicts the diversity of the learnt design space through random sampling of the prior distribution, and equivalent plots for both SDF-GAN and Bezier-GAN are shown for comparison. Figure 5 shows example reconstructions of aerofoils from the training set of UIUC aerofoils.

A visualisation of the learned latent space is shown in Fig. 6. Each individual sub-plot corresponds to a cross-sectional view of different two-dimensional subspaces, in which all other dimensions are fixed at the origin. Since there are sixteen dimensions in total, each sub-plot corresponds to a unique pair such that all dimensions are shown within the eight plots. The colour corresponds to the lift-to-drag ratio associated with each individual aerofoil; lighter colours represent higher values. The visualisation shows a clear structure within the latent space, with distinct regions of aerofoils showing similar aerodynamic performance. This is also true of the infeasible geometries within the latent space, which are depicted by areas of white space within the plots. Such a structure allows for intelligent searches through the space as required in an optimisation process. Note that the infeasible regions tend to occur towards the boundary of the space, as expected.



(a) AeroINR (b) SDF-GAN (c) Bezier-GAN

Fig. 4. Visualisation of diversity of different parameterisations through random sampling of the design space.

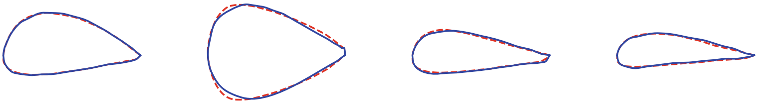


Fig. 5. Reconstructions of randomly selected aerofoils (blue, solid) with AeroINR. (Color figure online)

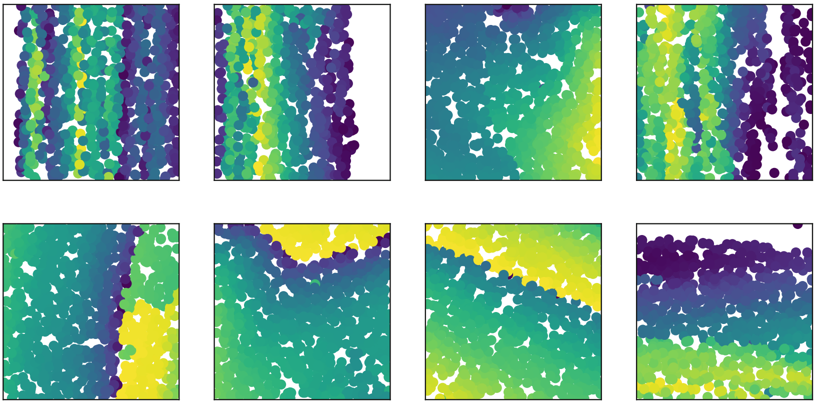


Fig. 6. Visualisation of the learned latent space through sampling of 2D cross-sections around the origin. Each point in space corresponds to a single aerofoil, colour-coded by its lift-to-drag ratio (light colours correspond to higher values). Note that white regions correspond to infeasible geometries.

Ablation Study. To establish the robustness of the proposed approach, an ablation study is carried out comparing AeroINR to a set of alternative configurations. Three components of the approach are varied: the chosen model framework, the encoding algorithm and the representation field. In particular, a Variational Auto-Encoder (VAE) - being a similar auto-encoding approach to generative modelling - is investigated as an alternative to the baseline AAE. Inspired by DeepSDF [32], the auto-decoding approach to learning shape embeddings is also implemented, which removes the encoder and instead treats latent variables as learnable parameters both during training and inference, with all other parameters kept fixed in the latter case. This has two advantages. Firstly, the number of parameters - and thus memory overhead - in the generative model is reduced; secondly, the raw data is no longer input into the network, which removes the constraints on resolution and shape that are enforced by an encoder.

The results are displayed in Table 2, which shows only the encoded embeddings for the occupancy representation produce impractically low feasibility fractions. In general it is also apparent that the reconstruction accuracy is significantly lower for the occupancy cases (as expected), whilst relative diversity can often be higher; given a less faithful reconstruction, a correspondingly higher diversity of aerofoils is not unsurprising. It is also worth stating that whilst training was stopped through a convergence test (typically after 2–3 h) for speed and convenience, performance may have been improved by periodically pausing training and re-evaluating against the desired functionality, as is done in [3].

Note that the encode time is for an entire batch of aerofoils, rather than individual geometries. The convergence tolerance is increased from 5 to 10 for the occupancy cases with the AAE framework, and when used alongside a direct encoding the learning rate is also reduced by a factor of 2, to stabilise training.

Table 2. Ablation study showing AeroINR performance (top row; bold) relative to a variety of alternative configurations. Displayed results are averaged across three runs. All times are in minutes.

