Enhancing Session-Based Recommendation with Multi-granularity User Interest-Aware Graph Neural Networks

Cairong Yan(✉), Yiwei Zhang, Xiangyang Feng, and Yanglan Gan

School of Computer Science and Technology, Donghua University, Shanghai, China
{cryan,fengxy,ylgan}@dhu.edu.cn, ywzhang@mail.dhu.edu.cn

Abstract. Session-based recommendation aims at predicting the next interaction based on short-term behaviors within an anonymous session. Conventional session-based recommendation methods primarily focus on studying the sequential relationships of items in a session while often failing to adequately consider the impact of user interest on the next interaction item. This paper proposes the **M**ulti-granularity **U**ser **I**nterest-aware **G**raph **N**eural **N**etworks (MUI-GNN) model, which leverages item attributes and global context information to capture users' multi-granularity interest. Specifically, in addition to capturing the sequential information within sessions, our model incorporates individual and group interest of users at item and global granularity, respectively, enabling more accurate item representations. In MUI-GNN, a session graph utilizes the sequential relationships between different interactions to infer the scenario of the session. An item graph explores individual user interest by searching items with similar attributes, while a global graph mines similar behavior patterns between different sessions to uncover group interest among users. We apply contrastive learning to reduce noise interference during the graph construction process and help the model obtain more contextual information. Extensive experiments conducted on three real-world datasets have demonstrated that the proposed MUI-GNN outperforms state-of-the-art session-based recommendation models.

Keywords: Recommender system · Session-based recommendation · Graph neural network · Self-supervised learning

1 Introduction

Recommender systems can effectively alleviate the issue of information overload encountered in the digital age. They are widely used in domains such as online shopping, social applications, and news media. Traditional recommendation methods [1,7,23] predict items or services based on the interaction history of a specific user over a long period. However, in some real-world scenarios, user information may be anonymous, while platforms can record only short-term

H. Gao et al. (Eds.): CollaborateCom 2023, LNICST 563, pp. 291–307, 2024.
https://doi.org/10.1007/978-3-031-54531-3_16

interactions of a user within a session. The research on session-based recommendation has consequently emerged, which employs interactions logged in the session to predict the next item or service, without relying on the user identity [22].

Early researchers use Markov chains (MCs) to capture short-term transition relationships between items in a session [6,18], which assume that the next item is only related to the most recent one or a few preceding items. Recurrent Neural Networks (RNNs) are then utilized to solve sequence-related problems. RNN-based approaches [8,19] consider each session as a sequential sequence and have shown fine results. However, RNNs heavily rely on the temporal order of items in a session, limiting the accuracy of item predictions. In recent years, Graph Neural Networks (GNNs) [27] have become popular in session-based recommendation tasks [16,17,25,26,30]. GNN methods employ graph structures to transfer information between items in a session. These methods have demonstrated superior performance compared to previous models.

Despite achieving remarkable results, most previous methods primarily focus on analyzing the sequential patterns of items in sessions without exploring the multi-granularity user interest adequately. As a result, the potential wealth of auxiliary information can be untapped during calculation. While session-based recommendation mainly addresses short and anonymous sequences, it is possible to incorporate individual and global user interest by combining item attribute information and global contextual information, respectively. For instance, Fig. 1 illustrates an example of a session-based recommendation task, demonstrating the collective utilization of sequential relationships with individual and group user interest contributes to more comprehensive and enriched recommendation outcomes.

From the granularity of a session, we can infer the session scenario and user demands by the sequential relationships among different items within the session. As in Fig. 1, we recommend items such as a mouse and a USB drive based on the presence of a keyboard and a computer in the session. At the item granularity, we explore individual user interest by identifying items with similar attributes to those already recorded in the session. As can be seen, we may recommend a black phone (in the same category as iPhone) or an iPad (in the same brand as iPhone). In terms of global granularity, we measure the similarity between different sessions by comparing the frequency of occurrences of the same item. Then we are able to understand the group interest of multiple users with similar behavior patterns. In the figure, a data cable and Airpods are recommended based on the contextual information derived from other sessions. Therefore, by analyzing user interest at different levels of granularity, the comprehensiveness and diversity of recommendation results can be further improved.

Self-supervised learning(SSL) [9,14] has recently gained significant attention recommender systems. In our case, we employ self-supervised learning to facilitate the model in studying global representation information. Additionally, given the large number of items with similar attributes, contrastive learning methods

