Neural Processing Letters (2024) 56:3
https://doi.org/10.1007/s11063-024-11452-z

SR-MACL: Session-Based Recommendation with Multi-layer Aggregation Augmentation in Contrastive Learning

Shiwei Gao[1] · Yufeng Zeng[1] · Xiaochao Dang[1] · Xiaohui Dong[1]

Accepted: 12 December 2023 / Published online: 5 February 2024

Abstract
Session-based recommendation (SBR) aims to predict the next item of interest in chronological order based on a given sequence of short-term behaviour of anonymous users. Due to the limited data available for short-term user interactions, its performance is more susceptible to data sparsity problems than traditional recommendation methods. Contrastive learning is often used to solve the data sparsity problem due to its ability to extract general features from the raw data. Existing session-based recommendation methods based on graph contrastive learning typically build graph contrastive learning by using information from other sessions to generate augmented views. While this avoids the problem that the use of dropout in traditional contrast learning methods can cause damage to the session context, it inevitably introduces irrelevant item information, which interferes with accurately modelling user interests and leads to sub-optimal model performance. To address these issues, we propose a new session recommendation method based on multi-layer aggregation augmentation contrastive learning, namely SR-MACL. In SR-MACL we construct a contrastive view by adding noise to the embedding representation and forming a contrastive embedding representation by multi-layer aggregation, which not only effectively solves the problem that traditional graph enhancement methods can destroy the context of the whole session, but also avoids the interference of irrelevant items. Experimental results on three real datasets have shown that SR-MACL can improve the accuracy of recommendation results and predict the user's next interaction more effectively.

Keywords Session-based recommendation · Graph contrastive learning · Representation learning

1 Introduction

Recommender systems are widely used for short video recommendations, online shopping and news recommendations due to their ability to predict users' interests based on their historical behaviour. Most existing recommendation systems rely on users' personal information

✉ Shiwei Gao
gaoshiwei@nwnu.edu.cn

[1] College of Computer Science and Engineering, Northwest Normal University, Lanzhou 730070, China

and long-term historical behaviour data as input features, use machine learning and deep learning related techniques to predict what users are interested in. However, in many modern online platforms, capturing a user's historical browsing behaviour results in a performance loss and the preference of the user's current session can easily be overwhelmed by the long history of behaviour. Consequently, session recommendation systems have been created to improve the user experience. As a sub-task of the recommendation system, the advantage of the session recommendation system is that it relies only on the user's click behaviour for the current session. The only input features that can be used for session recommendation are the user and item ID information, which makes it more difficult to capture the user's interest. Session recommendation aims to predict items of interest to the user in chronological order based on a given sequence of short-term behaviour of anonymous users.

In early research, Markov chain-based approaches were first proposed for session recommendation [1]. Rendle et al. [1] combined matrix factorization with Markov chains to capture the interests of users. With the advancement of deep learning techniques, many methods based on deep learning have been proposed for session recommendation, which are mainly based on recurrent neural networks (RNNs), attention networks and their intricate fusion [2–4]. Since sequences in session recommendation have other more complex transition relationships in addition to simple time-dependent relationships, extracting the transition relationships in session sequences only in temporal order is not sufficient for session recommendation. Methods based on graph neural networks (GNNs) have demonstrated the effective of capturing the complex transition relationships of items in a given session [5–8]. GNNs have a strong ability to model the dependencies between nodes in a graph. The GNN-based recommendation model differs from previous models in that it models session sequences as session graphs and employs graph encoders to further mine the rich hidden information between items in the session graph to obtain good prediction results. Although the graph neural network-based approach can more accurately model the transformation relationships of items in a session sequence, the GNN-based recommendation model is not as accurate as the previous model. However, GNN-based recommendation models still face the problem of data sparsity. Due to the lack of long-term historical user behaviour data, session-based recommendations can only be made using user interaction records generated from short-term sessions, but the number of user interactions in a session is very limited and far less than long-term user behaviour data. This lack of available data prevents session-based recommendation models from learning accurate user preferences, resulting in sub-optimal recommendation performance.

Graph comparison learning uses the intrinsic relationships of the data itself and learns the features of the data itself through different augmented views, without relying on manually annotated data, which has a great advantage in solving the data sparsity problem. Most of the current recommendation models based on graph contrastive learning adopt random node or edge dropping to generate contrast views. However, since the session sequences in session-based recommendation are extremely sparse, randomly dropping nodes or edges is likely to disrupt the current session context, therefore, the traditional data augmentation methods of graph contrastive learning are not suitable for session-based recommendation. Existing graph contrastive learning methods for session-based recommendation focus on utilizing information from other sessions to help generate different views [9–12]. S2-DHCN employed hypergraphs to generate two enhanced views, constructed global graphs to extract information from other sessions, and utilized contrastive learning as an auxiliary task to improve the recommendation performance of the main task [9]. COTREC proposed a framework based on contrastive learning to enhance the accuracy of session recommendations and used information from other sessions to generate session views and item views for contrastive learning [10]. SimCGNN increased the diversity of sessions with the same last item by using them as negative samples and designing a contrastive module based on cosine similarity [11]. While these methods successfully prevent the corruption of session context information caused by traditional graph augmentation techniques, the incorporation of additional session information may introduce items that are irrelevant or even contradictory to the user's current interests. This interference impedes the accurate modeling of user interests, ultimately leading to suboptimal performance. A series of papers in collaborative filtering recommendation that investigate whether data augmentation in contrastive learning is necessary [13–15]. They believe that the effectiveness of graph contrastive learning in recommendation tasks is mainly due to the contrastive loss. Therefore, they proposed several simple but effective data augmentation methods. Although these studies are not specifically tailored to the field of session-based recommendation, they have great reference value and can provide valuable insights for further exploration in session-based recommendation.

To address the aforementioned issues, we propose a new session-based recommendation model, called Session-based Recommendation with Multi-layer Aggregated Contrastive Learning (SR-MACL). We not only abandon the graph augmentation methods of randomly dropping nodes or edges that are frequently used in traditional contrastive learning, but also do not generate enhanced views by introducing information from other sessions. We just use a simple and effective noise-based multi-level aggregated embedding enhancement to create contrastive views. In our model, two views share initial embeddings and adjacency matrices. Then, the complex transition patterns of the session sequence are modelled by stacking Star Graph Neural Networks (SGNN). Based on the Gated Graph Neural Network, a star node is added to consider non-adjacent items to solve the problem of long-distance transition information propagation. Uniform noise is added to the learned representations at each layer. We then generate a new contrastive view by aggregating the representations of each layer, thereby achieving more effective representation-level data augmentation without disrupting the context of the session sequence. Through contrastive learning, we maximize the mutual information between the learned session representations of the two session embeddings to improve the performance of item/session feature extraction. Then, we unify the recommendation task and self-supervised task under one framework through multi-task learning. By jointly optimizing these two tasks, we learn more robust embedding representations to accurately predict the next item that the user is interested in. The main contributions of this paper are as follows:

(1) We introduce noised-based contrastive learning to alleviate the data sparsity problem in session-based recommendation.
(2) We propose a novel multi-layer aggregated contrastive learning method that can avoid the influence of irrelevant items when introducing information from other sessions, achieve more effective representation-level data augmentation, and provide a new perspective on how to apply graph contrastive learning to session-based recommendation.
(3) The experimental results show that our model outperforms the state-of-the-art baseline models and demonstrates significant performance improvements.

