Intent Enhanced Self-supervised Hypergraph Learning for Session-Based Recommendation

Xiu Susie Fang[1], Yonggang Wu[1], Jinhu Lu[1], Xiaoyu Gu[1], Guohao Sun[1](✉), and Yong Zhan[2]

[1] Donghua University, Shanghai, China
{xiu.fang,ghsun}@dhu.edu.cn, {2222742,1209126,211310113}@mail.dhu.edu.cn
[2] Qingdao Feixiong Lingxian Technology Co., Qingdao, China
zhanyong@linkingfresh.com

Abstract. Session-based recommendation (SBR) aims to capture user intents based on a set of anonymous sessions for recommending the next item. Recent works in SBR often employ graph neural networks (GNNs) to model the transition patterns between items and have made impressive progress. However, the performance is still limited by data sparsity and complex dependency in sessions. Recently, self-supervised learning (SSL) has been applied in recommender systems because of its good ability to mine ground-truth samples from raw data, and great potential in relaxing data sparsity. We note that both sessions and individual items contain implicit user intents, and there is consistency between intents. Therefore, the SSL can be applied to construct self-supervised signals based on the implicit user intents to further alleviate the data sparsity problem in SBR, and thus improve the performance. In this paper, we propose a novel model called **I**ntent Enhanced **S**elf-Supervised **H**yper**g**raph **L**earning for session-based recommendation (ISHGL) to improve the performance. We first model the session sequence data as a global hypergraph to capture complex high-order relationships in sessions. Then, we devise a new contrastive method for self-supervised learning without additional data augmentation and complex positive/negative sample constructions. Extensive experiments on three datasets demonstrate the superiority of our model over the state-of-the-art methods.

Keywords: Recommender systems · Session-based recommendation · Self-supervised learning · Graph neural networks

1 Introduction

Recommender systems have been widely applied in various scenarios to alleviate information overload, such as e-commerce websites, mobile stream media, and so forth [25]. Due to user privacy concerns, recommender systems are often unable

A. Bifet et al. (Eds.): ECML PKDD 2024, LNAI 14950, pp. 83–99, 2024.
https://doi.org/10.1007/978-3-031-70381-2_6

to identify each user and track their long-term interests [28]. Session-based recommendation focuses on modeling user intents based on short anonymous behavior sequences to predict the next item, which has attracted wide attention due to its highly practical value.

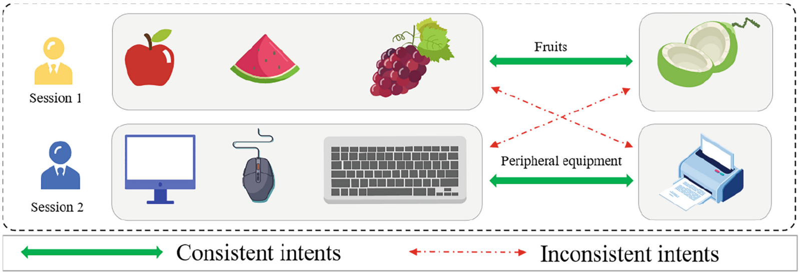

Fig. 1. An example of intent consistency between sessions and target items.

Early efforts in the SBR mainly employed K Nearest Neighbor [15] based approaches to identify similar items to the current session or Markov Chain [14] based approaches to capture sequential signals between adjacent items. The performance of these traditional methods is limited by their own modeling capabilities and data sparsity issues. Afterward, recurrent neural networks (RNNs) were introduced into SBR by considering their abilities to model sequential signals. RNN-based methods [5,6,8] model session data as unidirectional sequences in a strictly-ordered way. Nevertheless, the models that rely overwhelmingly on strict order to model session data may lead to the problem of overfitting [23]. Recently, graph neural networks (GNNs) have sparked widespread discussions, and various GNN-based methods [19,22] have also been employed to enhance SBR systems due to their ability to model complex transition relationships between items [10]. In GNN-based methods, adjacent items in a session sequence are modeled as pairwise relations, which can alleviate the temporal dependence between items. Since simple graphs are not good at modeling non-pairwise relationships between nodes [1], GNN-based methods can not model high-order relationships in session sequences well. Moreover, in SBR, the sessions usually are very short (6–7 items in one session), GNN-based methods have to use the short-term interaction data to refine the representations of items and sessions for SBR, which still suffer from the problem of data sparsity and fail to obtain accurate user intents.

Self-supervised learning (SSL) is an emerging learning paradigm that is able to mine ground-truth samples from raw data and shows good ability to alleviate data sparsity issues. Thus, some works have explored SSL in recommender systems to alleviate data sparsity. For instance, SGL [21] utilizes graph structure augmentation to create contrastive views for SSL. CL4SRes [24] introduces three data augmentation strategies for sequence-level contrastive learning. To create self-supervised signals from the raw interaction graph/sequence, these methods require additional data augmentation such as node dropout [27], where they randomly extract part of the nodes from the session sequence. Due to the fact that the sessions in SBR usually are very short, randomly extracting part of the nodes will break the connectivity of items in the sessions, limiting the performance of SSL. In order to avoid this limitation, in SBR, S^2-DHCN [23] proposes to introduce a line graph to generate self-supervised signals for SSL. However, in this method, it needs to introduce additional graph channels and graph convolution operations to construct positive/negative samples for the generation of self-supervised signals, which is complex and inefficient. Therefore, it is necessary to further improve self-supervised learning in session-based recommendations.