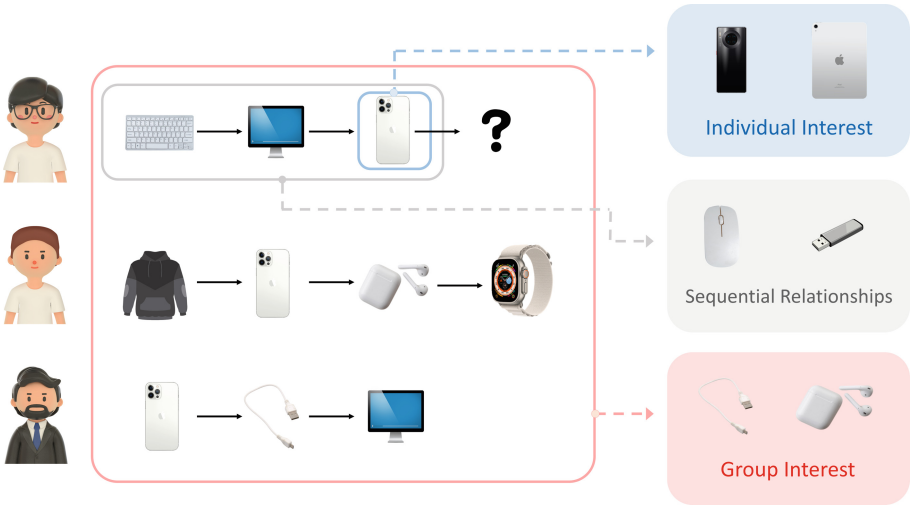

Fig. 1. An example of session-based recommendation incorporating multi-granularity user interest. The gray box recommends items by sequential relationships in the session. The blue and red boxes capture the users' multi-granularity interest and offer richer options to enhance recommendation performance. (Color figure online)

can also be used to reduce noise interference during the initial construction process of the item graph.

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

- We propose a model structure which captures sequential relationships, individual interest, and group interest of users from three different levels of granularity: session, item, and global, respectively. Multi-granularity user interest is used to enhance the result of the model prediction.
- We apply contrastive learning to fuse global representations and reduce noise interference during model construction, thereby enabling the model to obtain more contextual information and improve recommendation effectiveness.
- Extensive experiments on three real-world datasets have shown that our model achieves greater performance compared to the state-of-the-art methods in session-based recommendation tasks.

The remainder of this paper is organized as follows. Section 2 briefly discusses related work. Section 3 presents the methodology. In Sect. 4, we demonstrate the effectiveness of this method through experiments. Section 5 gives a conclusion.

2 Related Work

2.1 Session-Based Recommendation

The initial research on session-based recommendation primarily focuses on the temporal information of items in a session, and Markov chains have been widely applied [6,18]. FPMC [18] combines MCs with matrix factorization techniques to capture user sequence behaviors and preferences simultaneously. Fossil [6] addresses data sparsity issues and the long-tailed distribution problem in the datasets by fusing similarity-based methods with MCs. The models mentioned above only consider the most recent few interactions while predicting and fail to capture the transfer of user interest and higher-order information.

With the rise of deep learning techniques in fields such as computer vision and natural language processing, researchers started using RNNs for analyzing session data. Compared to conventional methods, RNN-based models offer superior learning capabilities and effectively extract data patterns within sequences. The GRU4REC [8] method is a typical representative, which introduces Gating Recurrent Units (GRUs) to session-based recommendation and yields good results. Later, data augmentation techniques [19] are used to further improve the recommendation performance by RNN-based models. NARM [12] proposes a hybrid encoder with an attention mechanism, which allows the model to focus on the most relevant items during the recommendation process. STAMP [13] employs simple multi-layer perception and attention networks to capture user interest and explicitly accounts for users' current behavior on their next action.

GNNs have recently gained significant attention in session-based recommendation tasks. Compared to previous methods, GNNs capture complex item transition relationships by the graph structure and offer improved accuracy in calculating item and session representations. SR-GNN [26] is a pioneering work in this area, employing Gated Graph Neural Networks (GGNNs) to learn the transition relationships of items in the session, and achieves promising results. GC-SAN [30] obtains local dependencies and long-range dependencies through GNNs and multi-layer self-attention networks, respectively. FGNN [17] uses a multi-head attention layer to help aggregate neighbor information by nodes with different weights. TAGNN [31] takes into account the diverse interest of users towards target items, thereby personalizing the recommendation task. GCE-GNN [24] utilizes a subtle approach to exploit the item transition relationships across all sessions to better infer the user preferences in the current session. SHARE [21] employs hypergraph attention networks to exploit item correlations within various contextual windows.

Most existing works analyze sessions primarily based on the sequential information of items in the session and do not comprehensively consider user interest from multi-granularity, which impedes the model performance.

2.2 Self-supervised Learning

Self-supervised learning is a type of unsupervised learning method where the label of positive and negative samples is marked through the inherent properties of the data itself, without manual intervention. Self-supervised learning can mainly be categorized into contrastive learning [4,9] and generative learning [3,15]. Generative learning, represented by auto-encoding [10], transforms the original data into vector representations using an encoder and then reconstructs it by a decoder. Contrastive learning usually consists of the paradigm of agent tasks combined with an objective function. Positive and negative sample pairs are automatically generated by agent tasks, and the Noise Contrastive Estimation (NCE) is used for the loss function. Notable examples of contrastive learning methods include SimCLR [2] and MoCo [5].

