



# Session-based recommendation with fusion of hypergraph item global and context features

Xiaohong Han<sup>1</sup> · Xiaolong Chen<sup>1</sup> · Mengfan Zhao<sup>2</sup> · Ting Liu<sup>1</sup>

Received: 14 June 2023 / Revised: 31 October 2023 / Accepted: 26 December 2023 /  
Published online: 24 January 2024

© The Author(s), under exclusive licence to Springer-Verlag London Ltd., part of Springer Nature 2024

## Abstract

Session-based recommendation (SBR) is to predict the items that users are likely to click afterward by using their recent click history. Learning item features from existing session data to capture users' current preferences is the main problem to be solved in session-based recommendation domain, and fusing global and local information to learn users' preferences is an effective way to obtain this information more accurately. In this paper, we propose a session-based recommendation with fusion of hypergraph item global and context features (FHGIGC), which learns users' current preferences by fusing item global and contextual features. Specifically, the model first constructs a global hypergraph and a local hypergraph and uses the hypergraph neural network to learn item global features and local features by relevant session information and item contextual information, respectively. Then, the learned features are fused by the attention mechanism to obtain the final item features and session features. Finally, personalized recommendations are generated for users based on the fused features. Experiments were conducted on three datasets of session-based recommendation, and the results demonstrate that the FHGIGC model can improve the accuracy of recommendations.

**Keywords** Session-based recommendation · Hypergraph convolutional network · Hypergraph attention network · Attention mechanism · Feature fusion

---

Xiaohong Han and Xiaolong Chen contributed equally to this work.

---

✉ Xiaolong Chen  
little\_xl\_chen@163.com

Xiaohong Han  
382665448@qq.com

Mengfan Zhao  
zhaomengfan@tju.edu.cn

Ting Liu  
303779601@qq.com

<sup>1</sup> Information and Electrical Engineering, Hebei University of Engineering, Handan 056038, China

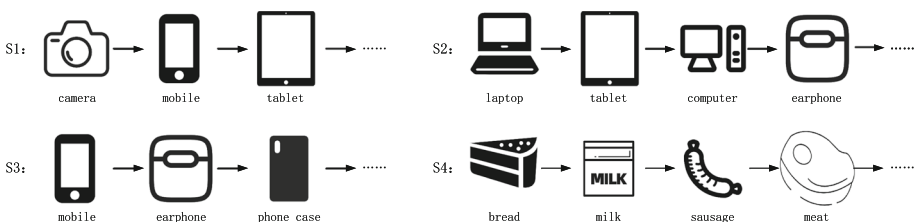
<sup>2</sup> Computer and Software Engineering, Sias University, Zhengzhou 451150, China

## 1 Introduction

With the rapid development of the information age, Internet users are in an environment of information explosion. An ensuing problem is how users can quickly and accurately obtain the required information from the huge and complex web data. Recommender systems [1, 2] have been proposed to solve the problem due to their powerful ability for capturing users' interests and have attracted a lot of attention from researchers. However, traditional recommendation systems have some problems in capturing users' short-time preferences, such as noise in users' historical interaction data, failure to reflect users' short-time preference shifts through historical information and personal data [3, 4], which leads to unsatisfactory recommendation results. In order to solve the problems of traditional recommendation systems, the session-based recommendation was born.

Session-based recommendation aims to predict the next item a user is likely to click through the user's click sequence in the recent period. Based on this feature, session-based recommendation can better capture the current interest preference shift of users, thus improving the effectiveness and accuracy of the recommendation system. How to learn and capture the features of items in a session and user preferences from session data has also become a core problem in the field of session-based recommendation. As shown in Fig. 1, among the four user browsing records, sessions S1, S2, and S3 reflect users' interest in electronics and accessories, while users in session S4 are clearly more likely to buy food.

In response to the problem of the above example, a large number of researchers have explored how to obtain user preferences and generate accurate recommendation results from such session data. In early studies of session-based recommendation, researchers mainly used collaborative filtering (CF) [5–7] and Markov chain (MC) [8, 9], which largely ignore item interactions in session data and are ineffective in making recommendations on sparse data. Matrix factorization (MF) [10–12] methods perform well in dealing with sparse matrices, but pose the problem of high time complexity as well as poor interpretability. Subsequently, recurrent neural networks (RNNs) [13, 14] as well as attention mechanisms [15–17] have also been used in the field of recommender systems due to their excellent handling of sequential data, and have been used to capture the sequential relationships of a session in session-based recommendation. Although this approach, which relies on sequential structure, brings some improvement to session-based recommendation, it still has many limitations. Thanks to the higher-order feature representation capability of graph structures, graph neural networks (GNNs) [18–21] have been widely used in various fields in recent years, especially in the field of recommender systems, which has received wide attention and become a mainstream approach. The session-based recommendation based on graph neural networks [22–24] learns the similarity of items in a session by combining multiple sessions in a graph neural network through graph structures, and although it can provide more information for items, this method



**Fig. 1** A toy example of item conversion in different sessions

of transferring information only through connected pairs of nodes still limits the performance of session-based recommendation.