We note that, for one session, there is consistency of the intents between the session and its target item. For example, as shown in Fig. 1, Session 1 with (“Apple”, “Watermelon”, “Grapes”) reveals the intent of buying fruits and the target item is a coconut which is consistent with the intent. Likewise, Session 2 and its target item reveal the consistent intent to buy peripheral equipment. In contrast, the intents of different sessions are usually different from each other (inconsistency). For example, in Fig. 1, the intent of Session 1 is inconsistent with the intent of the target item in Session 2. The consistency and inconsistency between intents facilitate the creation of self-supervised signals. By this observation, we can use consistency to construct positive samples and inconsistency to construct negative samples without additional data augmentation and complex positive/negative sample constructions.

Based on the discussions mentioned above, we creatively propose Intent Enhanced Self-supervised Hypergraph Learning for session-based recommendation (ISHGL). Specifically, first, to capture complex high-order relationships between items, we model session data as a global hypergraph and construct a hypergraph convolutional neural network for information propagation. In this global graph, each hyperedge denotes a session and all items are connected, which can relax strict order dependence. Then, to overcome the data sparsity issue, we explore self-supervised learning via a new contrastive method that creates self-supervised signals between intents without extra data augmentation and complex positive/negative sample constructions. By the proposed self-supervised learning method, we can maximize the mutual information between session intents and target item intents, and thus, optimize session and item representations. Finally, we unify the self-supervised task and the recommendation task under a learning framework. The performance of the recommendation task is boosted by jointly optimizing the two tasks.

Overall, the main contributions of this paper are summarized as follows:

- We innovatively propose to create self-supervised signals between session intents and item intents to optimize the representations of sessions and items. In this way, we do not need additional data augmentation and complex positive/negative sample constructions.
- We propose an intent-enhanced self-supervised learning in hypergraph neural networks, termed ISHGL, for the session-based recommendation. ISHGL can

model complex and high-order relationships among sessions and items for effective session-based recommendation.
– Extensive experiments are conducted on three datasets, demonstrating that our approach is superior compared with the state-of-the-art models.

2 Related Work

In this section, we first review the related methods for session-based recommendation. Then, we introduce the self-supervised learning methods in session-based recommendation.

2.1 Traditional Methods

Early studies in SBR primarily rely on nearest neighbors [15]. Some of them employ cosine similarity to calculate similarity scores, but these methods ignore the transition patterns between items. Then, numerous sequential methods have been proposed to utilize chronological order to model the users' intents. For instance, FPMC [14] utilizes Markov Chains (MC) and personalized matrix factorization to capture sequential patterns and long-term preferences of users for predicting the next action. Nevertheless, Markov Chain-based methods usually focus on the transition of adjacent items, which have difficulty in capturing more complex and high-order sequential relationships.

2.2 Deep Learning-Based Methods

With the boom of deep learning, recurrent neural networks (RNNs) have been widely employed in session-based recommendations for modeling sequential relationships between items. For instance, Hidasi et al. propose GRU4Rec [5], a model that employs Gate Recurrent Units (GRUs) to model the entire session for the next-item recommendation. Li et al. [6] propose a hybrid encoder with GRUs and an attention mechanism to model the user's main intent in the current session. Liu et al. [8] propose STAMP to capture both short-term and long-term interests with multilayer perceptrons (MLPs) networks and attention mechanisms. Besides RNN-based methods, convolutional neural networks (CNNs) are another commonly used deep learning-based method in SBR to model sequential information. For example, Caser [16] regards the representations of items within the sequences as latent matrices and utilizes CNNs to model users' general preferences and sequential patterns. Despite achieving remarkable success through various deep learning-based approaches, these methods rely heavily on sequential relationships between adjacent items to generate representations of sessions and items, which ignore information between non-adjacent items.

Due to the advantages of graph neural networks (GNNs) in modeling complex transition relationships between nodes, they have been widely adopted in SBR in recent years. For instance, Wu et al. [22] propose SR-GNN, which first constructs historical session sequences as directed graphs and uses a gated graph neural network (GGNN) to capture intricate item transition information. Based on this work, Xu et al. [25] introduce a graph contextualized self-attention model (GC-SAN), which leverages GNNs to capture local dependencies and employs a self-attention mechanism for long-range dependencies. Qiu et al. [13] investigate the inherent order of session sequences and develop FGNN to exploit users' latent intents. To alleviate the problem of lossy session encoding, Chen et al. [3] propose edge-order preserving aggregation and shortcut graph attention to get a lossless encoding of sessions. GCE-GNN [19] proposes a unified model that exploits the session-level item embeddings within the current session and the global-level item embeddings over all sessions, and integrates both item embeddings to generate the final session embedding. To capture information from items without direct connections and deal with the overfitting problems of GNN-based approaches, Pan et al. [11] propose a star graph neural network and apply a highway network.