Self-supervised learning has also been introduced into sequence-related recommendation tasks. S^3-Rec [32] employs Mutual Information Maximization (MIM) principle to establish correlations among attributes, items, subsequences, and sequences. S^2-DHCN [29] first introduces self-supervised learning to session-based recommendation and applies contrastive learning on two hypergraph channels to improve recommendation performance. COTREC [28] explores the internal and external connectivity of sessions from two different perspectives. MGS [11] utilizes attribute information in sessions and adopts a contrastive learning strategy to reduce noise generated by neighboring items with similar attributes. Self-supervised learning can effectively alleviate the data sparsity issue that appeared in session-based recommendation.

3 The Proposed Method

In this section, we first formalize the definition of session-based recommendation. Then we present a detailed introduction to the proposed model. Figure 2 gives a graphic illustration of the problem definition. Figure 3 demonstrates the overall structure of the MUI-GNN model.

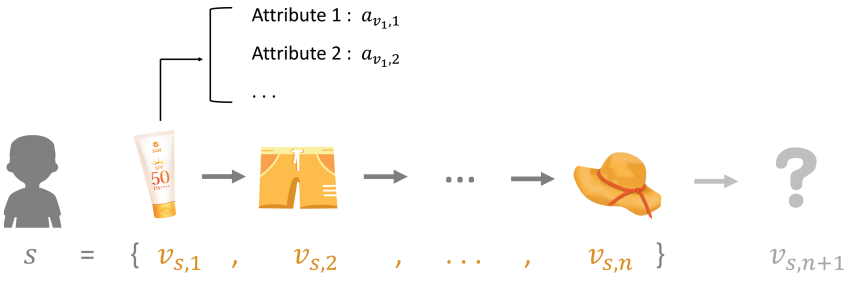

Fig. 2. A graphic illustration of a session-based recommendation task.

3.1 Problem Definition

Session-based recommendation tasks aim to predict the next item based on a user's limited historical interaction sequence. Here, we use $V = \{v_1, v_2, \ldots, v_{|V|}\}$ to represent the item set, where $|V|$ is the total number of the items. Each session is composed of several chronological items, denoted as $s = \{v_{s,1}, v_{s,2}, \ldots, v_{s,n}\}$, where $v_{s,i} \in V (1 \leq i \leq n)$ denotes the i-th item in session s, and n is the length of the session s. The embedding vector for each item v_i is represented as x_i. The ultimate goal of the recommendation model is to predict the next item $v_{s,n+1}$ by recommending top-K items under the given session s.

For each item v_i in the session, we label its attribute values as $\mathcal{A}_{v_i} = \{a_{v_i,1}, a_{v_i,2}, \ldots, a_{v_i,o}\}$, where $a_{v_i,j}$ $(1 \leq j \leq o)$ denotes the value of item v_i under the j-th attribute. The number of attribute types o varies from different datasets. By utilizing item attributes, we can find k neighbors which share the same attribute value with v_i. We formalize this item set as $\mathcal{N}_{v_i}$ for each attribute, $|\mathcal{N}_{v_i}| = k$.

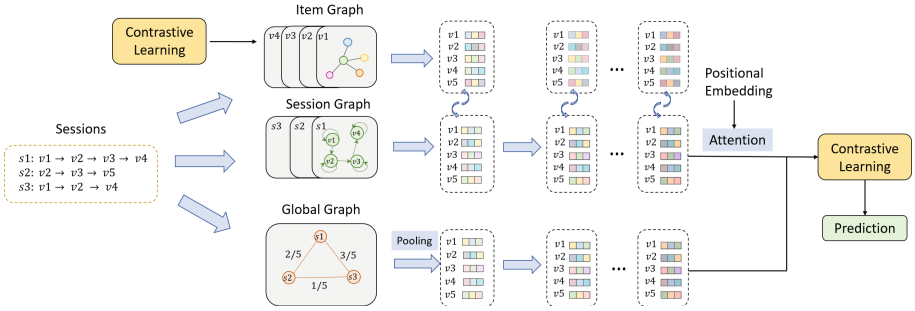

Fig. 3. The overall structure of MUI-GNN.

3.2 Session Graph

Session Graph Construction. Graph neural networks are applied to capture sequential information about items in a session. Specifically, we represent each session s as a directed graph $G_s = \{\mathcal{V}_s, \mathcal{E}_s\}$, where each node $v_{s,i} \in \mathcal{V}_s$ denotes an item in session s and each edge $e = (v_{s,i}, v_{s,i+1}) \in \mathcal{E}_s$ connects two adjacent items $v_{s,i}$ and $v_{s,i+1}$ in the session, representing the transition relationship between two items. A self-loop is added to each node to prevent information loss during the iterative process. The middle part in Fig. 3 shows the construction process of the session graph for session $s1$.