The remaining parts of this paper are organized as follows. Section 2 introduces the related work of contrastive learning and session-based recommendation. Section 3 details the implementation of our SR-MACL model. Then, Sect. 4 presents a series of experiments, including performance comparisons between our proposed SR-MACL model and other baseline models, and demonstrates the effectiveness of our model. At last, we summarize our current work and looks forward future work in Sect. 5.

2 Related Work

We provide an overview of related research on session-based recommendation, primarily categorized into four areas: traditional methods, deep learning-based methods, GNN-based methods, and GCL (graph contrastive learning)-based methods. We then focus on research related to GNN-based and GCL-based, as these are most closely related to our study.

2.1 Session Recommendation

In earlier studies, Markov chain-based approaches convert the session sequence into a Markov chain, and subsequently predict the following action of the user based on their prior action. FPMC captures sequential patterns and long-term user preferences by combining matrix factorization and first-order Markov chains [2]. However, the Markov chain-based approach only employs the relationship of adjacent items to model the sequential transformation of session sequences and does not consider the connection between disjoint items. Subsequent research has shifted its focus to using deep learning techniques to learn transition relationships between items. For example, GRU4REC utilized GRU to model session sequences. On the other hand, techniques based on recurrent neural networks possess notable sequential assumptions, which fail to encapsulate the transitive connections between items that are widely separated in the session sequence. In addition, the attention mechanism has been widely used in SBR [3], which can be used to distinguish items in the session based on their importance. It also can be combined with other methods such as RNN to emphasize user's main intention [16, 17]. Regarding modelling sessions as graph structured data, SR-GNN was the pioneer method [5]. This approach transforms a session sequence into an unweighted directed graph and utilizes the gated graph neural network (GGNN) to comprehend intricate item transitions within a session [18]. NISER pointed out the presence of popularity bias in GNN-based recommendation models and provided evidence that this issue is partly associated with the size or norm of the learned item and session graph representations (embedding vectors). And proposed a training procedure to alleviate this problem by employing normalized representations [19]. This approach has been employed in a subsequent series of GNN-based models. GCE-GNN acquires knowledge from two levels of item representations obtained from the session graph and the global graph correspondingly [6]. The session graph is formed using the current session, while the global graph is formed by consolidating all item sequences and their adjacent items. SGNN-HN argued that previous methods ignore information from items that are not directly connected and suffer from the commonly observed overfitting problem in graph neural networks [7]. Therefore, the star graph neural network (SGNN) is used to learn the complex transition relationships between items in the session sequence. To avoid overfitting, the highway network (HN) is used to select embeddings from item representations in the form of weights. GC-HGNN employed hypergraphs to construct global graphs and obtain global contextual information through hypergraph convolution [8]. The graph attention network is employed by GC-HGNN to incorporate local information, while the attention mechanism is utilized to handle merged features and learn the final representation of the session sequence.

2.2 Contrastive Learning in Recommender Systems

Contrastive learning has shown impressive performance in computer vision and natural language processing. Recently, it has gained traction in many fields of artificial intelligence, including graph neural networks, leading to a series of significant advances [20–24]. Due to the ability of contrastive learning to learn general features from unlabelled data, contrastive learning is an effective method to address the problem of data sparsity [25–27], making it a popular research direction in the field of recommender systems. Currently, many graph contrastive learning recommendation models have been proposed, and significant effects have been achieved [9, 11, 12, 28–31].

SGL generated different augmented views through node or edge dropout and random walks [28], and then maximized the consistency of the representations learned by the graph encoder under different views. DCRec proposed a new debiased recommendation contrastive learning paradigm (DCRec) [29], which incorporates global information into the augmented views through adaptive perceptual augmentation. The paradigm combines sequence pattern encoding with modelling of global collaborative relationships through adaptive consistency perception enhancement. CL4SRec conducted contrastive learning by generating different data augmentations based on sequence construction through item cropping, masking, and re-ordering [30]. S2-DHCN used hypergraph to generate two augmented views, constructed a global graph to extract information from other sessions, and employed contrastive learning as an auxiliary task to improve the recommendation performance of the main task [9]. COTREC proposed a contrastive learning-based framework to enhance the accuracy of session recommendation and utilized information from other sessions to generate session views and item views for contrastive learning [10]. SimCGNN increased the specificity of sessions with the same last item by using sessions with the same last item as negative samples and designed a contrast module based on cosine similarity to enhance the difference between sessions with the same last item [11]. CGL employed the self-supervised module that combines global and session graphs, decoupled the current session's intention, enriched item representations, and designed a label confusion method to prevent overfitting [12]. Existing graph contrastive learning session-based recommendation methods focus on introducing information from other sessions to generate the global graph to construct augmented views. However, introducing other session information may introduce items that are irrelevant or even opposite to the user's current interests, which interferes with accurately modelling the user's interests.

3 Methodology

This section commences by elucidating the fundamental concepts of session-based recommendation and graph construction. Subsequently, we explicate our model in detail, the overall architecture is shown in Fig. 1.

3.1 Problem Definition

Session-based recommendation primarily aims to provide next-item recommendations for anonymous users. Therefore, the model needs to accurately capture the user's general interest based on the session sequence. Session-based recommendation systems aim to provide attractive recommendations for anonymous users, so the model must meticulously apprehend the user's interests by relying on short sessions rather than the complete historical record of interactions.

Assuming there are m items and n sessions, let $\mathrm{V} = \{\mathrm{v}_1, \mathrm{v}_2, \ldots, \mathrm{v}_\mathrm{m}\}$ and $\mathrm{S} = \{\mathrm{s}^1, \mathrm{s}^2, \ldots, \mathrm{s}^\mathrm{n}\}$ represent the sets of items and sessions, respectively. s^i represents the i-th session. Each session s^i is an ordered sequence of clicks arranged in chronological order $\left[\mathrm{v}^\mathrm{i}_1, \mathrm{v}^\mathrm{i}_2, \ldots, \mathrm{v}^\mathrm{i}_\mathrm{k}\right]$, where $\mathrm{v}^\mathrm{i}_\mathrm{j} \in \mathrm{V}$ represents the j-th clicked item in the i-th session, and k represents the length of the session. Our goal is to predict the next item $\mathrm{v}^\mathrm{i}_{\mathrm{k}+1}$ for any session s^i.