However, the above-mentioned GNN-based methods face challenges in capturing high-order relationships between items to generate more accurate representations. Recently, some works have extended GNN to hypergraph to enhance item representations in recommender systems. Wang et al. [18] propose SHARE that constructs a hypergraph for each session with a sliding window and uses a hypergraph attention network (HGAT) to distinguish the importance of different intents. Xia et al. [23] propose a dual-channel hypergraph convolutional network with a self-supervised task to enhance hypergraph modeling. Li et al. [7] propose HIDE that constructs a hypergraph for each session and disentangles the intents under each item click at micro and macro levels. Our work is based on hypergraph neural networks, considering their advantages in modeling high-order relationships.

2.3 Self-supervised Learning

Self-supervised learning has received considerable attention in recent years as a learning paradigm that reduces the dependence on manual labels and can enable training on large amounts of unlabeled data. Initially, SSL was used in the domains of computer vision and natural language processing, where it augments the raw data by employing techniques such as image rotation/clipping and sentence masking [27]. Recently, SSL has also been applied to graph-structured data. DGI [17] maximizes the mutual information between pairs of local patches and global graphs to learn node representations. GraphCL [26] designs four types of graph data augmentations to obtain correlated views for invariant representation learning. ASP [2] effectively preserves both attribute and structure information from the input graph and learns node representations by performing contrastive learning across different graph views.

Inspired by the success of SSL in other tasks, some works have applied SSL to sequential recommendation tasks. S^3-Rec [29] utilizes the intrinsic data correlation to extract self-supervision signals and enhances the data representations via pre-training methods. S^2-DHCN proposes a dual channel hypergraph convolutional network and integrates a self-supervised task to enhance the performance of SBR [23]. Although this method has achieved satisfactory results, it suffers from complex positive/negative sample constructions. Specifically, it additionally introduces a line graph channel and utilize two encoders to generate self-supervised signals, which is inefficient, especially in big-scale data.

3 Preliminaries

3.1 Problem Statement

In session-based recommendation, the set of sessions is represented as $S = \{s_1, s_2, ..., s_M\}$ and $V = \{v_1, v_2, ..., v_N\}$ denotes all items in the dataset, where M and N are the number of sessions and items respectively. Each session is represented as a sequence $s_i = \{v_{i,1}, v_{i,2}, ..., v_{i,m}\}$ ordered by timestamps, where $v_{i,k} \in V (1 \leq k \leq m)$ denotes an interacted item of an anonymous user within the session s_i and m represents the length of the session. Given session s_i, the task of session-based recommendation is to recommend the next most likely clicked item $v_{i,m+1}$. In fact, the recommender system generates the probability distribution of all candidate items. The items with top-K largest probability scores are recommended.

3.2 Hypergraph

In many real applications, the data structure could go beyond pairwise connections and even far more complicated [7]. To model this type of relationship, hypergraphs introduce the hyperedges which can connect more than two nodes. Formally, a hypergraph is defined as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, which includes a node set $\mathcal{V}$ containing N unique nodes and a hyperedge set $\mathcal{E}$ containing M hyperedges. Each hyperedge $e \in \mathcal{E}$ connects at least two nodes which is assigned a weight W_{ee}, and all the weights formulate a diagonal matrix $W \in \mathbb{R}^{M \times M}$. The hypergraph $\mathcal{G}$ can be represented by an association matrix $H \in \mathbb{R}^{N \times M}$, with entries defined as:

$$H_{ie} = \begin{cases} 0, & if \quad v_i \notin e \\ 1, & if \quad v_i \in e \end{cases}. \tag{1}$$

For every node and hyperedge, their degree D_{vv} and B_{ee} are defined as $D_{vv} = \sum_{e=1}^{M} W_{ee} H_{ve}$ and $B_{ee} = \sum_{v=1}^{N} H_{ve}$. Also, both D and B are diagonal matrices.

4 Methodology

In this section, we mainly present the details of the model. We first construct a hypergraph based on session sequences. Then, we obtain item representations via hypergraph convolutional neural networks. After getting the item representations, we utilize the attention mechanism to obtain session representation and generate user intent for recommendation. Finally, we construct of SSL task and optimize the ISHGL model via joint learning. The overview of ISHGL is shown in Fig. 2.

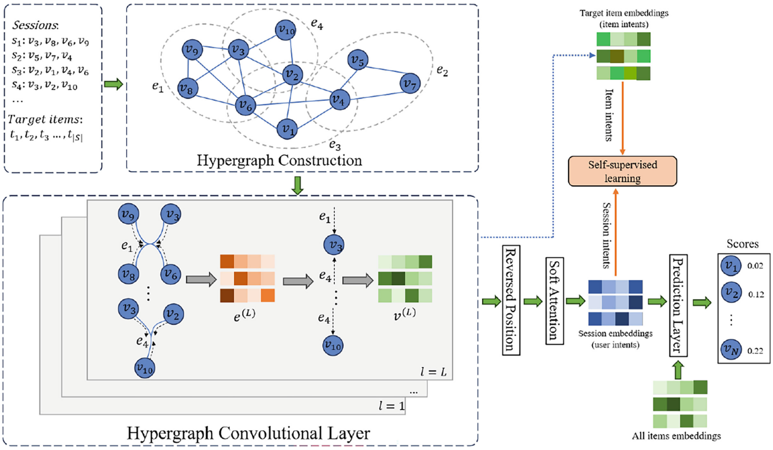

Fig. 2. Overview of ISHGL.