Session Graph Convolution. Graph convolution is operated to learn the transition relationships of adjacent items in the session. We use Graph Attention Networks(GAT) [20] to help each node learn the representation of the neighboring nodes. Specifically, for each item v_i in the session, the attention coefficients of each neighboring node are computed:

$$\alpha_{ij} = \frac{exp\left(LeakyReLU\left(e_{ij}^T\left(x_i^{(l-1)} \odot x_j^{(l-1)}\right)\right)\right)}{\sum_{v_k \in N_{v_i}} exp\left(LeakyReLU\left(e_{ij}^T\left(x_i^{(l-1)} \odot x_k^{(l-1)}\right)\right)\right)}, \tag{1}$$

where e_{ij} is the embedding of the relation between x_i and x_j, and l denotes the layer of graph convolution. $x_i^{(l-1)}$ is the embedding vector of item v_i in the previous layer. The representation $x_i^{(l)}$ in the l-th layer is then summed:

$$x_i^{(l)} = \sum_{x_j \in N_{x_i}} \alpha_{ij} x_j^{(l-1)}, \tag{2}$$

where $x_i^{(0)} = x_i$ in the first layer.

3.3 Item Graph

Item Graph Construction. Collecting items with similar attributes helps us better explore the individual interest of users at the item granularity. Thus we construct an item graph. To initialize the representations in the item graph, an attention mechanism is utilized to aggregate attribute information from the neighbors of item v_i:

$$\alpha_{ij} = \frac{exp\left(LeakyReLU\left(q_j^T\left(x_i || x_j\right)\right)\right)}{\sum_{x_k \in N_{v_i}} exp\left(LeakyReLU\left(q_j^T\left(x_i || x_k\right)\right)\right)}, \tag{3}$$

$$m_i = \sum_{x_j \in N_{x_i}} \alpha_{ij} v_j, \tag{4}$$

where m_i is the initial representation of the item v_i in the item graph. $||$ is the concatenation operator, and q_j is the embedding vector of the j-th attribute.

Item Graph Convolution. We update representations in item and session graphs together by a dual refinement method. First, we update x_i with m_i:

$$\beta_i = \frac{\left(W_1^s x_i^{(l)}\right)^T W_2^s m_i^{(l-1)}}{\sqrt{d}}, \tag{5}$$

$$x_i^{(l)} = x_i^{(l)} + \beta_i\left(m_i^{(l-1)} - x_i^{(l)}\right), \tag{6}$$

where d is the size of the embedding vector, $W_1^s, W_2^s \in \mathbb{R}^{d\times d}$ are learnable parameters, and l denotes the number of layers in the item graph convolution, $m_i^{(0)} = m_i$ in the first layer.

Then we use the updated $x_i^{(l)}$ to adjust $m_i^{(l)}$ by following attention mechanism,

$$\alpha_{ij} = \frac{exp\left(\left(W_1^m m_i^{(l-1)}\right)^T W_2^m x_j^{(l)}\right)}{\sum_{k=1}^{n} exp\left(\left(W_1^m m_i^{(l-1)}\right)^T W_2^m x_k^{(l)}\right)}, \tag{7}$$

$$m_i^{(l)} = \sum_{j=1}^{N} \alpha_{ij} m_j^{(l-1)}, \tag{8}$$

where $W_1^m, W_2^m \in \mathbb{R}^{d\times d}$ are also learnable parameters. Note that they are different from previous W_1^s and W_2^s. Finally, after the convolution of L layers, we get the final representation of the session items:

$$\pi_i = sigmoid\left(W_h\left[x_i^{(0)}||x_i^{(L)}\right]\right), \tag{9}$$

$$x_i^{(L)} = \pi_i x_i^{(0)} + (1-\pi) x_i^{(L)}, \tag{10}$$

where $W_h \in \mathbb{R}^{d\times 2d}$ is a learnable parameter.

3.4 Global Graph

Global Graph Construction. The group interest of users can be captured by mining similar behavior patterns across sessions. Specifically, we treat each session as a node in the global graph. Sessions are connected if there is at least one common item appears.

The weight of the edge is set to $W_{i,j}$, whose value is the size of the intersection set of items in the two sessions divided by the size of the union set. For example, in Fig. 3, the edge weight between session $s1$ and session $s2$ is $2/5$, as the intersection set is $\{v2, v3\}$, and the union set is $\{v1, v2, v3, v4, v5\}$.

Global Graph Convolution. The initial embedding $\theta^{(0)}$ of each session is the average of all item embedding x_i in the session, and then a convolution process is operated.

$$\theta^{(l+1)} = \hat{D}^{-1}\hat{A}\theta^{(l)}, \tag{11}$$

where $\hat{D}^{-1}$ is the inverse matrix of the degree matrix. At each convolution layer, $\theta^{(l)}$ learns different levels of cross-session information. Thus we average the embedding of each layer to obtain the final session representation.

$$\theta_f = \frac{1}{L+1}\sum_{l=0}^{L} \theta^{(l)}. \tag{12}$$

3.5 Prediction Layers

Session Representation. Intuitively, items at different positions in the session contain different semantic information, so we impose a positional encoding p_i on each item in the session:

$$h_i = x_i^{(L)} + p_i \ . \tag{13}$$

We use a soft attention mechanism to obtain the session representations fusing the user's individual interest and the last item.

$$\beta_i = g^T sigmoid \left(W_1 h_i + W_2 m_i^{(L)} + W_3 x_n^{(L)} + b \right), \tag{14}$$

$$z_s = \sum_{i=1}^{n} \beta_i hi \ , \tag{15}$$

where $W_1, W_2, W_3 \in \mathbb{R}^{d \times d}$ and $g, b \in \mathbb{R}^d$ are all learnable parameters. $m_i^{(L)}$ and $x_n^{(L)}$ denote the final result in the item graph and session graph convolution, respectively.