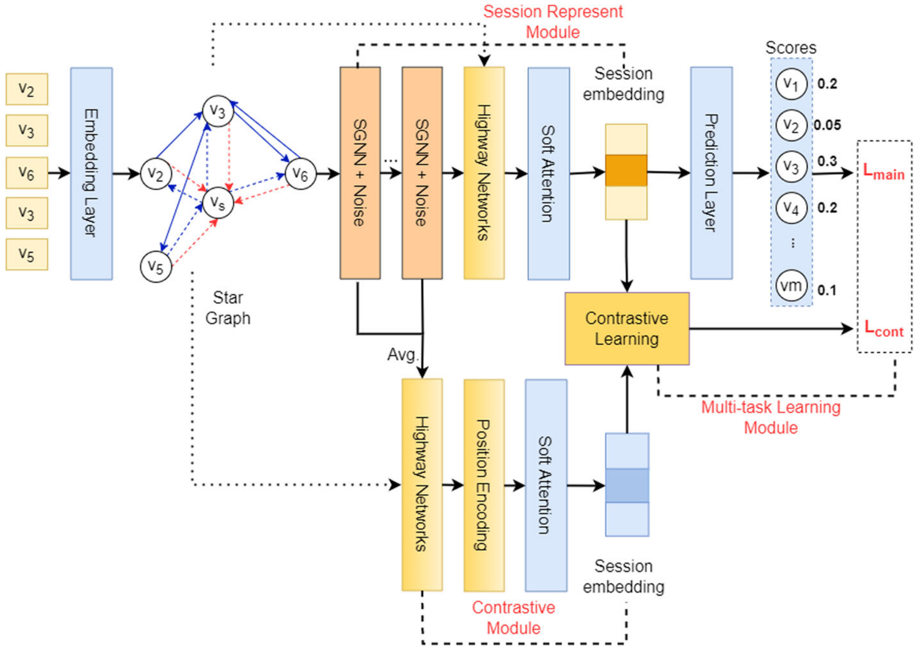


Fig. 1 Overview of the SR-MACL model

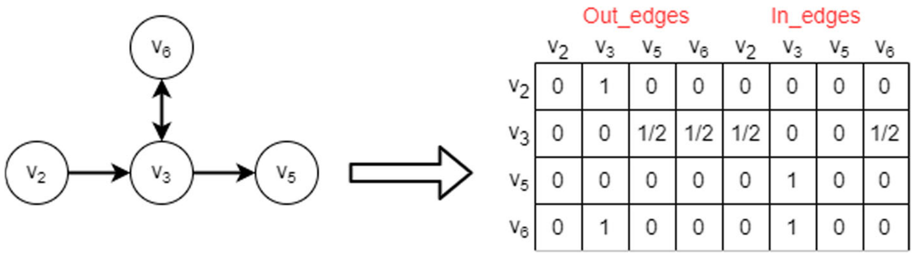
The process involves creating a session representation using the item representations within the session. From this, probabilities are determined by measuring the similarity between the session representation and all item embeddings. Finally, the model performs top-N recommendation based on these probabilities.

3.2 Graph Construction

For each session sequence S, we can model it as a session graph $G = (V, E)$, where V represents the nodes in session $S = \left\{\left\{v_1^s, v_2^s, ..., v_m^s\right\}, v_s\right\}$, v_s represents the star node, and the edge $\left(v_i^s, v_{i+1}^s\right)$ represents items clicked at two adjacent time points in a session. First, we divide the edges into input and output edges and assign a normalized weight, which is calculated as the occurrence frequency of the edge divided by the out-degree of the starting node of the edge. An example of the graph construction is shown in Fig. 2. Inspired by [32], we treat the items in the session sequence as satellite nodes, and we add a star node v_s to capture long-range dependencies between non-adjacent nodes. The edges between satellite nodes are unidirectional, while the edges between star nodes and satellite nodes are bidirectional. The construction of the star graph is shown in Fig. 1.

3.3 Model Overview

In this section, we propose SR-MACL, a session-based recommendation model based on multi-layer aggregation enhanced contrastive learning. The architecture of SR-MACL is shown in Fig. 2. We divide the model into three modules, namely the session representation module, multi-layer aggregation contrast module, and multi-task learning module. We first



	Out_edges				In_edges			
	v2	v3	v5	v6	v2	v3	v5	v6
v2	0	1	0	0	0	0	0	0
v3	0	0	1/2	1/2	1/2	0	0	1/2
v5	0	0	0	0	0	1	0	0
v6	0	1	0	0	0	1	0	0

Fig. 2 An illustration for the construction of the session graph

give a brief overview, and later sections will describe these three modules in detail. The session representation module is used to generate the session embedding for the main task. The contrastive module is used to generate the session embedding for the auxiliary task and through contrastive learning achieves information exchange between the two session embeddings. Finally, the multi-task learning module combines the two tasks to jointly optimize the model.

3.4 Session Representation Module

3.4.1 Initialization

Before learning item representations, we need to encode all items in V into a unified embedding space R^d, where d represents the size of the embedding dimension. We embed each session S and item v_i in the same space. Note that the initialization of satellite nodes and star nodes is different. We directly use the embedding of the unique item in the session as the representation of the satellite node:

$$h^0 = [v_1, v_2, v_3, \ldots, v_k] \tag{1}$$

Here, $v_i \in R^d$ represents the d-dimensional embedding of the satellite node i in the star graph. The initial embedding of the star node is obtained by averaging the initial embeddings of the satellite nodes:

$$v_s^0 = \frac{1}{k}\sum\nolimits_{i=1}^{k} v_i \tag{2}$$

3.4.2 Item Embedding Learning

We employ Star Graph Neural Network (SGNN) to learn the satellite nodes in the star graph, mainly updating the satellite node embeddings by propagating information from neighbouring nodes and star nodes.

First, we consider the information from the neighbouring nodes. For the satellite nodes v_i in the star graph, the update function is shown as follows:

$$a_{s,i}^{l} = A_{s,i:}\left[v_1^{(l-1)}, \cdots, v_k^{(l-1)}\right]^T W + b \tag{3}$$

$$z_{s,i}^{l} = \sigma\left(W_z a_{s,i}^{l} + U_z v_i^{(l-1)}\right) \tag{4}$$

$$\mathrm{r}_{\mathrm{s,i}}^{\mathrm{l}} = \sigma\left(\mathrm{W_r a_{s,i}^l} + \mathrm{U_r v_i^{(l-1)}}\right) \tag{5}$$

$$\widetilde{\mathrm{v_i^l}} = \tanh\left(\mathrm{W_o a_{s,i}^l} + \mathrm{U_o}\left(\mathrm{r_{s,i}^l} \odot \mathrm{v_i^{(l-1)}}\right)\right) \tag{6}$$

$$\widehat{\mathrm{v_i^l}} = \left(1 - \mathrm{z_{s,i}^l}\right) \odot \mathrm{v_i^{(l-1)}} + \mathrm{z_{s,i}^l} \odot \widetilde{\mathrm{v_i^l}} \tag{7}$$

where $\mathrm{W, W_z, W_r, W_o} \in \mathrm{R^{d\times 2d}}$ and $\mathrm{U_z, U_r, U_o} \in \mathrm{R^{d\times d}}$ are trainable parameters. $\mathrm{z_{s,i}}$ and $\mathrm{r_{s,i}}$ are the update gate and reset gate, respectively. $\left[\mathrm{v_1^{(l-1)}}, \cdots, \mathrm{v_k^{(l-1)}}\right]$ is the node embedding list of session S in the l-1th layer, $\odot$ denotes element-wise product, and σ represents a Sigmoid activation function. $\mathrm{A_s} \in \mathrm{R^{k\times 2k}}$ denotes the concatenation of the adjacency matrices of the input and output edges. For a session $\mathrm{s} = [\mathrm{v_2, v_3, v_6, v_3, v_5}]$, corresponding matrix $\mathrm{A_s}$ is shown in Fig. 1. $\mathrm{A_{s,i:}} \in \mathrm{R^{1\times 2k}}$ corresponds to two columns in the adjacency matrix $\mathrm{A_s}$ of the node $\mathrm{v_{s,i}}$. $\mathrm{z_{s,i}^l}$ and $\mathrm{r_{s,i}^l}$ are the update gate and the reset gate, respectively, which decide which information should be kept or dropout. The final state $\widehat{\mathrm{v_i^l}}$ is a combination of the previous hidden state $\mathrm{v_i^{(l-1)}}$ and the candidate state $\widetilde{\mathrm{v_i^l}}$. Update gate updates all satellite nodes in the star graph.