Independent Variables			Dependent Variables				
Framework	Embedding	INR	Feas.	Rel. div	Recon. error	Train time	Encode time
AAE	Encoded	SDF	0.53	3.78	1.3e-02	220	~ 0
AAE	Encoded	Occ	0.18	5.55	2.6e-01	514	~ 0
AAE	Auto-dec.	SDF	0.58	0.71	6.0e-03	104	20
AAE	Auto-dec.	Occ	0.46	0.83	1.5e-01	110	45
VAE	Encoded	SDF	0.48	2.65	3.3e-03	130	~ 0
VAE	Encoded	Occ	0.32	9.71	2.1e-01	260	~ 0
VAE	Auto-dec.	SDF	0.45	0.73	4.7e-03	105	20
VAE	Auto-dec.	Occ	0.48	0.89	1.1e-01	110	45

Scalability. Figure 7 shows the growth in training parameter number with geometric resolution for the meta-learning approach of AeroINR relative to the grid-based learning of SDF-GAN. This plot assumes the networks remain fixed in

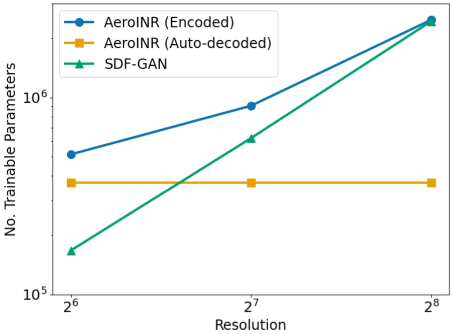


Fig. 7. Trainable parameter growth with resolution for the grid-based SDF-GAN and AeroINR.

size as resolution is increased, apart from those which process the input/output geometric grid directly. The benefit of AeroINR for scalable learning is apparent for both encoding approaches, but the auto-decoding setup in particular allows complete separation of network size from geometric resolution, highlighting the potential for both high-resolution and three-dimensional learning.

5 Conclusions and Future Work

This work presents a novel approach for aerodynamic shape parameterisation in which a prior distribution of candidate geometries is learnt through their implicit neural representations. The training algorithm utilises a meta-learning approach, leveraging hypernetworks to learn downstream neural representation weights and instantiating on-the-fly to allow a fully contained training process. Results show that this method is competitive with state-of-the-art deep-learning aerofoil parameterisations, outperforming competitors in both output diversity and consistency of the latent space. Whilst results are promising in the two-dimensional setting, the key benefit of this approach is in the geometric compression gained by learning a distribution over neural representation weights rather than directly over the image-based representations, allowing a significant reduction in the number of parameters required for accurate shape description - up to an order of magnitude or more. The increased scalability does however come with additional overhead; the auto-decoding implementation requires additional training during inference, whilst the alternative encoding approach increases the baseline number of parameters, and correspondingly time spent during optimisation of hyper-parameters. Future work will investigate the application of this method for three-dimensional design, where learning remains a significant challenge due to the large number of parameters required for sufficient geometric resolution.

Acknowledgements. The authors gratefully acknowledge funding support from Rolls-Royce PLC, as well as the University of Southampton for the use of the IRIDIS HPC facility.