Finally, a gated mechanism is employed to reinforce the last behavior $x_n^{(L)}$ to the importance of the session representation explicitly.

$$\theta = sigmoid \left(W_4 \left[z_s || x_n^{(L)} \right] \right), \tag{16}$$

$$s_f = (1 - \mu\theta) \odot z_s + \mu\theta \odot x_n^{(L)}, \tag{17}$$

where $W_4 \in \mathbb{R}^{d \times 2d}$ is the learnable parameter. μ controls the weight of the gating unit. s_f is the session representation we eventually gain.

Prediction. We implement the inner product of the embedding x_i of the item and the final representation s_f of the session to get a score $\hat{z}_i$ for each item.

$$\hat{z}_i = s_f^T x_i \ . \tag{18}$$

A softmax function is used to calculate the probability $\hat{y}$ that the item will be recommended.

$$\hat{y} = softmax \left(\hat{z}_i \right) \ . \tag{19}$$

The model is optimized by a cross-entropy loss function, which is also commonly used in recommender systems.

$$\mathcal{L}_r \left(\hat{y} \right) = -\sum_{i=1}^{N} y_i \log \left(\hat{y}_i \right) + \left(1 - y_i \right) log \left(1 - \hat{y}_i \right) \ , \tag{20}$$

where y_i is the one-hot representation of the item x_i being recommended ground truth.

Self-supervised Learning. To further enhance the feature representation, we also set up self-supervised auxiliary tasks to optimize the model, which is primarily used to learn feature information across sessions. We use InfoNCE [5] with a standard binary cross-entropy loss function on positive and negative samples.

$$\mathcal{L}_{SSL} = -\log\sigma\left(f_D\left(\theta_i^h,\ \theta_i^l\right)\right) - \log\sigma\left(1 - f_D\left(\widetilde{\theta}_i^h,\ \theta_i^l\right)\right). \tag{21}$$

The final loss function of the model is:

$$\mathcal{L} = \mathcal{L}_r + \phi\mathcal{L}_{SSL_1} + \beta\mathcal{L}_{SSL_2}\ , \tag{22}$$

where ϕ and β are hyper-parameters controlling the weights of $\mathcal{L}_{SSL_1}$ and $\mathcal{L}_{SSL_2}$, respectively, which are shown as contrastive learning modules in Fig. 3.

4 Experiments

In this section, we first describe the experimental settings. Then a series of evaluations of model performance are conducted by answering the following questions:

RQ1: Does the MUI-GNN model surpass state-of-the-art session-based recommendation baseline models on several real-world datasets?

RQ2: Does capturing user interest from item and global granularity effectively enhance the performance of our model?

RQ3: How do key parameters, such as GNN layer number and embedding size, affect model performance?

4.1 Experimental Settings

Datasets. We conduct experiments on three real-world datasets, namely Diginetica, 30music, and Tmall, from different fields and with different data sparsity. All datasets contain session sequences and attribute information of items. Diginetica is a personalized e-commerce search challenge dataset in the CIKM Cup 2016 competition. Here, referring to [17,26,30], we only use its transaction data. 30music is a dataset collected and extracted through the Last. fm API, which contains user music playback data and divides playback events into different sessions. Tmall records the purchase logs of anonymous users on the Tmall website during the first six months of "Double Eleven" and the day of "Double Eleven" from the IJACI-15 competition.

We process the datasets in the same way as [26,29]. We remove sessions of length 1 and items that appear less than 5 times in the dataset. To enrich the training and testing data, we subdivide each session into several sub-sessions. Specifically, on each dataset, from a session $s = \{v_1,\ v_2,\ v_3,\ \ldots,\ v_n\}$, we generate new sequences $([v_1],\ v_2)$, $([v_1,\ v_2],\ \ v_3)$, $\ldots$, $([v_1,\ v_2,\ \ldots,\ v_{n-1}],\ v_n)$ for both training and testing sets. Similar to [26,29], sessions with interactions in the past week are used as the testing data, while other sessions are used as the training data. Table 1 shows the statistical information of three datasets after preprocessing.

Table 1. Statistical details of datasets.