Next, we consider how to integrate the information of the star nodes into the satellite nodes to capture long-range dependencies. We use gate mechanisms to fuse the information of adjacent nodes $\widehat{\mathrm{v_i^l}}$ and star nodes $\mathrm{v_s^{l-1}}$.

$$\mathrm{v_i^l} = \left(1 - \mathrm{a_i^l}\right)\widehat{\mathrm{v_i^l}} + \mathrm{a_i^l v_s^{l-1}} \tag{8}$$

Here, $\mathrm{a_i^l}$ is the weight estimated for the importance of adjacent node $\widehat{\mathrm{v_i^l}}$ and star node $\mathrm{v_s^{l-1}}$. Therefore, we implement $\mathrm{a_i^l}$ as follows:

$$\mathrm{a_i^l} = \frac{\left(\mathrm{W_1}\widehat{\mathrm{v_i^l}}\right)^{\mathrm{T}} \mathrm{W_2 v_s^{l-1}}}{\sqrt{\mathrm{d}}} \tag{9}$$

$\mathrm{W_1, W_2} \in \mathrm{R^{d\times d}}$ are the weight matrix, $\widehat{\mathrm{v_i^l}}$ and $\mathrm{v_s^{l-1}}$ are the item representations corresponding to the satellite node $\mathrm{v_i}$ and star node $\mathrm{v_s}$, respectively, and $\sqrt{\mathrm{d}}$ is the scaling coefficient.

Inspired by XSimGCL [26], we achieve data augmentation by adding noise to the representation of the satellite nodes. Formally, for a satellite node $\mathrm{v_i}$ and its representation in l-th layer, we can implement the following representation-level augmentation:

$$\mathrm{v_i^{l+n}} = \mathrm{v_i^l} + \Delta_{\mathrm{i}}^{'} \tag{10}$$

$$\odot = \mathrm{X} \odot \mathrm{sign}\left(\mathrm{v_i^{l+n}}\right), \ \mathrm{X} \in \mathrm{R^d} \sim \mathrm{U}(0, 1) \tag{11}$$

$\Delta_{\mathrm{i}}^{'}$ is the added scaled noise vector and $\|\Delta\|_2 = \epsilon$. ϵ is a small constant. Geometrically, by adding the scaled noise vector, rotating the original vector at a small angle can be achieved, as shown in Fig. 3. Each rotation corresponds to a deviation of $\mathrm{e_i}$ and generates an augmented representation. Since the angle of rotation is small enough, the representation after adding noise preserves most of the information from the original representation, while introducing variation. We choose to generate noise from a uniform distribution, which provides uniformity to the augmentation.

After updating the embedding representation of satellite nodes, we also need to update the embedding representation of star nodes. We use an attention mechanism to distinguish the

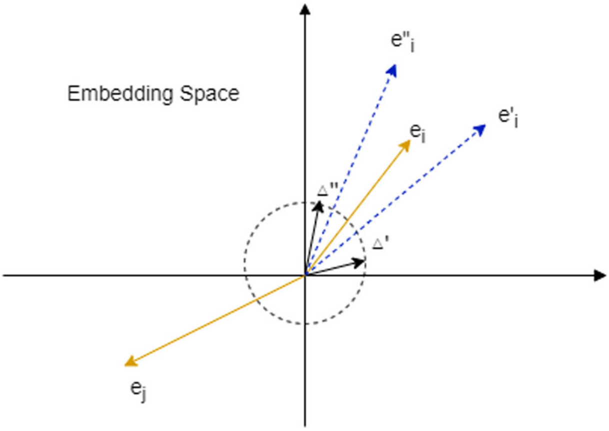


Fig. 3 An illustration for the noise-based representation enhancement

importance of different satellite nodes. The importance of each satellite node is determined by the similarity between the star node and the satellite node:

$$\upbeta_{\mathrm{i}} = \mathrm{softmax}\left(\frac{\left(\mathrm{W}_3\mathrm{v}_{\mathrm{i}}^{\mathrm{l}}\right)^{\mathrm{T}}\mathrm{W}_4\mathrm{v}_{\mathrm{s}}^{\mathrm{l}-1}}{\sqrt{\mathrm{d}}}\right) \tag{12}$$

$\mathrm{W}_3, \mathrm{W}_4 \in \mathrm{R}^{\mathrm{d}\times\mathrm{d}}$ are the weight matrix. we update the embedding representation of star nodes by calculating the linear combination of satellite nodes corresponding to each coefficient:

$$\mathrm{v}_{\mathrm{s}}^{\mathrm{l}} = \sum_{\mathrm{i}=1}^{\mathrm{k}} \upbeta_{\mathrm{i}}\mathrm{v}_{\mathrm{i}}^{\mathrm{l}} \tag{13}$$

To alleviate the overfitting problem in graph neural networks, we apply the highway network [33] in the last layer of the SGNN. The highway gate is used to calculate the final hidden state h^f, the weighted sum of the initial embedding h^0 of the satellite node and the embedded v^{L+n} of the last layer. Highway network can be described as follows:

$$h^{\mathrm{f}} = \gamma \odot h^0 + (1-\gamma) \odot v^{L+n} \tag{14}$$

$$\gamma = \upsigma\left(\mathrm{W}_5\left[h^0 || v^{L+n}\right]\right) \tag{15}$$

where $\odot$ is the element product, σ is a Sigmoid function, || represents the concatenation operation, and $\mathrm{W}_5 \in \mathrm{R}^{\mathrm{d}\times 2\mathrm{d}}$ is a trainable weight matrix.

3.4.3 Session Embedded Learning

After obtaining the embedding representations of satellite nodes and star nodes, we can get the item embeddings $\mathrm{x} \in \mathrm{R}^{\mathrm{d}\times\mathrm{k}}$ from the corresponding satellite nodes $h^{\mathrm{f}} \in \mathrm{R}^{\mathrm{d}\times\mathrm{m}}$. Then, we consider the user's global and current preferences to generate the final session representation as the user's preference. Similar to previous research [5, 6], we take the last item in the session sequence ie.,$\mathrm{S}_{\mathrm{last}} = \mathrm{x}_{\mathrm{k}}$ as the user recent preference.