References

1. Abbott, I., Von Doenhoff, A.: Theory of Wing Sections, Including a Summary of Airfoil Data. Dover Publications, New York (1959)
2. Al-masoodi, A.H.H., Alkhatib, F.H., Shafiq, N., Wahab, M.: The aerodynamic performance of tall buildings by utilizing aerodynamic modifications - a review study. IOP Conf. Ser. Earth Environ. Sci. **1022**(1), 012046 (2022). https://doi.org/10.1088/1755-1315/1022/1/012046
3. Bamford, T., Toal, D., Keane, A.: SDF-GAN: Aerofoil shape parameterisation via an adversarial auto-encoder. In: AIAA Aviation 2024 Forum (2024)
4. Bhatnagar, S., Afshar, Y., Pan, S., Duraisamy, K., Kaushik, S.: Prediction of aerodynamic flow fields using convolutional neural networks. Comput. Mech. **64**(2), 525–545 (2019). https://doi.org/10.1007/s00466-019-01740-0
5. Charles, D.: Nasa supercritical airfoils: a matrix of family-related airfoils. Technical Paper 2969, NASA (1990)
6. Chen, Q., Pope, P., Fuge, M.: Learning airfoil manifolds with optimal transport. In: AIAA SCITECH 2022 Forum (2022). https://doi.org/10.2514/6.2022-2352
7. Chen, W., Chiu, K., Fuge, M.D.: Airfoil design parameterization and optimization using bézier generative adversarial networks. AIAA J. **58**(11), 4723–4735 (2020). https://doi.org/10.2514/1.J059317
8. Chen, W., Ramamurthy, A.: Deep generative model for efficient 3D airfoil parameterization and generation. In: AIAA Scitech 2021 Forum (2021). https://doi.org/10.2514/6.2021-1690
9. Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., Abbeel, P.: Infogan: interpretable representation learning by information maximizing generative adversarial nets. In: Proceedings of the 30th International Conference on Neural Information Processing Systems, NIPS 2016, pp. 2180–2188. Curran Associates Inc., Red Hook (2016). https://doi.org/10.5555/3157096.3157340
10. Chiapperi, J., Greitzer, E., Tan, C.: Attributes of bi-directional turbomachinery for pumped thermal energy storage. J. Turbomach. **145**(3), 031007 (2023). https://doi.org/10.1115/1.4055647
11. Davies, T., Nowrouzezahrai, D., Jacobson, A.: Overfit neural networks as a compact shape representation. arXiv (2020)
12. Drela, M.: XFOIL: an analysis and design system for low reynolds number airfoils. In: Mueller, T.J. (ed.) Low Reynolds Number Aerodynamics, pp. 1–12. Springer, Heidelberg (1989). https://doi.org/10.1007/978-3-642-84010-4_1
13. Dupont, E., Kim, H., Eslami, S.M.A., Rezende, D.J., Rosenbaum, D.: From data to functa: your data point is a function and you can treat it like one. In: Proceedings of the 39th International Conference on Machine Learning, vol. 162, pp. 5694–5725. PMLR (2022)
14. Ebeida, M.S., Davis, R.L., Freund, R.W.: A new fast hybrid adaptive grid generation technique for arbitrary two-dimensional domains. Int. J. Numer. Meth. Eng. **84**(3), 305–329 (2010). https://doi.org/10.1002/nme.2900
15. Finkel, H.B., Bentley, J.L.: Quad trees: a data structure for retrieval on composite keys. Acta Informatica **4**(1), 1–9 (1974). https://doi.org/10.1007/BF00288916

16. Finn, C., Abbeel, P., Levine, S.: Model-agnostic meta-learning for fast adaptation of deep networks. In: Proceedings of the 34th International Conference on Machine Learning, vol. 70, pp. 1126–1135. JMLR (2017)
17. Frey, P., Marechal, L.: Fast adaptive quadtree mesh generation. In: Proceedings of the 7th International Meshing Roundtable, pp. 211–224 (2000)
18. Guendelman, E., Bridson, R., Fedkiw, R.: Nonconvex rigid bodies with stacking. ACM Trans. Graph. **22**(3), 871–878 (2003). https://doi.org/10.1145/882262.882358
19. Ha, D., Dai, A.M., Le, Q.V.: Hypernetworks. In: International Conference on Learning Representations (2017)
20. Hicks, R.M., Henne, P.A.: Wing design by numerical optimization. J. Aircr. **15**(7), 407–412 (1978). https://doi.org/10.2514/3.58379
21. Hussain, S., Shah, S., Shahzad, A.: Optimization and aerodynamic design of a soccer ball using numerical analysis. In: International Conference on Engineering and Emerging Technologies (ICEET), pp. 1–7 (2020). https://doi.org/10.1109/ICEET48479.2020.9048220
22. Igali, D., Mukhmetov, O., Zhao, Y., Fok, S.C., Teh, S.L.: Comparative analysis of turbulence models for automotive aerodynamic simulation and design. Int. J. Automot. Technol. **20**(6), 1145–52 (2019). https://doi.org/10.1007/s12239-019-0107-7
23. Kulfan, B., Bussoletti, J.: "fundamental" parameteric geometry representations for aircraft component shapes. In: 11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference (2006). https://doi.org/10.2514/6.2006-6948
24. Kummel, A., Ress, R., Pulfer, M., Hirsch, F., Breitsamter, C.: Aerodynamic design modification for utility helicopters. J. Aerosp. Eng. **36**(4), 04023022 (2023). https://doi.org/10.1061/JAEEEZ.ASENG-4785
25. Lee, J., Lee, K., Kim, B.: Aerodynamic optimal blade design and performance analysis of 3 MW wind turbine blade with AEP enhancement for low-wind-speed-sites. J. Renewable Sustain. Energy **8**(6), 063303 (2016). https://doi.org/10.1063/1.4967971
26. Lin, J., Zhang, C., Xie, X., Shi, X., Xu, X., Duan, Y.: CST-GANs: a generative adversarial network based on CST parameterization for the generation of smooth airfoils. In: 2022 IEEE International Conference on Unmanned Systems (ICUS), pp. 600–605 (2022). https://doi.org/10.1109/ICUS55513.2022.9987080
27. Liu, D., Wang, C., Gonzalez-Libreros, J., Tu, Y., Elfgren, L., Sas, G.: A review on aerodynamic load and dynamic behavior of railway noise barriers when high-speed trains pass. J. Wind Eng. Ind. Aerodyn. **239** (2023). https://doi.org/10.1016/j.jweia.2023.105458
28. Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I.: Adversarial autoencoders. In: International Conference on Learning Representations (2016)
29. Martins, J.R.: Aerodynamic design optimization: challenges and perspectives. Comput. Fluids **239**, 105391 (2022). https://doi.org/10.1016/j.compfluid.2022.105391
30. Masters, D.A., Taylor, N.J., Rendall, T.C.S., Allen, C.B., Poole, D.J.: Geometric comparison of aerofoil shape parameterization methods. AIAA J. **55**(5), 1575–1589 (2017). https://doi.org/10.2514/1.J054943
31. Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S., Geiger, A.: Occupancy networks: learning 3D reconstruction in function space. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4455–4465 (2019). https://doi.org/10.1109/CVPR.2019.00459