Datasets	Diginetica	30music	Tmall
train sessions	719,470	1,153,622	351,268
test sessions	60,858	122,517	25,898
clicks	982,961	1,429,251	443,479
items	43,097	132,647	40,727
average length	4.85	9.33	6.69

Baseline Methods. We compare our model with the following classical and state-of-the-art models:

FPMC [18]: It includes Markov Chains and Matrix Factorization models to capture user interest and interaction sequence.

GRU4REC [8]: It utilizes session parallel mini-batches and a ranking loss function to enable GRUs to study sequence behaviors.

NARM [12]: It combines RNNs with an attention mechanism to understand interactions in a session.

SR-GNN [26]: It is the first model to apply GNNs to session-based recommendation, utilizing GGNNs and a soft attention mechanism to learn the representation of items.

GCE-GNN [24]: It uses a subtle approach to combine local and global item transition relationships.

S^2-DHCN [29]: It devises two different hypergraph channels to study inter-session and cross-session information, applying self-supervised learning to enhance recommendation.

MGS [11]: It constructs a mirror graph based on item attribute information and employs an iterative dual refinement mechanism to transfer data.

Evaluation Metrics. Similar to previous works [11,29], we measure model performance through two widely used metrics in session-based recommendation tasks: P@K and MRR@K.

P@K (Precision) indicates whether ground truth is at the top-K position of the prediction list:

$$P@K = \frac{1}{|U|}\sum_{v \in U}\prod(R_v < K), \tag{23}$$

where $|U|$ is the size of testing set, R_v represents the position of ground truth item v in the top K of the prediction list.

MRR@K (Mean Reciprocal Rank) measures the position of ground truth in the top-K recommendation list. Compared to P@K, MRR@K considers the impact of the result order.

$$MRR@K = \frac{1}{|U|}\sum_{v \in U}\frac{1}{R_v}. \tag{24}$$

Hyper-parameters Settings. According to [26,30], the size of each mini-batch is selected based on the size of the hidden vectors. To ensure a fair comparison, we quote the experimental results of baseline models reported in the original paper as possible, which is in their best hyper-parameter settings. All parameters follow a Gaussian distribution with an average value of 0 and a standard deviation of 1. The L2 norm value of the model is set to 10^{-5} and uses the Adam optimizer. The initial learning rate is 0.001, and there is a decay rate of 0.1 after the first three epochs.

Table 2. Performance Comparison of Different Methods on Diginetica dataset.

Method	Diginetica			
	P@10	MRR@10	P@20	MRR@20
FPMC	15.43	6.20	26.53	6.95
GRU4REC	17.93	7.33	29.45	8.33
NARM	35.44	15.13	49.70	16.17
SR-GNN	36.86	15.52	50.73	17.59
GCE-GNN	41.54	<u>18.29</u>	54.64	<u>19.20</u>
S^2-DHCN	41.16	18.15	53.18	18.44
MGS	<u>41.80</u>	18.20	<u>55.05</u>	19.13
Ours	**42.17**	**18.29**	**55.67**	**19.23**

Table 3. Performance Comparison of Different Methods on 30music dataset.

Method	30music			
	P@10	MRR@10	P@20	MRR@20
FPMC	1.51	0.55	2.40	0.61
GRU4REC	15.91	10.46	18.28	10.95
NARM	37.81	25.95	39.40	26.55
SR-GNN	36.49	26.71	39.93	26.94
GCE-GNN	39.93	21.21	44.71	21.55
S^2-DHCN	40.05	17.58	45.49	17.79
MGS	<u>41.51</u>	<u>27.67</u>	<u>46.46</u>	<u>28.01</u>
Ours	**42.04**	**28.58**	**47.26**	**28.89**

Table 4. Performance Comparison of Different Methods on Tmall dataset.

Method	Tmall			
	P@10	MRR@10	P@20	MRR@20
FPMC	13.10	7.12	16.06	7.32
GRU4REC	9.47	5.78	10.93	5.89
NARM	19.17	10.42	23.30	10.70
SR-GNN	23.41	13.45	27.57	13.72
GCE-GNN	29.19	15.55	34.35	15.91
S^2-DHCN	26.22	14.60	31.42	15.05
MGS	<u>35.39</u>	<u>18.15</u>	<u>42.12</u>	<u>18.62</u>
Ours	**36.33**	**18.42**	**42.69**	**18.85**

4.2 Experimental Results (RQ1)

Tables 2 to 4 show the performance of our model and baselines on three real-world datasets. The best and second-best results are highlighted in boldface and underlined, respectively. From the above tables, we can obtain the following observations:

RNN-based models (e.g. GRU4REC, NARM) utilize deep learning methods to capture item transition relationships in sessions, significantly improving model performance compared to earlier methods (e.g. FPMC), indicating the importance of utilizing sequential information in sessions. Among them, NARM performs better than GRU4REC because it uses an attention mechanism to model user interaction and purpose distinctively.