For the user's global preference, we consider generating a session embedding that represents the global preference by aggregating the embeddings of all the satellite nodes in the session sequence. Since different items have different levels of importance for modelling user preferences, we use a soft attention mechanism to weight the importance of each item. It is worth noting that the importance of each item in the session sequence is jointly determined by the star node v_s, the current item x_i, and the user's recent preference S_{last}:

$$u_i = q_0^T \sigma(q_1 x_i + q_2 v_s + q_3 S_{last}) \tag{16}$$

$$S_g = \sum_{i=1}^{k} u_i x_i \tag{17}$$

$q_0 \in R^d$, $q_1 \in R^{d \times d}$, $q_2 \in R^{d \times d}$ and $q_3 \in R^{d \times d}$ are trainable parameters. We then combine the user's global preference S_g with the current interest S_{last} as the final session presentation.

$$S_h = q_4 [S_g || S_{last}] \tag{18}$$

|| represents the concatenation operation and $q_4 \in R^{d \times 2d}$ is a trainable weight matrix.

3.5 Multi-layer Aggregation Contrastive Module

Data augmentation is a key component of contrastive learning, where we abandon traditional image augmentation methods and instead employ a simple yet effective noise-based embedding augmentation and multi-layer aggregation approach to create views for contrastive learning. Specifically, both views have the same initial embedding and adjacency matrix. We employ cross-layer contrastive learning to generate different contrastive views. We aggregate the embeddings of different layers of the item obtained through multiple SGNN layers as a new view of the item v^c. The aggregation method is mean aggregation:

$$v_i^c = \frac{1}{L} \sum_{l=1}^{L} v_i^{l+n} \tag{19}$$

The contrastive module is basically the same as the generating module with only two differences. One is the different input of the highway network. The other is that we add position coding to the project in the contrastive module to integrate the sequence information into the presentation. The highway network of the contrastive module is shown below:

$$h_c^f = \gamma_c \odot h^0 + (1 - \gamma_c) \odot v^c \tag{20}$$

$$\gamma_c = \sigma(W_6 [h^0 || v^c]) \tag{21}$$

where $\odot$ is the element product, σ is a Sigmoid function, || represents the concatenation operation, and $W_6 \in R^{d \times 2d}$ is a trainable weight matrix. The final session representation S_c in the contrastive module is generated according to Eqs. (16–18). It should be noted that the two modules share the same star node v_s.

3.6 Multi-Task Learning Module

Graph contrastive learning is a multi-task learning method that can improve the performance of session recommendation models by simultaneously optimizing multiple objectives. In our model, the main task is the next-item recommendation, and contrastive learning serves as an auxiliary task to help extract general features of items from different contrastive views. We unify the two tasks and jointly optimize them:

$$L_{total} = L_{main} + \lambda L_{cl} \tag{22}$$

where λ controls the magnitude of the contrast loss.

For the next-item recommendation task, we have generated the embedded representation of the session sequence through the session representation generation module, and then use a prediction layer for next-item recommendation. To alleviate the common popularity bias problem in recommendation, we apply layer normalization separately to the session embedding $\mathrm{S_h}$ and the item embedding $\mathrm{v_i}$, and then calculate the product of the normalized session embedding $\widetilde{\mathrm{S_h}}$ and the normalized item embedding $\widetilde{\mathrm{v_i}}$ to obtain the recommendation score. Finally, we use the Softmax function to obtain the final output probability $\widehat{\mathrm{y}}$ for all items $\widehat{\mathrm{z_i}}$:

$$\widetilde{\mathrm{S_h}} = \mathrm{LayerNorm}(\mathrm{S_h}) \tag{23}$$

$$\widetilde{\mathrm{v_i}} = \mathrm{LayerNorm}(\mathrm{v_i}) \tag{24}$$

$$\widehat{\mathrm{z_i}} = \widetilde{\mathrm{S_h}}^{\mathrm{T}}\widetilde{\mathrm{v_i}} \tag{25}$$

$$\widehat{\mathrm{y}} = \mathrm{Softmax}(\widehat{\mathrm{z}}) \tag{26}$$

$\widehat{\mathrm{z_i}}$ represent the recommended score for all candidate items $\mathrm{v_i} \in \mathrm{V}$. We employed the cross-entropy loss function as the loss function of the main task, which can be expressed as:

$$\mathrm{L}(\widehat{\mathrm{y}}) = -\sum_{\mathrm{i}=1}^{\mathrm{m}} \mathrm{y_i}\log\widehat{\mathrm{y_i}} + \left(1 - \mathrm{y_i}\right)\log\left(\left(1 - \widehat{\mathrm{y_i}}\right)\right) \tag{27}$$

where $\mathrm{y_i}$ represents the probability that the item $\mathrm{v_i}$ in the next click item.

Contrastive learning can be viewed as maximize the mutual information between two potential representations. We adopt InfoNCE [34] as our contrastive loss function, and different representations of the same session sequence as our positive pair (ie., $\left\{\left(\widetilde{\mathrm{S_{h,i}}}, \widetilde{\mathrm{S_{c,i}}}\right)|\mathrm{i} \in \mathrm{S}\right\}$). $\widetilde{\mathrm{S_{h,i}}}and\widetilde{\mathrm{S_{c,i}}}$ are the normalized session representations generated by the session presentation module and the contrastive module, respectively. The negative pairs are other sessions (ie., $\left\{\left(\widetilde{\mathrm{S_{h,i}}}, \widetilde{\mathrm{S_{c,j}}}\right)|\mathrm{i, j} \in \mathrm{S}, \mathrm{i} \neq \mathrm{j}\right\}$) in the same batch. We simply implement the $\mathrm{sim(a, b)}$ function as the dot product between two vectors:

$$\mathrm{L_{cl}} = -\log\frac{\exp\left(\mathrm{sim}\left(\widetilde{\mathrm{S_{h,i}}}, \widetilde{\mathrm{S_{c,i}}}\right)/\tau\right)}{\sum_{\mathrm{j}=1,\mathrm{j}\neq\mathrm{i}}^{\mathrm{B}}\exp\left(\mathrm{sim}\left(\widetilde{\mathrm{S_{h,i}}}, \widetilde{\mathrm{S_{c,j}}}\right)/\tau\right)} \tag{28}$$

$$\mathrm{sim}\left(\widetilde{\mathrm{S_{h,i}}}, \widetilde{\mathrm{S_{c,i}}}\right) = \widetilde{\mathrm{S_{h,i}}}\widetilde{\mathrm{S_{c,i}}} \tag{29}$$

where τ is the temperature parameter and B is the size of the batch. Contrast loss encourages consistency between $\widetilde{\mathrm{S_{h,i}}}$ and $\widetilde{\mathrm{S_{c,i}}}$, which are positive samples of each other, while minimizing consistency between $\widetilde{\mathrm{S_{h,i}}}$ and $\widetilde{\mathrm{S_{c,j}}}$, which are negative samples of each other. Optimizing information loss is actually maximizing the tight lower bound of mutual information. Finally, the entire training process of SR-MACL is shown in Algorithm 1.

Algorithm 1 The whole procedure of SR-MACL

```
Input: Session S, the initial node embedding V;
Output: Recommendation lists
Construct session graph
for epoch in range(epochs) do
    for each session s do
        Learn item and session representations through Eqs. (3) - (18) ;
        Learn item and session representations of the contrastive view through Eqs.
        (16) - (21) ;
        Compute contrastive learning loss of two views via Eq. (28);
    end for
        Using Adam jointly optimize $L_{main}$ and $L_{cl}$ in Eq. (22);
end for
```

4 Experiments

4.1 Datasets and Preprocessing

To thoroughly evaluate the proposed approach, we selected three public datasets containing user interactions: Diginetica, Tmall, and Nowplaying. These three datasets differ in size, sparsity, and scenario.