4.1 Hypergraph Construction

In order to capture the high-order relations among sessions, we employ an undirected hypergraph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ to model sessions, in which each hyperedge denotes a session and connects any number of nodes. Formally, we denote each hyperedge as $[v_{s,1}, v_{s,2}, ..., v_{s,m}] \in \mathcal{E}$. For a more detailed description, Fig. 2 has shown the process of constructing a global hypergraph from sessions. First, the original sessions s_1, s_2, s_3, s_4 are transformed into four hyperedges e_1, e_2, e_3, e_4. Then, the items within each session are connected in pairs regardless of sequential order. The size of hypergraph is determined by the number of sessions and items in the datasets.

4.2 Hypergraph Convolutional Neural Network

Based on the construction of the hypergraph, we apply the hypergraph convolutional neural network (HGCN) layer to obtain complex high-order relationships between items in all sessions. Hypergraph convolution is an extension of traditional graph convolution that utilizes the hyperedges as a transition during information propagation. The whole information transfer process can be divided into two stages, the first stage is information aggregation from nodes to hyperedges and the second stage is information aggregation from hyperedges to nodes. The main challenge of defining a convolutional network on a hypergraph is how to propagate the representations of neighbor nodes. Referring to previous works [1,4,20], we define hypergraph convolution as follows:

$$x_t^{(l+1)} = \sum_{i=1}^{N} \sum_{e=1}^{M} H_{te} H_{ie} W_{ee} x_i^{(l)}, \tag{2}$$

where $x_t^{(l+1)}$ represents the embedding of the t-th node at the $(l+1)$-th layer and W_{ee} is configured as 1 for each e. After row normalization, we get the matrix form of Eq. (2) as follows:

$$X^{(l+1)} = D^{-1}HWB^{-1}H^{T}X^{(l)}, \tag{3}$$

where D and B are the degree matrices of the node and hyperedge in a hypergraph. H represents the incidence matrix of the hypergraph and W denotes the weight matrix of hyperedges. $X^{(l)}$ denotes the representations of the whole item set at l-th layer. The hypergraph convolution can be viewed as a two-stage node-hyperedge-node information transfer process, which can better extract node features based on the hypergraph structure. Specifically, $H^{T}X^{(l)}$ represents the aggregation process of features from nodes to hyperedges. After getting hyperedge features, the updated node features are obtained by multiplying matrix H to aggregate the related hyperedge features. For the given initial item embeddings $X^{(0)}$, each layer can generate the representations of $X^{(i)}$ from hypergraph convolutional layers 0 to L. We average the representations of all layers to get the item representations:

$$X_h = \frac{1}{L+1}\sum_{l=0}^{L} X^{(l)}, \tag{4}$$

where X_h denotes the final global-level item representations.

4.3 Session Representation Learning

After getting the item representations, we aggregate them to obtain the session embedding. The session embedding denotes the current intent of the anonymous user, and there exists consistency between the current intent and the next item. In the session sequence, each item carries positional information and the items clicked later are more representative of the user intents [19]. We adopt reversed position information for session representation learning. The position embedding matrix is represented as $P = [p_1, p_2, ..., p_m]$, where $p_i \in \mathbb{R}^d$ denotes the vector of position and m represents the length of the session sequence. For item v_i in a given session s, we integrate item embedding h_i with position information p through the following formula:

$$x_i^* = \tanh\left(W_1\left(h_i \| p_{m-i+1}\right) + b\right), \tag{5}$$

where $\|$ represents the concatenation operation. W_1 and b denote learnable parameters, respectively.

To better extract the user's current intent, we further consider the different priorities of items within the session sequence. To get the representation of the current session, we average the embeddings of items within the session sequence:

$$s^* = \frac{1}{m}\sum_{i=1}^{m} h_i, \tag{6}$$

where s^* is the embedding of session s. Then, we employ a soft-attention mechanism to compute the weight coefficient for each item. The formula is as follows:

$$\alpha_i = q^T \sigma(W_2 s^* + W_3 x_i^* + b_2), \tag{7}$$

where $\sigma(\cdot)$ represents sigmoid function. $q \in \mathbb{R}^d$ is the learnable attention parameter which is used to learn the item weight α_i. $\{W_2, W_3\} \in \mathbb{R}^{d\times d}$ and b_2 are learnable parameters. Then, the user's general intent embedding U_h is calculated as follows:

$$U_h = \sum_{i=1}^{m} \alpha_i h_i. \tag{8}$$

4.4 Recommendation Generation

Based on the learned user's intent U_h, the score $\hat{z}_i$ for each candidate item is calculated by doing the inner product:

$$\hat{z}_i = U_h^T h_i. \tag{9}$$

After that, we apply a softmax function to compute the probabilities of each item being the next one in the session:

$$\hat{y}_i = softmax(\hat{z}_i). \tag{10}$$

We adopt cross-entropy as the optimization objective to learn the parameters and the loss function is:

$$\mathcal{L}_{ce} = -\sum_{i=1}^{N} y_i \log\left(\hat{y}_i\right) + (1 - y_i)\log\left(1 - \hat{y}_i\right), \tag{11}$$

where y_i represents the one-hot encoding vector of the ground truth.