GNN-based methods demonstrate superior performance compared to RNN-based models, reflecting GNNs' ability to model sequential information and understand complex item transition relationships. The SR-GNN model, as it does not use any information out of session sequence, has a relatively lower performance than GCE-GNN, which combines global information to model user interest. S^2-DHCN uses the strategy of self-supervised learning combined with the hypergraph structure of multiple channels. MGS uses the attributes of items to build mirror graphs to help learn the representation information of items. Their improvement in model results supports that the application of auxiliary information plays an important role in session-based recommendation.

Our proposed MUI-GNN shows superior results on all three datasets compared to these traditional and deep learning approaches mentioned above, suggesting that considering multi-granularity user interest with sequential relationships in sessions can effectively enhance the model performance in session-based recommendation tasks.

4.3 Model Analysis and Discussion (RQ2 and RQ3)

Impact of Item/Global Graph. To explore the distinctive influence of multi-granularity user interest on prediction results, we adjust the MUI-GNN model to verify the effectiveness of the item graph and global graph respectively. Ablation experiments are conducted on Diginetica and 30music datasets. Table 5 demonstrates the results of the experiments, with the best performance highlighted in boldface.

Table 5. Impact of item/global graph

Method	Diginetica		30music	
	P@20	MRR@20	P@20	MRR@20
MUI-GNN	**55.78**	**19.21**	**47.26**	**28.89**
MUI-GNN w/o global	54.97	19.06	46.49	28.69
MUI-GNN w/o item	55.32	18.73	44.38	**28.89**

In the table, MUI-GNN w/o global and MUI-GNN w/o item indicates the MUI-GNN model without global graph/ item graph, respectively. We find that model's performance is optimal when the MUI-GNN model contains both the item and global graph modules, which captures the individual and group interest of users simultaneously. In general, the absence of either module will cut down the performance in recommendation results.

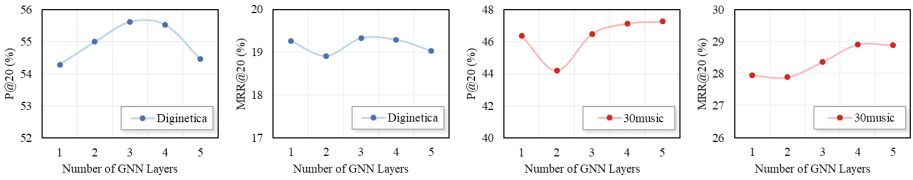

Fig. 4. Impact of depths in GNNs.

Impact of Depths in GNNs. We set the item graph and global graph into the same layer number and tested the performance of models with different depths over two datasets. The layer number of the graph neural networks in our model is set to {1, 2, 3, 4, 5}.

As can be seen from Fig. 4, in the Diginetica dataset, it is reasonable to set the depth of the graph neural networks to 3 or 4, as an over-smoothing problem will arise when the networks get too deep. While in the 30music dataset, increasing layers of graph neural networks can further improve the performance of the model. A possible reason is the average length of the 30music dataset is nearly as double as of Diginetica, which shows that the data pattern in the 30music dataset is more complex and requires deeper networks to mine the users' individual and global interest.

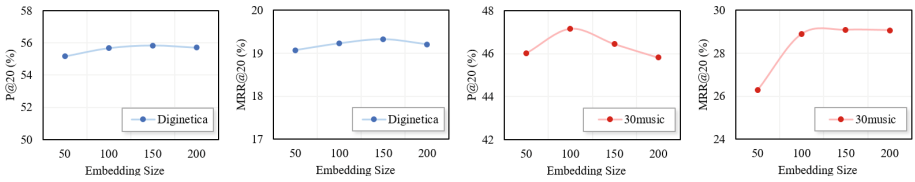

Fig. 5. Impact of embedding size.

Impact of Embedding Size. We test the embedding size in {50, 100, 150, 200} of the MUI-GNN model. Results are shown in Fig. 5. When the embedding vectors rise to the size of 100, our model achieves considerable performance. However, the continuous increase of embedding size will not always lead to better results, as an overly large embedding size may cause massive computation and overfitting problems.

5 Conclusion

The short and anonymous nature of sessions brings great challenges to session-based recommendation tasks. In this paper, we propose the MUI-GNN model, which not only incorporates the sequential relationships in a session but also explores users' interest at the item and global granularity to enhance model performance. Existing models often fail to capture this valuable auxiliary information comprehensively. Contrastive learning methods are used to reduce the noise during the graph construction and help the model learn the contextual information of the session better. Extensive experiments conducted on three real-world datasets exhibit the effectiveness and superiority of our model considering multi-granularity user interest.