- **Diginetica** is from the CIKMCup in 2016, which consists of typical transaction data.
- **Tmall** is from the 2015 IJCAI competition, which contains anonymous users on the Tmall online shopping platform shopping log.
- **Nowplaying** describes music listening behaviour extracted from Twitter.

Our processing of the datasets is consistent with previous work [5, 7, 35]. Specifically, sessions of length 1 and entries with less than 5 occurrences were filtered in all three public datasets. In addition, for each session sequence$S = \left\{v_1^s, v_2^s, \ldots, v_m^s\right\}$, we process the splits into sequences and the corresponding labels, i.e.,$\left(\left[v_1^s\right], v_2^s\right)$,$\left(\left[v_1^s, v_2^s\right], v_3^s\right)$,…,$\left(\left[v_1^s, v_2^s, \ldots, v_{n-1}^s\right], v_n^s\right)$. The processed dataset statistics are shown in Table 1.

4.2 Evaluation Metric

As described in [2, 7, 36], the evaluation indicators include: P@20 and MRR @20. P@20 is widely used as a measure of predictive accuracy. It represents the percentage of correctly recommended items among the top 20 items as defined by Eq. (30) and is used to measure

Table 1 Statistics of the dataset

Dataset	Diginetica	Tmall	Nowplaying
# click	982,961	818,479	1,367,963
# train	719,470	351,268	825,304
# test	60,858	25,898	89,824
# items	43,097	40,728	60,417
avg.len	5.12	6.69	7.42

the accuracy of the recommendation.

$$\mathrm{P@20} = \frac{1}{\mathrm{N}} \sum_{\mathrm{i}=1}^{\mathrm{N}} \mathrm{y_i} \tag{30}$$

N is the total number of sessions, and $\mathrm{y_i}$ indicates whether the top 20 recommended results in the session contain the target item. If the recommended item contains the corresponding label, the value is 1. Otherwise, it is 0.

MRR@20 (Mean Reciprocal Rank) is calculated based on the average rank of the target items in the top 20 recommendations. As soon as the rank surpasses 20, the reciprocal rank's value is 0. The MRR metric considers the order in which the recommendations are sorted, where a larger MRR value indicates that the correct recommendation is at the front of the sorted list. As shown in Eqs. (31) and (32):

$$\mathrm{MRR@20} = \frac{1}{\mathrm{N}} \sum_{\mathrm{i}=1}^{\mathrm{N}} \mathrm{Rec(i)} \tag{31}$$

$$\mathrm{Rec(i)} = \begin{cases} \frac{1}{\mathrm{Rank(i)}}, \ Rank(\mathrm{i}) \le 20 \\ 0, \ Rank(\mathrm{i}) > 20 \end{cases} \tag{32}$$

where Rank(i) is the rank of tags in session i, Rec(i) is the reciprocal of the rank of target items in session i. If the rank is greater than 20, the value of Rec(i) is set to 0. Both evaluation metrics P@20 and MRR@20 have larger values representing better recommendation performance.

4.3 Baseline Algorithm

We compared our method with ten baseline models, which can be divided into three categories: (1) traditional deep learning models: GRU4REC, NARM, STAMP; (2) graph neural network model: SR-GNN, SGNN-HN, GCE-GNN and GC-HGNN; (3) graph comparison models: COTREC, S^2-DHCN, CGL.

- GRU4REC [2]: It applied RNN to SRS for the first time together with GRU, demonstrates the effectiveness of deep learning methods in SRS
- NARM [3]: It incorporated an attention mechanism that apprehends the user's primary goal and integrated it with persistent behavioural features to form a final representation to predict the next item.
- STAMP [4]: It is an approach that uses a simple multilayer perceptron (MLP) augmented by an attention mechanism to capture both the general interest of the user and the current interest of the current session.

- SR-GNN [5]: It is first application of graph neural networks for SBR, which first introduces gated graph neural networks (GGNNs) to capture complex item transitions. To generate the next item for the current session, it employed the same idea as STAMP, and used the attention mechanism to capture the general and current interests of the user.
- SGNN-HN [6]: It used a star node to capture long-range dependencies and employed a highway network (HN) to adaptively select embeddings from item representations to prevent overfitting.
- GCE-GNN [7]: It employed local graph and global utilization graph attention networks (GAT) to capture item transfer relations from local and global contexts and used reverse location encoding to generate session representations of SBR.
- GC-HGNN [8]: It constructed global graph and models information from other sessions using hyper-graph convolution and fused global and local information by pooling.
- S2-DHCN [9]: It employed hypergraph convolution neural networks and graph attention networks to obtain global contextual and local information and used attention mechanisms to process fused features to learn the final representation of session sequences.
- COTREC [10]: It proposed a self-supervised collaborative training method based on contrastive learning as a secondary task to alleviate the data sparsity problem and retained the complete session information by generating enhanced intra-session and inter-session views.
- CGL [12]: It constructed a self-supervised module to enrich item representation using global graph and decoupling learning and designed the label obfuscation method to prevent overfitting.

4.4 Parameter Settings

For a fair comparison, we employed the same data preprocessing method for all baseline models. The initial parameters were all initialized using a Gaussian distribution with mean 0 and standard deviation 0.1. All baseline models use L_2 regularization as a penalty term with values of 10^{-5}. All models have an embedding size of 100 and use Adam as the optimizer with an initial learning rate of 0.001 and decay by 0.1 after every 3 epochs. A search is performed on the validation set to obtain its optimal value. Randomly selected 10% of the data in the training set is used as the validation set. The layer of the model is three.

Table 2 shows the overall performance of SR-MACL compared to the baseline models, in which the best results are highlighted in bold, and the second-best results are italic. We use the average of the results from five runs as the final result, the values in parentheses represent the standard deviations. By comparing the experimental results, we can draw the following three experimental conclusions:

(1) The graph neural network-based models outperform the traditional deep learning-based models (RNN-based, Attention-based), which show the powerful ability of graph neural networks in modelling the transduction relations of session sequences and also show that the graph neural network approach is more suitable for session recommendation.

(2) As shown in Table 2, the performance of existing session recommendation models based on graph contrastive learning is not as good as the session models based on graph neural networks alone. This indicates that the approach of using other session information to generate augmented views for comparison learning may introduce information about irrelevant items resulting in sub-optimal model performance.

(3) SR-MACL outperforms the other baseline models on all three datasets (except for Nowplaying's P@20 metric), which indicates that our proposed approach of augmenting

Table 2 Comparison of the performance of the baseline models

Model	Diginetica		Tmall		Nowplaying	
	P@20	MRR@20	P@20	MRR@20	P@20	MRR@20
GRU4REC	30.85	8.32	10.93	5.89	7.92	4.48
NARM	48.32	16.03	23.30	10.70	18.59	6.93
STAMP	45.98	14.52	26.47	13.36	17.66	6.88
SR-GNN	51.30	17.80	27.57	13.72	18.87	7.47
GCE-GNN	54.22	19.04	35.09	15.80	22.43	8.40
SGNN-HN	*55.34*	*19.25*	*37.16*	17.78	23.03	*8.48*
GC-HGNN	54.10	18.64	36.83	17.37	**23.65**	7.83
S^2-DHCN	53.66	18.57	31.42	15.05	*23.50*	8.18
COTREC	54.18	19.07	36.35	*18.04*	22.56	7.73
CGL	54.26	19.12	35.73	16.21	22.59	8.32
SR-MACL	**55.48(0.13)**	**19.52(0.09)**	**37.81(0.23)**	**18.10(0.15)**	23.14(**0.06**)	**8.94(0.08)**

item representations for comparison learning by multi-layer aggregation outperforms the above approach of using other session information to generate augmented views for comparison learning.