4.5 Enhancing SBR with Self-supervised Learning Task

Although hypergraph learning captures more high-order relationships compared to traditional graph learning, session-based recommendation tends to suffer from the issue of data sparsity which leads to sub-optimal item representations. Inspired by the successful practices of self-supervised learning on traditional graphs, we innovatively propose a new contrastive learning approach to enhance session-based recommendation.

By observing session data, we notice that different session sequences usually reflect various user intents and each individual item also contains user intents. For a given session $s_i = [v_{i,1}, v_{i,2}, ..., v_{i,m}]$, the intent of this session is represented as U_h^i and the intent reflected in next item $v_{i,m+1}$ is represented as $u^i \in X_h$. The next item $v_{i,m+1}$ is also called target item. We note that there is consistency between the current session intent and the user interest reflected in the next item. By maximizing the mutual information between the session intent and the item intent through contrastive learning, the recommendation model can obtain more accurate item representations for better recommendation performance.

For each mini-batch including n sessions and n next items in training, there is a bi-directional mapping between them. Specifically, each session or item embedding represents user intent and a pair of intents (U_h^i, u^i) belonging to the same session can be labeled as the ground truth of each other for self-supervised learning. We label other items as the negative samples U^-. For pairs of intents from different sessions, we label them as negative sample pairs. We build self-supervised signals in this way without complex data augmentation operations. To maximize the mutual information between the intent pairs, we follow noise-contrastive estimation with a standard binary cross-entropy loss between positive samples and negative samples [9] as our learning objective:

$$\mathcal{L}_{ssl} = -\sum_{i=1}^{M} \left(\log \sigma \left(sim\left(U_h^i, u^i\right)\right) + \log \sigma \left(1 - sim\left(U_h^i, u^-\right)\right)\right), \tag{12}$$

where $u^- \in U^-$ represents the negative sample and $sim(\cdot,\cdot)$ represents a similarity metric function to measure the similarity between each representation. We use dot product as $sim(\cdot,\cdot)$ to simplify the calculation.

4.6 Model Optimization

Finally, the recommendation task and the self-supervised task are unified into a primary & auxiliary learning framework, where the former is the primary task and the latter is the auxiliary task. The joint learning objective is as follows:

$$\mathcal{L}_{loss} = \mathcal{L}_{ce} + \beta \mathcal{L}_{ssl}, \tag{13}$$

where β represents a hyperparameter to control the magnitude of the self-supervised task.

5 Experiments

In this section, we mainly describe the experimental settings, including datasets, baselines, evaluation metrics and detailed analysis of experimental results. We are committed to finding answers to the following questions:

- **Q1:** How dose the performance of ISHGL compare to state-of-the-art (SOTA) session-based recommendation approaches?
- **Q2:** How do different components in the ISHGL affect the performance?
- **Q3:** Is the proposed model ISHGL sensitive to hyperparameters? How do different hyperparameter settings affect the model's performance?

5.1 Experimental Setup

Datasets and Preprocessing. We evaluate the proposed model on three real-world datasets: Tmall[1], Diginetica[2] and Last.FM[3], which are commonly used in session-based recommendation research. The statistics of the used datasets are presented in Table 1.

Following the previous works [13,22], we filter out sessions whose length is 1 and filter out the items that appear less than 5 times. We set the most recent data (e.g., last week) as test data and the remaining data as training data. Furthermore, we adopt a sequence splitting method to generate sequence $([v_1], v_2)$, $([v_1, v_2], v_3)$, ..., $([v_1, v_2, v_3, ..., v_{m-1}], v_m)$ for session $S = \{v_1, v_2, v_3, ..., v_{m-1}, v_m\}$. The target items is a set of the labels of the sessions.

Table 1. Statistics of datasets.

Datasets	Tmall	Diginetica	Last.FM
#clicks	818,479	982,291	3,835,706
#train sessions	351,268	719,470	2,837,644
#test sessions	25,898	60,858	672,519
#items	40,782	43097	38,615
#average length	6.69	5.12	9.16

Baselines. To investigate the performance of the proposed model, we choose the competitive methods as baselines for comparison:

- **Item-KNN** [15] recommends items which are similar to the previous items in the ongoing session by computing cosine similarity between two items.
- **FPMC** [14] is a Markov-chain based hybrid model for the next-basket recommendation.
- **GRU4Rec** [5] is an RNN-based model which employs Gate Recurrent Units (GRU) to capture sequential information and utilizes a session-parallel mini-batch training strategy.
- **NARM** [6] is also an RNN-based model which combines Gate Recurrent Units (GRU) and an attention mechanism to model user's sequential behavior and extract user's main intent in the current session.
- **STAMP** [8] is a short-term attention/memory priority model that can simultaneously capture both the users' long-term interests in general and their short-term attention.

[1] https://tianchi.aliyun.com/dataset/42.
[2] https://competitions.codalab.org/competitions/11161.
[3] http://mtg.upf.edu/static/datasets/last.fm/lastfm-dataset-1K.tar.gz.