32. Park, J.J., Florence, P., Straub, J., Newcombe, R., Lovegrove, S.: Deepsdf: learning continuous signed distance functions for shape representation. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 165–174 (2019). https://doi.org/10.1109/CVPR.2019.00025
33. Perry, R.N., Frisken, S.F.: Kizamu: a system for sculpting digital characters. In: Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 2001, pp. 47–56. Association for Computing Machinery, New York (2001). https://doi.org/10.1145/383259.383264
34. Pu, J., Zheng, B., Leader, J.K., Wang, X.H., Gur, D.: An automated CT based lung nodule detection scheme using geometric analysis of signed distance field. Med. Phys. **35**(8), 3453–3461 (2008). https://doi.org/10.1118/1.2948349
35. Reuther, J., Jameson, A.: Aerodynamic shape optimization of wing and wing-body configurations using control theory. In: 33rd Aerospace Sciences Meeting and Exhibit (1995). https://doi.org/10.2514/6.1995-123
36. Robinson, M., MacManus, D., Christie, R.M., Sheaf, C., Grech, N.: Nacelle design for ultra-high bypass ratio engines with CFD based optimisation. Aerosp. Sci. Technol. **113**, 106191 (2020). https://doi.org/10.1016/j.ast.2020.106191
37. Sederberg, T.W., Parry, S.R.: Free-form deformation of solid geometric models. SIGGRAPH Comput. Graph. **20**(4), 151–160 (1986). https://doi.org/10.1145/15886.15903
38. Selig, M.: UIUC airfoil data site. Department of Aeronautical and Astronautical Engineering University of Illinois at Urbana-Champaign (1996)
39. Sitzmann, V., Chan, E.R., Tucker, R., Snavely, N., Wetzstein, G.: MetaSDF: meta-learning signed distance functions. In: Advances in Neural Information Processing Systems, pp. 10136–10147. Curran Associates, Inc. (2020)
40. Sitzmann, V., Zollhöfer, M., Wetzstein, G.: Scene representation networks: continuous 3D-structure-aware neural scene representations. In: Advances in Neural Information Processing Systems. Curran Associates, Inc. (2019)
41. Sobieczky, H.: Parametric airfoils and wings. In: Recent Development of Aerodynamic Design Methodologies: Inverse Design and Optimization, pp. 71–87 (1999). https://doi.org/10.1007/978-3-322-89952-1_4
42. Spurkl, L., Bardal, L.M., Saetran, L., Oggiano, L.: Low aerodynamic drag suit for cycling design and testing, Lisbon, Portugal, pp. 89–96 (2015). https://doi.org/10.5220/0005589600890096
43. Toal, D.J., Bressloff, N.W., Keane, A.J., Holden, C.M.: Geometric filtration using proper orthogonal decomposition for aerodynamic design optimization. AIAA J. **48**(5), 916–928 (2010). https://doi.org/10.2514/1.41420
44. Vignesh, S., Kumar, M.: Computational flow prediction and design optimization of vortex generators for a conventional passenger car. Int. J. Veh. Struct. Syst. **12**(3), 287–291 (2020). https://doi.org/10.4273/ijvss.12.3.12
45. Wainwright, T.R., Poole, D.J., Allen, C.B.: GPU-accelerated aerodynamic shape optimisation framework for large turbine blades. In: AIAA SCITECH 2022 Forum (2022). https://doi.org/10.2514/6.2022-1292
46. Yang, J., Hu, B., Tao, Y., Li, J.: A flexible method for geometric design of axial compressor blades. Proc. Inst. Mech. Eng. Part G J. Aerosp. Engineering **236**, 2420–2432 (2022). https://doi.org/10.1177/09544100211063078