4.5 Ablation Experiments

We further analysed the model by experimentally analysing the impact of each component in SR-MACL on the model performance. We design two SR-MACL variants: SR-MACL-HW, SR-MACL-CL and compare these variants with the full SR-MACL model on the Diginetica, Nowplaying, and Tmall datasets. It should be noted that the ablation experiments were all conducted with the layer number is 3.

- SR-MACL-HW: Removal of high-speed networks
- SR-MACL-CL: Removal of the contrastive learning module

The experimental results are shown in Table 3, in which the best results are highlighted in bold. The two key components of our model SR-MACL, highway network and contrastive learning, both contribute to the model performance improvement. Highway network has the greatest impact on the model performance because it can solve the overfitting problem caused

Table 3 Impact of different components

Method	Diginetica		Tmall		Nowplaying	
	P@20	MRR@20	P@20	MRR@20	P@20	MRR@20
SR-MACL-HW	54.61	18.55	35.99	15.23	21.92	7.71
SR-MACL-CL	55.34	19.25	37.16	17.78	23.03	8.48
SR-MACL	**55.48**	**19.52**	**37.81**	**18.10**	**23.14**	**8.94**

by using three stacked graph encoders. In addition, we also demonstrate the effectiveness of contrastive learning in our model by the above experiments.

4.6 Effect of the Hyper Parameter

We introduce a hyper parameter λ in SR-MACL to balance the contrast module. We experimentally investigate the performance of SR-MACL at different values to explore the impact of the contrast module, the values range is [0, 0.1, 0.01, 0.001]. The experimental results are shown in Fig. 4. We can see that the results on three used datasets are similar. When λ = 0.001, the model achieved the best results on all three datasets, and there is a significant decrease in model performance as the value increases, which we believe is due to excessive contrastive loss that interferes with the learning of the main task. Also, when λ = 0, the model performance decreases somewhat compared to the best performance, which suggests that the addition of the contrast module learns the richer representation and thus improves the model performance.

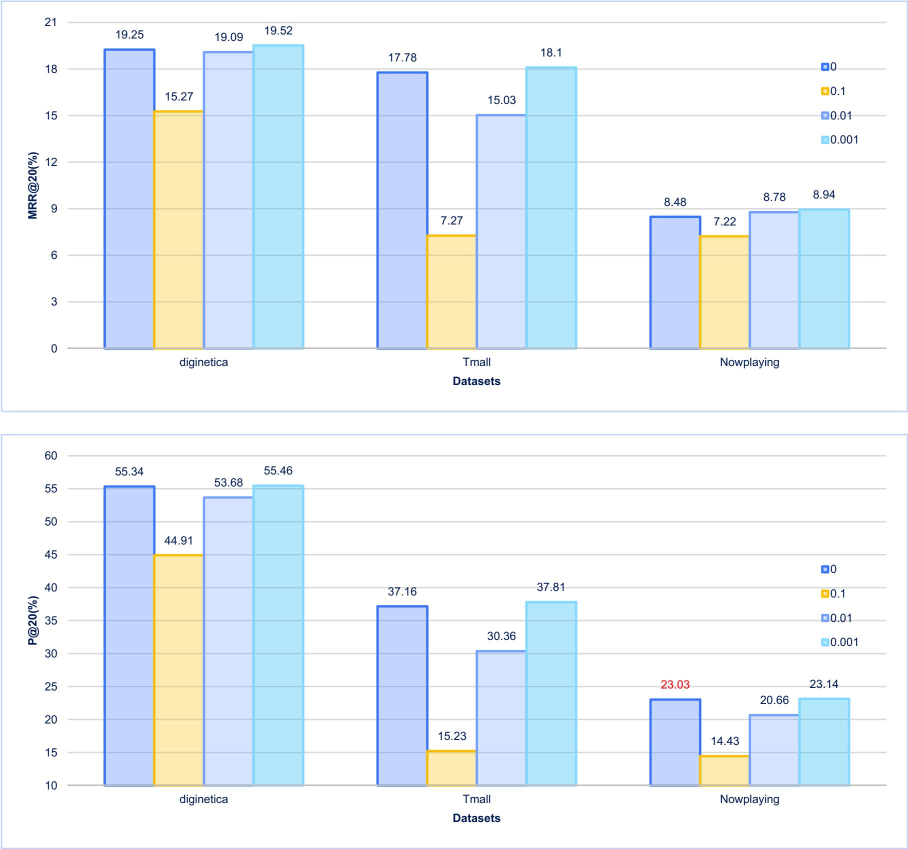


Fig. 4 Performance comparison under different contrastive loss parameters

Table 4 Effects of different aggregation approaches

Method	Diginetica		Tmall		Nowplaying	
	P@20	MRR@20	P@20	MRR@20	P@20	MRR@20
Mean-pooling	54.61	18.55	35.99	15.23	21.92	7.71
Max-pooling	55.34	19.25	37.16	17.78	23.03	8.48
SA-pooling	**55.48**	**19.52**	**37.81**	**18.10**	**23.14**	**8.94**

4.7 Impact of Aggregation Operations

We conducted further analysis of the model to investigate the influence of different aggregation methods on its performance. Aggregating item representations within the session sequence is vital for session-based recommendation. Consequently, we conducted several comparative experiments to assess the impact of various aggregation methods on the model's performance.

- **Mean-Pooling**: The average pooling is used to aggregate the items represents in the session sequence to generate session embedding.
- **Max-Pooling**: The max pooling is used to aggregate the items represents in the session sequence to generate session embedding.
- **SA-Pooling**: The soft attention pooling is used to aggregate the items represents in the session sequence to generate session embedding.

As can be seen from Table 4, in which the best results are highlighted in bold, the aggregate methods Meanpooling and Max-Pooling do not achieve satisfactory results. Compared with the above two aggregation methods, the aggregate methods SA-Pooling can get better results because it assigns a different weight to each item in the session sequence. This enables the aggregation of features based on their relative importance, resulting in improved results.

4.8 Data Sparsity

We compare the performance of models trained with different proportions of data and list them in Table 5, in which the best results are highlighted in bold. The experimental results show that our model performs significantly better on smaller data sets than other models that

Table 5 Experiment results trained on the sparse

Method	Diginetica		Diginetica 1/2		Diginetica 1/4	
	P@20	MRR@20	P@20	MRR@20	P@20	MRR@20
SRGNN	49.30	15.92	44.30	14.12	37.22	11.37
GCE-GNN	54.22	19.04	50.31	17.24	45.23	15.14
SR-MA	55.34	19.25	52.25	18.34	47.15	16.03
SR-MACL	**55.48**	**19.52**	**55.02**	**19.21**	**54.25**	**18.53**
SRGNN	49.30	15.92	44.30	14.12	37.22	11.37

do not use contrastive learning. This shows that our model helps mitigate the data sparsity problem.