- **SR-GNN** [22] constructs each session sequence as a directed session graph and employs gated graph neural networks to capture complex transition patterns between items.
- **FGNN** [13] proposes a multiple weighted graph attention layer (WGAT) to propagate the information between items and a Readout function to generate graph level representation for item recommendation.
- **GC-SAN** [25] employs both graph neural network and self-attention mechanism to extract local contextual information of sequences and capture global dependencies between distant items.
- **SHARE** [18] proposes to model the session sequence as a hypergraph with sliding windows and employs a hypergraph attention network to extract user intent from various contextual windows.
- S^2**-DHCN** [23] proposes a dual channel hypergraph convolutional network to capture beyond pairwise relations and integrate a self-supervised task to improve hypergraph modeling.
- **HIDE** [7] models the possible interest transitions from distinct perspectives and disentangles the intents in micro and macro manners.

Evaluation Metrics. Following the previous works [7,16], we evaluate the performance of the proposed model adopting the metrics of P@K (Precision) and M@K (Mean Reciprocal Rank) where K is 10 or 20.

Implementation Details. Following the previous works [12,22], we fix both embedding dimension and batch size at 100 for all models. Additionally, We adopt the Adam optimizer with an initial learning rate of 0.001, which decreases by a rate of 0.1 for every 3 epochs. To alleviate the overfitting problem, we set the L_2 regularization to 10^{-5} and apply an early terminating strategy. All parameters of these models are initialized using Gaussian distribution with a mean of 0 and a standard deviation of 0.1 and the initial item embeddings $X^{(0)}$ are also randomly initialized. For the GNN-based models, we search for the optimal number of layers within $\{1, 2, 3, 4, 5\}$. The coefficient for the strength of SSL is chosen from {0.001, 0.005, 0.01, 0.02, 0.03, 0.04, 0.05, 0.1, 0.5, 1}. Furthermore, for all baselines, we use their reported optimal parameter settings to achieve a fair comparison. The implementation of our proposed model is available at https://github.com/YanYanYanYuYao/ISHGL.

5.2 Overall Performance (Q1)

In this section, we compare the proposed model ISHGL with the state-of-the-art baselines to demonstrate its performance. The experimental results of all methods are shown in Table 2. We have the following observations.

First, traditional methods underperform all deep learning-based methods except GRU4Rec, proving the limitation of traditional methods in capturing complex session information. This indicates that it's insufficient to make recommendations only based on co-occurrence or a simple first-order transition

Table 2. Experimental results (%) on the three datasets

Models	Tmall				Diginetica				Last.FM			
	P@10	M@10	P@20	M@20	P@10	M@10	P@20	M@20	P@10	M@10	P@20	M@20
Item-KNN	6.65	3.11	9.15	3.31	25.07	10.77	35.75	11.57	11.89	4.42	15.27	4.79
FPMC	13.10	7.12	16.06	7.32	15.43	6.20	26.53	6.95	8.53	3.56	12.91	3.85
GRU4Rec	9.47	5.78	10.93	5.89	17.93	7.33	29.45	8.33	12.85	5.18	14.94	5.57
NARM	19.17	10.42	23.30	10.70	35.44	15.13	49.70	16.17	14.64	6.39	18.07	6.73
STAMP	22.63	13.12	26.47	13.36	33.98	14.26	45.64	14.32	13.97	6.26	17.23	6.68
SR-GNN	23.41	13.45	27.57	13.72	36.86	15.52	50.73	17.59	15.53	6.74	19.74	7.18
FGNN	20.67	10.07	25.24	10.39	37.72	15.59	50.58	16.84	14.86	6.51	18.95	6.94
GC-SAN	21.32	12.43	25.38	12.72	37.86	16.89	50.84	17.79	16.25	7.49	22.65	8.42
SHARE	25.14	14.13	30.46	14.57	39.52	17.12	52.73	18.05	15.57	6.68	19.87	7.01
S^2-DHCN	28.64	16.06	34.43	16.47	39.53	17.17	52.76	18.09	18.12	8.06	24.84	8.52
HIDE	30.43	16.76	36.36	17.20	39.69	17.18	53.01	18.11	18.65	8.11	25.15	8.56
ISHGL	**31.11**	**17.82**	**37.01**	**18.32**	**40.34**	**17.63**	**53.61**	**18.55**	**19.37**	**8.68**	**26.42**	**9.14**
Improv.(%)	2.23	6.32	1.79	6.51	1.63	2.60	1.13	2.54	3.86	7.02	5.05	6.78

* The best results are highlighted in bold face and the underline means the second-best results. *Improve.* means improvement over the state-of-art methods

matrix. Deep learning-based methods achieve better performance. Compared with GRU4Rec, both NARM and STAMP not only leverage sequential information but also use the attention mechanism to capture long-term preferences. In addition, we find that GNN-based models are better than previous methods, which demonstrates that modeling session sequences as graphs can better capture the transition relationships between items. Specifically, the hypergraph-based methods (i.e., SHARE, S^2-DHCN and HIDE) exhibit better performance than simple graph-based methods (i.e., SR-GNN, FGNN, and GC-SAN). This demonstrates that capturing high-order relationships through hypergraph neural networks is beneficial for the session-based recommendation task.

Second, our model ISHGL outperforms all baselines. Compared with traditional methods and RNN-based methods, the proposed method has achieved better performance because our model can utilize graph-structured data to capture more accurate user intent. In addition, our proposed method obtains more competitive results compared with other self-supervised learning methods. Although S^2-DHCN and our proposed method both have hypergraph architecture, the proposed method of constructing SSL enhances the performance of our method. Specifically, we create self-supervised signals between session intents and target item intents without additional data augmentation and complex positive/negative sample constructions, while S^2-DHCN additionally introduces line graph channel and two encoders for SSL, which may obtain sub-optimal signals.

