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.

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:

$$head_n(\hat{X}) = SoftMax(\frac{q_n \cdot k_n^\top}{\sqrt{d_k}}), \tag{4}$$

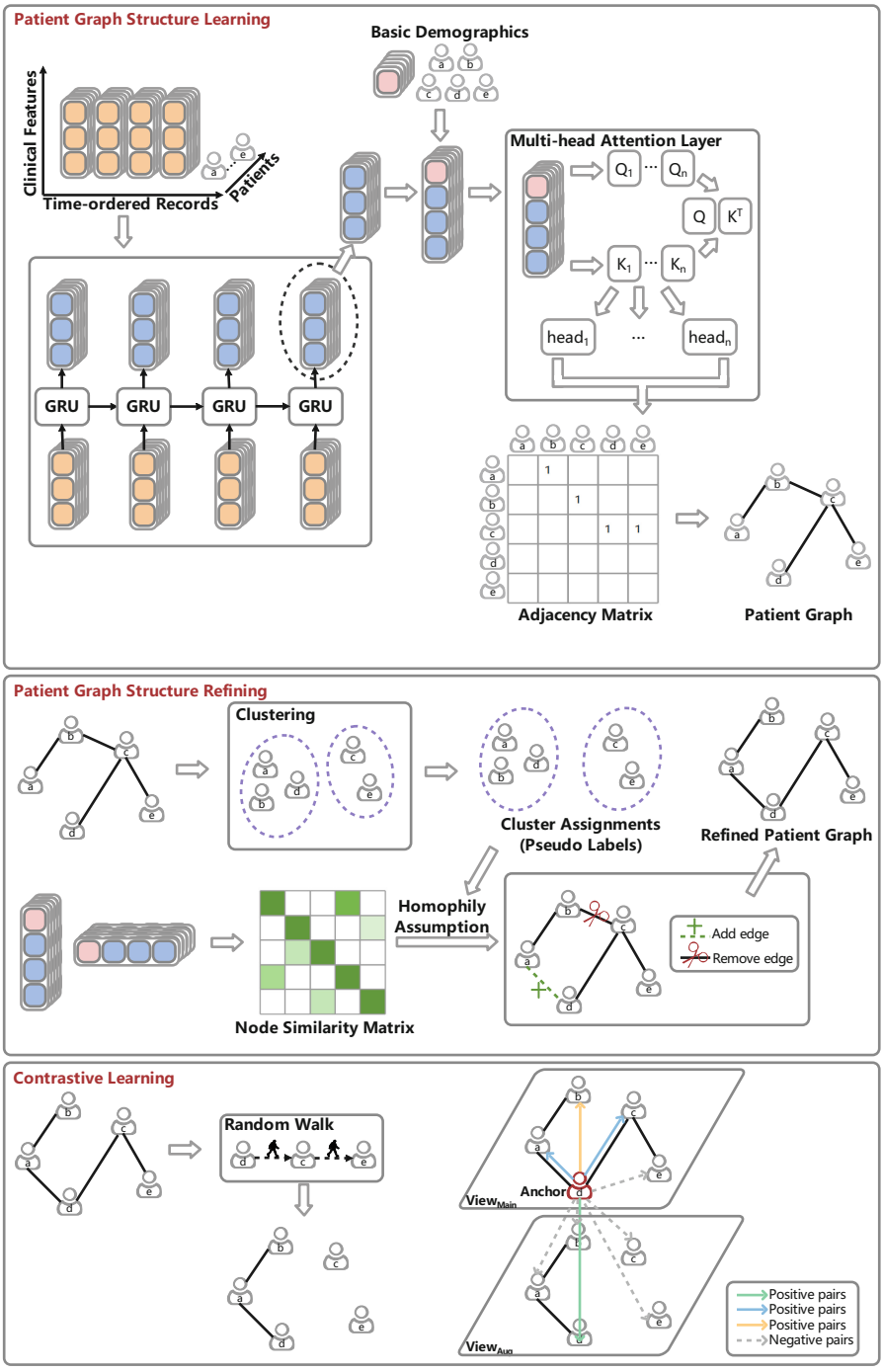

Fig. 1. The proposed network architecture.

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:

$$\begin{aligned} \varepsilon_{add}^k = \{(v_i, v_j) | Rank(Z_{ij}) \leq \gamma_{add} \cdot |\varepsilon| \cdot \tfrac{m_k}{m}, (v_i, v_j) \notin \varepsilon, \tilde{y}_i = \tilde{y}_j = k\}, \varepsilon_{add} = \bigcup_k^K \varepsilon_{add}^k, \\ \varepsilon_{del} = \{(v_i, v_j) | Rank(Z_{ij}) \geq (1 - \gamma_{del}) \cdot |\varepsilon|, (v_i, v_j) \in \varepsilon, \tilde{y}_i \neq \tilde{y}_j\}, \end{aligned} \tag{8}$$

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:

$$\begin{aligned} \mathcal{L}_{CL} = -\sum_{i=1}^{m} \frac{1}{|\mathcal{N}^i_{Main}| + N_{\tilde{y}_i} + 1} \\ log \frac{exp(\varphi(S^i_{Main}, S^i_{Aug})/\tau) + \sum_{j \in \mathcal{N}^i_{Main}} exp(\varphi(S^i_{Main}, S^j_{Main})/\tau)}{\sum_{j=1}^{m} \mathbb{1}_{[j \neq i]} exp(\varphi(S^i_{Main}, S^j_{Main})/\tau)}, \\ + \frac{\sum_{j=1, j \notin \mathcal{N}^j_{Main}}^{m} \mathbb{1}_{[\tilde{y}_i = \tilde{y}_j]} exp(\varphi(S^i_{Main}, S^j_{Main}))}{\sum_{j=1}^{m} exp(\varphi(S^i_{Main}, S^j_{Aug})/\tau)} \end{aligned} \tag{12}$$

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: $\mathbf{Our}_{\alpha}$: we treat only the anchor and its counterparts in different graph views as positives; $\mathbf{Our}_{\beta}$: we omit the node connected to the anchor, which tests the efficacy of node-level clustering on patient graphs; $\mathbf{Our}_{\gamma}$: we omit the neighbors of the anchor; $\mathbf{Our}_{\delta}$: 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_{α}	0.7311(0.0389)	0.2606(0.0285)	0.3235(0.0181)	0.2813(0.0324)
Our_{β}	0.7432(0.0215)	0.2713(0.0166)	0.3378(0.0163)	0.2809(0.0190)
Our_{γ}	0.7592(0.0156)	0.2839(0.0162)	0.3549(0.0226)	0.2978(0.0168)
Our_{δ}	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_{α}	0.7707(0.0324)	0.9088(0.0128)	0.8247(0.0152)	0.8401(0.0135)
Our_{β}	0.7861(0.0407)	0.9150(0.0195)	0.8035(0.0235)	0.8473(0.0138)
Our_{γ}	0.7943(0.0278)	0.9179(0.0151)	0.8241(0.0153)	0.8495(0.0099)
Our_{δ}	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_{α}	0.7779(0.0158)	0.9761(0.0026)	0.8665(0.0241)	0.9455(0.0021)
Our_{β}	0.7831(0.0224)	0.9772(0.0029)	0.8601(0.0276)	0.9458(0.0027)
Our_{γ}	0.7909(0.0577)	0.9767(0.0077)	0.8628(0.0253)	0.9446(0.0041)
Our_{δ}	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)