5 Conclusions and Future Work

Existing session-based recommendation methods based on graph contrast learning usually incorporate other session information to generate augmented views to construct graph contrastive learning, which inevitably introduces irrelevant item information and interferes with accurately modelling user interests, resulting in sub-optimal model performance. We propose a new session-based recommendation method based on multi-layer aggregated augmented contrast learning, namely (SR-MACL). In SR-MACL we construct a contrast view by adding noise to the embedding representation and forming a contrast embedding representation by multi-layer aggregation, which not only effectively solves the problem that traditional graph enhancement methods can destroy the context of the whole session, but also avoids the interference of irrelevant items. The experimental results show that our model outperforms other session recommendation models and provides a new way of thinking for the application of graph contrast learning to session recommendation. In the current work, we focus on how to enhance the item representation, and in future work we intend to further investigate how to enhance the session representation in session-based recommendation.

Acknowledgements This work is supported by the National Natural Science Foundation of China (Grant No. 62341314), the Gansu Natural Science Foundation (Grant No. 22JR5RA156), the Industrial Support Program Project of Gansu Province (Grant No. 2021CYZC-06), the Backbone Fund of Youth Teachers' Capability Promotion (Grant No. NWNU-LKQN2020-14).

Author contributions SG: Conceptualization, Methodology, Writing—original draft, Funding acquisition. YZ: Methodology, Software, Resources, Writing—review & editing. XD: Methodology, Funding acquisition. XD: Software, Resources, Formal analysis.

Declarations

Conflict of interest The authors declare that they have no conflict of interest. The authors have no relevant financial or non-financial interests to disclose.

References

1. Rendle S, Freudenthaler C, Schmidt-Thieme L (2010) Factorizing personalized Markov chains for next-basket recommendation. In: WWW, pp 811–820
2. Hidasi B, Karatzoglou A, Baltrunas L et al (2016) Session-based recommendations with recurrent neural networks. In: ICLR
3. Jing L, Ren P, Chen Z et al (2017) Neural attentive session-based recommendation. In: CIKM, pp 1419–1428

4. Liu Q, Zeng Y, Mokhosi R et al (2018) STAMP: short-term attention/memory priority model for session-based recommendation. In: SIGKDD, pp 1831–1839
5. Wu S, Tang Y, Zhu Y et al (2019) Session-based recommendation with graph neural networks. In: AAAI, pp 346–353
6. Wang Z, Wei W, Cong G et al (2020) 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
7. Pan Z, Cai F, Chen W et al (2020) Star graph neural networks for session-based recommendation. In: Proceedings of the 29th ACM international conference on information & knowledge management, pp 1195–1204
8. Peng D, Zhang S (2020) GC-HGNN: a global-context supported hypergraph neural network for enhancing session-based Recommend. Electron Commer Res Appl 52:101129
9. Xia X, Yin H, Yu J et al (2021) Self-supervised hypergraph convolutional networks for session-based recommendation. In: AAAI
10. Xia X, Yin H, Yu J et al (2021) Self-Supervised graph co-training for session-based recommendation. In: CIKM, pp 2180–2190
11. Cao Y, Zhang X, Zhang F et al (2023) SimCGNN: simple contrastive graph neural network for session-based recommendation. arXiv:2302.03997
12. Pan Z, Cai F, Chen W et al (2022) Collaborative graph learning for session-based recommendation. ACM Trans Inf Syst 40(4):1–26
13. Lee D, Kang S, Ju H et al (2021) Bootstrapping user and item representations for one-class collaborative fifiltering. In: SIGIR, pp 1513–1522
14. Yu J, Yin H, Xia X, Chen T, Cui L, Nguyen QVH (2022) Are graph augmentations necessary? Simple graph contrastive learning for recommendation. In: SIGIR, pp 1294–1303
15. Yu J, Yin H, Xia X et al (2022) XSimGCL: towards extremely simple graph contrastive learning for recommendation. Trans Knowl Data Eng
16. Pan Z, Cai F, Ling Y et al (2020) Rethinking item importance in session-based recommendation. In: SIGIR, pp 1837–1840
17. Chen W, Cai F, Chen H et al (2019) A dynamic co-attention network for session-based recommendation. In: CIKM, pp 1461–1470
18. Li Y, Tarlow D, Brockschmidt M et al (2015) Gated graph Sequence neural Networks. Comput Sci
19. Gupta P, Garg D, Malhotra P et al (2019) NISER: normalized item and session representations to handle popularity bias. arXiv:1909.04276
20. You Y, Chen T, Sui Y et al (2020) Graph contrastive learning with augmentations, In: NeurIPS
21. Chen T, Kornblith S, Norouzi M, Hinton G (2020) A simple framework for contrastive learning of visual representations. In: ICML, pp 1597–1607
22. Gao T, Yao X, Chen D (2021) Simcse: simple contrastive learning of sentence embeddings. In: EMNLP, pp 6894–6910
23. He K, Fan H, Wu Y et al (2020) Momentum contrast for unsupervised visual representation learning. In: CVPR, pp 9729–9738
24. Grill JB, Strub F, Altché F et al (2020) Bootstrap your own latent: a new approach to self-supervised learning. In: NeurIPS
25. Singh M (2020) Scalability and sparsity issues in recommender datasets: a survey. Knowl Inf Syst 62:1–43
26. Nguyen TT, Weidlich M, Thang DC et al (2017) Retaining data from streams of social platforms with minimal regret. In: IJCAI, pp 2850–2856
27. Chen T, Yin H, Nguyen QVH et al (2020) Sequence-aware factorization machines for temporal predictive analytics. In: 2020 IEEE 36th international conference on data engineering (ICDE), pp 1405–1416
28. Wu J, Wang X, Feng F et al (2021) Self-supervised graph learning for recommendation. In: SIGIR, pp 726–735
29. Yang Y, Huang C, Xia L et al (2023) Debiased contrastive learning for sequential recommendation. In: WWW, pp 1063–1073
30. Xie X, Sun F, Liu Z et al (2022) Contrastive learning for sequential recommendation. In: ICDE. IEEE, pp 1259–1273
31. Guo Q, Qiu X, Liu P et al (2019) Star-transformer. In: NAACL, pp 1315–1325
32. Lin Z, Tian C, Hou Y, Zhao WX (2022) Improving graph collaborative fifiltering with neighborhood-enriched contrastive learning. In: WWW, pp 2320–2329.
33. Srivastava RK, Greff K, Schmidhuber J (2015) Highway networks. arXiv:1505.00387
34. Oord AVD, Li Y, Vinyals O (2018) Representation learning with contrastive predictive coding. arXiv:1807.03748

35. Xu C, Zhao P, Liu Y et al (2019) Graph contextualized self-attention network for session-based recommendation. In: IJCAI, pp 3940–3946
36. He R, McAuley J (2016) Fusing similarity models with Markov chains for sparse sequential recommendation. In: ICDM, pp 191–200

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.