Author Index

A
Abedjan, Ziawasch 21

B
Bamford, Tom 452
Banik, Swarnali 123
Barrett, Enda 256
Bi, Jianzhao 385
Bian, Jiang 335
Bian, Yuxuan 174

C
Chambers, Des 256
Chan, Jeffrey 223
Charland, Philippe 3
Che, Liwei 401
Chen, Ben 55
Chen, Wenguang 418
Cheng, Dawei 174
Claeys, Emmanuelle 351
Costa, Pedro Ákos 38

D
Dai, Huangyu 55
Denzler, Joachim 207
Dileo, Manuel 303
Du, Yupei 21

F
Fung, Benjamin C. M. 3

G
Gançarski, Pierre 351
Ganguly, Niloy 123
Gawlikowski, Jakob 207
Ghate, Kshitish 123
Ghosh, Surjya 123
Golebiowski, Jacek 21
Grushetskaya, Yulia 319
Gupta, Shrey 385
Gupta, Suyash 385

H
Hong, Shenda 271
Hong, Xiaojie 435
Huan, Zhaoxin 240

J
Jiang, Wen 55
Jin, Beihong 89, 141

K
Kadamala, Kevlyn 256
Kang, Le 106
Kang, Yufan 223
Keane, Andy 452

L
Lai, Weijiang 89, 141
Leckie, Christopher 223
Lepak, Łukasz 368
Li, Beibei 141
Li, Chenchen 240
Li, Fuyi 335
Li, Hongyan 271
Li, Liangzhi 435
Liang, Yuqi 174
Liu, Feiyan 435
Liu, Jingwei 271
Liu, Song 240
Liu, Xin 174, 190
Liu, Xinyue 401
Liu, Yang 385
Liu, Yao 287
Liu, Yuxi 335
Long, Guodong 335
Lu, Xuesong 190
Lv, Yimin 89

© The Editor(s) (if applicable) and The Author(s), under exclusive license
to Springer Nature Switzerland AG 2024

A. Bifet et al. (Eds.): ECML PKDD 2024, LNAI 14949, pp. 469–470, 2024.
https://doi.org/10.1007/978-3-031-70378-2

M
Ma, Fenglong 401
Ma, Xiang 55
Mahan, Akram 319
Mahdavifar, Samaneh 3
Maumy-Bertrand, Myriam 351
Metelo, Frederico 38
Miwa, Makoto 157
Mu, Tai-Jiang 106

N
Nambiar, Hrithik 123
Niebling, Julia 207
Ning, Wei 55
Ning, XiaoDong 106

O
Olmeda, Raffaele 303

P
Park, Yongbee 385
Parlier, Benjamin 72
Pindaro, Margherita 303

R
Racković, Stevo 38
Raychoudhury, Vaskar 123
Rewicki, Ferdinand 207

S
Saberioon, Mohammadmehdi 319
Saha, Snehanshu 123
Salaün, Lou 72
Salim, Flora D. 223, 335
Saqib, Mohd 3
Schachtschneider, Reyko 319
Schmidt, Philipp 21
Sen, Sougata 123
Shaikh, Alfiya M. 123
Shao, Wei 223
Shen, Jun 335
Shi, Linfeng 240
Sips, Mike 319
Soares, Cláudia 38
Sohrab, Mohammad Golam 157
Song, Zixin 435
Sun, Haoyu 174

T
Tang, Chengfu 240
Toal, David 452

W
Walenstein, Andrew 3
Wang, Jiaqi 401
Wang, Xiaoli 435
Wang, Xin 287
Wang, Xudong 240
Wang, Yu 190
Wawrzyński, Paweł 368
Wildani, Avani 385

X
Xiang, Tao 141

Y
Yang, Hong 72
Yepes, Antonio Jimeno 335

Z
Zhang, Xiaolu 240
Zhang, Yongfei 287
Zhang, Zhenhao 335
Zhao, Rui 89, 141
Zhao, Ying 418
Zheng, Yiyuan 89, 141
Zhou, Aoying 190
Zhou, Chunyi 240
Zhou, Jun 240
Zhou, Ziheng 418
Zhu, Peng 174
Zignani, Matteo 303
Züfle, Andreas 385
Zuo, Haojia 418