References

1. Chang, J., Gao, C., He, X., Jin, D., Li, Y.: Bundle recommendation with graph convolutional networks. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 1673–1676 (2020)
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. PMLR (2020)
3. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
4. Gidaris, S., Singh, P., Komodakis, N.: Unsupervised representation learning by predicting image rotations. In: International Conference on Learning Representations (2018)
5. 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)

6. He, R., McAuley, J.: Fusing similarity models with Markov chains for sparse sequential recommendation. In: 2016 IEEE 16th International Conference on Data Mining (ICDM), pp. 191–200. IEEE (2016)
7. He, X., Deng, K., Wang, X., Li, Y., Zhang, Y., Wang, M.: LightGCN: simplifying and powering graph convolution network for recommendation. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 639–648 (2020)
8. Hidasi, B., Karatzoglou, A., Baltrunas, L., Tikk, D.: Session-based recommendations with recurrent neural networks. arXiv preprint arXiv:1511.06939 (2015)
9. Jaiswal, A., Babu, A.R., Zadeh, M.Z., Banerjee, D., Makedon, F.: A survey on contrastive self-supervised learning. Technologies **9**(1), 2 (2020)
10. Kingma, D.P., Welling, M.: Auto-encoding variational Bayes. arXiv preprint arXiv:1312.6114 (2013)
11. Lai, S., Meng, E., Zhang, F., Li, C., Wang, B., Sun, A.: An attribute-driven mirror graph network for session-based recommendation. In: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 1674–1683 (2022)
12. Li, J., Ren, P., Chen, Z., Ren, Z., Lian, T., Ma, J.: Neural attentive session-based recommendation. In: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, pp. 1419–1428 (2017)
13. 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 International Conference on Knowledge Discovery & Data Mining, pp. 1831–1839 (2018)
14. Liu, X., et al.: Self-supervised learning: generative or contrastive. IEEE Trans. Knowl. Data Eng. **35**(1), 857–876 (2021)
15. Van den Oord, A., Kalchbrenner, N., Espeholt, L., Vinyals, O., Graves, A., et al.: Conditional image generation with PixelCNN decoders. In: Advances in Neural Information Processing Systems, vol. 29 (2016)
16. Pan, Z., Cai, F., Chen, W., Chen, H., De Rijke, M.: Star graph neural networks for session-based recommendation. In: Proceedings of the 29th ACM International Conference on Information & Knowledge Management, pp. 1195–1204 (2020)
17. Qiu, R., Li, J., Huang, Z., Yin, H.: Rethinking the item order in session-based recommendation with graph neural networks. In: Proceedings of the 28th ACM International Conference on Information and Knowledge Management, pp. 579–588 (2019)
18. Rendle, S., Freudenthaler, C., Schmidt-Thieme, L.: Factorizing personalized Markov chains for next-basket recommendation. In: Proceedings of the 19th International Conference on World Wide Web, pp. 811–820 (2010)
19. Tan, Y.K., Xu, X., Liu, Y.: Improved recurrent neural networks for session-based recommendations. In: Proceedings of the 1st Workshop on Deep Learning for Recommender Systems, pp. 17–22 (2016)
20. Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y., et al.: Graph attention networks. Stat **1050**(20), 10–48550 (2017)
21. Wang, J., Ding, K., Zhu, Z., Caverlee, J.: Session-based recommendation with hypergraph attention networks. In: Proceedings of the 2021 SIAM International Conference on Data Mining (SDM), pp. 82–90. SIAM (2021)
22. Wang, S., Cao, L., Wang, Y., Sheng, Q.Z., Orgun, M.A., Lian, D.: A survey on session-based recommender systems. ACM Comput. Surv. (CSUR) **54**(7), 1–38 (2021)

23. Wang, X., Wang, R., Shi, C., Song, G., Li, Q.: Multi-component graph convolutional collaborative filtering. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 6267–6274 (2020)
24. Wang, Z., Wei, W., Cong, G., Li, X.L., Mao, X.L., Qiu, M.: Global context enhanced graph neural networks for session-based recommendation. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 169–178 (2020)
25. Wu, S., Sun, F., Zhang, W., Xie, X., Cui, B.: Graph neural networks in recommender systems: a survey. ACM Comput. Surv. **55**(5), 1–37 (2022)
26. 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, vol. 33, pp. 346–353 (2019)
27. Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., Philip, S.Y.: A comprehensive survey on graph neural networks. IEEE Trans. Neural Netw. Learn. Syst. **32**(1), 4–24 (2020)
28. Xia, X., Yin, H., Yu, J., Shao, Y., Cui, L.: Self-supervised graph co-training for session-based recommendation. In: Proceedings of the 30th ACM International Conference on Information & Knowledge Management, pp. 2180–2190 (2021)
29. Xia, X., Yin, H., Yu, J., Wang, Q., Cui, L., Zhang, X.: Self-supervised hypergraph convolutional networks for session-based recommendation. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, pp. 4503–4511 (2021)
30. Xu, C., et al.: Graph contextualized self-attention network for session-based recommendation. In: IJCAI, vol. 19, pp. 3940–3946 (2019)
31. Yu, F., Zhu, Y., Liu, Q., Wu, S., Wang, L., Tan, T.: Tagnn: target attentive graph neural networks for session-based recommendation. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 1921–1924 (2020)
32. Zhou, K., et al.: S3-rec: Self-supervised learning for sequential recommendation with mutual information maximization. In: Proceedings of the 29th ACM International Conference On Information & Knowledge Management, pp. 1893–1902 (2020)