5.3 Ablation Study (Q2)

In this section, we conduct ablation study to investigate the contribution of each component in our model. We define the following three variants:

- ISHGL-H: This variant removes all hypergraph layers and only retains the attention mechanism to capture local context information.

– ISHGL-AT: This variant replaces the attention mechanism with averaging item representations to explore the effect of the soft-attention mechanism.
– ISHGL-SSL: This variant removes the self-supervised signal to investigate the impact of the proposed new self-supervised learning task.

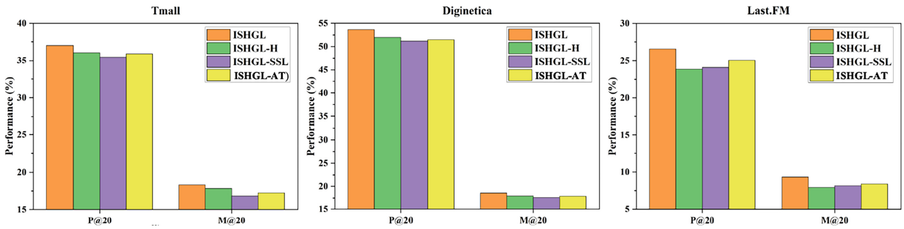

Fig. 3. Comparison of ablation experimental results.

From Fig. 3, we observe the contribution of each component on three datasets. In general, our proposed model ISHGL shows the best performance, which indicates that hypergraph neural networks combined with a new self-supervised task can obtain significant performance gains. When removing the self-supervised learning and hypergraph layers respectively, the recommendation performance decreases on three datasets. It is noticeable that the performance drops most significantly when removing hypergraph layers in *Last.FM*, which shows that global context in long sessions is more beneficial to capture user intents. In addition, the soft-attention mechanism plays an important role in short sessions because it aims to distinguish the importance of different items in a session.

5.4 Hyperparameters Analysis (Q3)

We explore how the key hyperparameters, such as the number of hypergraph layers L and the hyperparameters β, influence the performance of ISHGL.

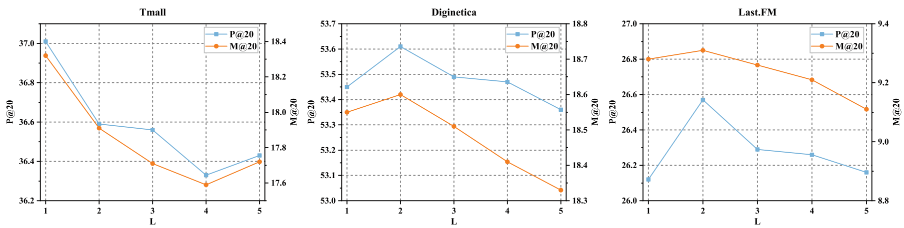

Fig. 4. The impact of different hypergraph convolution layers.

Effect of the Number of Hypergraph Layers. We report the results in Fig. 4 by ranging L within $\{1, 2, 3, 4, 5\}$. We can find that for both *Diginetica* and *Last.FM* datasets, the performance improves as the number of layers increases, and the best performance is achieved in a two-layer setting. Multi-layer hypergraph convolutions are able to mine more effective information in these two datasets. As the number of layers continues to increase, the performance gradually decreases. For *Tmall* dataset, one layer is the best and the performance decreases with the increasing number of layers. The possible reasons for the performance decrease are that integrating more extra information would disguise the true intent of the current session and too many hypergraph layers make the model over-smoothing.

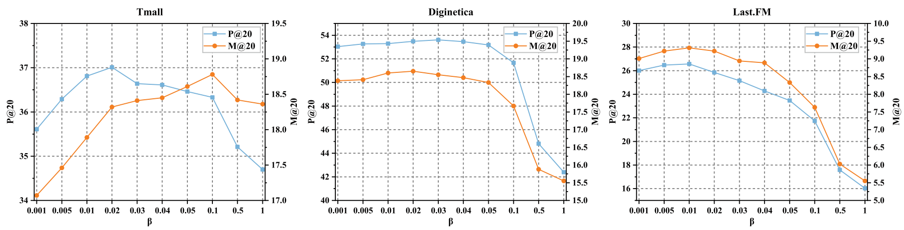

Fig. 5. The impact of the magnitude of the SSL task.

Effect of the Hyperparameter β. We report the performance in Fig. 5 with a set of representative β values in $\{0.001, 0.005, 0.01, 0.02, 0.03, 0.04, 0.05, 0.1, 0.5, 1\}$ to control the magnitude of the SSL task. From the results, we find that the recommendation task gets a performance boost when the self-supervised task is added into training. For all the datasets, with the rise of β, the recommendation performance increases first and then decreases. We think that this change in performance is owing to the gradient conflicts between the recommendation task and the self-supervised task. In addition, for *Tmall* dataset, we note that two evaluation metrics (P@20 and M@20) do not achieve optimal performance at the same β and a balance needs to be made in choosing the value of β. For *Diginetica* and *Last.FM* datasets, all evaluation metrics show similar variation trends.

6 Conclusion

In this paper, we propose a novel model ISHGL for session-based recommendation. In order to capture high-order interactive information of items, our work transforms all sessions into a hypergraph and uses hypergraph convolutional neural networks to propagate information. To alleviate the data sparsity problem, we explore self-supervised learning between session intents and item intents.

Then, we combine the recommendation task with the self-supervised learning task under a unified learning framework. In the experiments, our model outperforms the state-of-the-art methods, demonstrating the superiority of our model.

References

1. Bai, S., Zhang, F., Torr, P.H.: Hypergraph convolution and hypergraph attention. Pattern Recogn. **110**, 107637 (2021)
2. Chen, J., Kou, G.: Attribute and structure preserving graph contrastive learning. In: AAAI, pp. 7024–7032 (2023)
3. Chen, T., Wong, R.C.W.: Handling information loss of graph neural networks for session-based recommendation. In: SIGKDD, pp. 1172–1180 (2020)
4. Han, J., et al.: DH-HGCN: dual homogeneity hypergraph convolutional network for multiple social recommendations. In: SIGIR, pp. 2190–2194 (2022)
5. Hidasi, B., Karatzoglou, A., Baltrunas, L., Tikk, D.: Session-based recommendations with recurrent neural networks. arXiv preprint arXiv:1511.06939 (2015)
6. Li, J., Ren, P., Chen, Z., Ren, Z., Lian, T., Ma, J.: Neural attentive session-based recommendation. In: CIKM, pp. 1419–1428 (2017)
7. Li, Y., et al.: Enhancing hypergraph neural networks with intent disentanglement for session-based recommendation. In: SIGIR, pp. 1997–2002 (2022)
8. Liu, Q., et al.: STAMP: short-term attention/memory priority model for session-based recommendation. In: SIGKDD, pp. 1831–1839 (2018)
9. Oord, A.V.D., Li, Y., Vinyals, O.: Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748 (2018)
10. Pan, Z., Cai, F., Chen, W., Chen, C., Chen, H.: Collaborative graph learning for session-based recommendation. TOIS **40**(4), 1–26 (2022)
11. Pan, Z., Cai, F., Chen, W., Chen, H., De Rijke, M.: Star graph neural networks for session-based recommendation. In: CIKM, pp. 1195–1204 (2020)
12. Pang, Y., et al.: Heterogeneous global graph neural networks for personalized session-based recommendation. In: WSDM, pp. 775–783 (2022)
13. Qiu, R., Li, J., Huang, Z., Yin, H.: Rethinking the item order in session-based recommendation with graph neural networks. In: CIKM, pp. 579–588 (2019)
14. Rendle, S., Freudenthaler, C., Schmidt-Thieme, L.: Factorizing personalized Markov chains for next-basket recommendation. In: WWW, pp. 811–820 (2010)
15. Sarwar, B., Karypis, G., Konstan, J., Riedl, J.: Item-based collaborative filtering recommendation algorithms. In: WWW, pp. 285–295 (2001)
16. Tang, J., Wang, K.: Personalized top-n sequential recommendation via convolutional sequence embedding. In: WSDM, pp. 565–573 (2018)
17. Veličković, P., Fedus, W., Hamilton, W.L., Lió, P., Bengio, Y., Hjelm, R.D.: Deep graph infomax. In: ICLR (2019)
18. Wang, J., Ding, K., Zhu, Z., Caverlee, J.: Session-based recommendation with hypergraph attention networks. In: SDM, pp. 82–90 (2021)
19. Wang, Z., et al.: Global context enhanced graph neural networks for session-based recommendation. In: SIGIR, pp. 169–178 (2020)
20. Wu, F., Souza, A., Zhang, T., Fifty, C., Yu, T., Weinberger, K.: Simplifying graph convolutional networks. In: ICML, pp. 6861–6871 (2019)
21. Wu, J., et al.: Self-supervised graph learning for recommendation. In: SIGIR, pp. 726–735 (2021)

22. Wu, S., Tang, Y., Zhu, Y., Wang, L., Xie, X., Tan, T.: Session-based recommendation with graph neural networks. In: AAAI, pp. 346–353 (2019)
23. Xia, X., et al.: Self-supervised hypergraph convolutional networks for session-based recommendation. In: AAAI, pp. 4503-4511 (2021)
24. Xie, X., et al.: Contrastive learning for sequential recommendation. In: ICDE, pp. 1259–1273 (2022)
25. Xu, C., et al.: Graph contextualized self-attention network for session-based recommendation. In: IJCAI, pp. 3940–3946 (2019)
26. You, Y., Chen, T., Sui, Y., Chen, T., Wang, Z., Shen, Y.: Graph contrastive learning with augmentations. In: NeurIPS, vol. 33, pp. 5812–5823 (2020)
27. Yu, J., Yin, H., Xia, X., Chen, T., Li, J., Huang, Z.: Self-supervised learning for recommender systems: a survey. TKDE (2023)
28. Yuan, J., Ji, W., Zhang, D., Pan, J., Wang, X.: Micro-behavior encoding for session-based recommendation. In: ICDE, pp. 2886–2899 (2022)
29. Zhou, K., et al.: S3-rec: self-supervised learning for sequential recommendation with mutual information maximization. In: CIKM, pp. 1893–1902 (2020)