Compared with the ordinary graph where node pairs are connected to each other, the hypergraph [25] connects two or more nodes per hyperedge, which can effectively transfer information between multiple nodes. By introducing this hypergraph structure, a hyperedge can connect multiple items when constructing a session graph, which can effectively capture the complex relationship between two or more items in the session data. In session-based recommendation, researchers usually propagate information about all items at the session level [26, 27], which allows effective learning of the overall features of the session, but also ignores the structural features of the items within the session to some extent. In this paper, Session-based Recommendation with Fusion of Hypergraph Item Global and Context Features (FHGIGC) is proposed. Firstly, all session data are constructed as a global hypergraph, and different sessions are interconnected by shared nodes to learn item features by related sessions in the hypergraph neural network. Then, for each session, the local hypergraph is constructed by using the context window as a hyperedge, and the interrelationship of items within the session is learned by the contextual information of the items in the session. Next, the learned item features are fused according to the importance level by an attention mechanism to obtain the final item features and session features, which are finally used in the recommendation task. The main contributions can be summarized as follows:

- To learn the features of items in a global session, all sessions are constructed as global hypergraphs and information is propagated through Hypergraph Convolutional Neural Networks (HGCN). Then local hypergraphs are constructed in individual sessions with contextual information of items and the information is propagated through Hypergraph Attention Network (HGAT) to learn the item interaction features within the sessions.
- The attention mechanism learns the different importance of inter-session holistic features and intra-session item relationship features, and fuses the features learned from global and local hypergraph structures according to this importance, and finally generates personalized recommendations for sessions based on the fused feature information.
- Extensive experiments were conducted on three datasets, Tmall, Diginetica, and RetailRocket, respectively, to demonstrate that the FHGIGC model outperforms the existing baseline model, and the designed modules were also shown to improve the recommendation accuracy through ablation experiments.

## 2 Related works

### 2.1 SBR based on traditional methods

Collaborative filtering is mainly used to make recommendation by calculating the similarity between users and items. Item-KNN [5] inferred items that might be of interest to users by calculating the cosine similarity between different items and recommended them to users. With the development of machine learning, there are also many researches on combining machine learning in collaborative filtering for recommendation. NCF [28] used a multiple perceptron approach to learn user-item interactions. J-NCF [29] further combined neural collaborative filtering to learn deep user-item features from user-item interactions. Although these methods have achieved some success, the accuracy of the recommendation results is somewhat affected by ignoring the user preference shifts in the sequence.

Markov chain-based recommender systems predict subsequent possible items from the user's current continuous session items. FPMC [8] captures the sequence information of a session by combining a matrix decomposition model and a first-order Markov chain. Shani et al. [30] predicted user preferences by Markov decision process for the sequential purchase behavior of users. Similarly, Zimdars et al. [9] used the user interactions in a session as a sequence to make recommendations based on temporal relationships. However, the Markov chain-based approach only focuses on first-order interaction relations and only performs local feature extraction based on the current sequence, ignoring global session information resulting in unsatisfactory recommendation results.

## 2.2 SBR based on deep learning

Recurrent neural networks were first proposed to solve problems related to natural language processing, and are also of interest in the field of session-based recommendation due to their ability to learn sequence structure [31]. GRU4REC [13] first applied RNNs to session-based recommendation to iterate the item sequence information in a session through a multilayer gated recurrent unit (GRU). HRNN [32] proposed hierarchical recurrent neural network to model users' personal preference variations and also to improve the personalization of recommendations by fusing personal history information. Jannach et al. [33] combined the recommendation results of the GRU4REC and the KNN in a set ratio as the recommendation results. Considering that user interests are dynamically changing, ISLF [34] uses variable automatic encoder (VAE) combined with RNNs to capture user preferences in session sequences. These methods address to some extent the problem of separate items in traditional methods and the inability to model sequential continuity information over a certain period of time, but still fail to distinguish the importance between multiple items in a session.

The main use of the attention mechanism is to calculate the importance of each input using key-value pairs according to the task requirements, so as to focus on the more important inputs. NARM [35] captures the user's main preferences in the current session sequence by introducing an attention mechanism, and also incorporates the user's history information to accomplish the recommendation task. The short-term memory attention model proposed by STAMP [36] implements a session-based recommender by fusing attention mechanisms as well as RNNs. HARSAM [37] improves the recommendation capability of the model by introducing a soft attention mechanism to model the user's interaction information and to learn the potential performance of the user. Atten-Mixer [38] leverages both concept-view and instance-view readouts to achieve multi-level reasoning over item transitions. The introduction of attentional mechanisms has enabled deep learning approaches to make great progress in session-based recommendation. However, these approaches rely excessively on information about adjacent items in the session order structure and are unable to learn the interrelationship of non-adjacent items in a session.

## 2.3 SBR based on GNN

Graph neural networks provide a convenient and intuitive perspective for modeling complex relationships among nodes, and their applications in recommender systems have been expanding as research on graph neural networks has been pushed to a new high in recent years. SR-GNN [39] proposed a graph neural network approach to model the relationship of items in a session, capturing user as well as item interaction features while maintaining

structural information between items. PGGNN [40] proposes a dual position encoding (DPE) gated graph neural network to learn the positional information of items in a session. FGNN [41] generates session embeddings through the weighted graph attention layer (WGAT) and ultimately recommendations in the context of cross-session information. FGNN-BCS [42] further construct a broadly connected session (BCS) graph to connect sessions to improve session embedding. GC-SAN [43] learns long-term dependencies between session items by combining graph neural networks and self-attention mechanisms. GARG [44] further proposed an approach based on graph convolutional neural networks and attention mechanisms to improve recommendation performance. In order to avoid information loss when constructing session graphs, LESSR [45] used edge-order preserving aggregation (EOPA) and a fast graph attention layer by modeling sessions as directed graphs. COTREC [46] improves session data utilization through a self-supervised learning approach based on graph convolutional neural networks. In these graph methods, GNN can only capture the relationships between node pairs and cannot capture the higher-order relationships of multiple items in a session, which affects the effectiveness of recommendations to some extent.

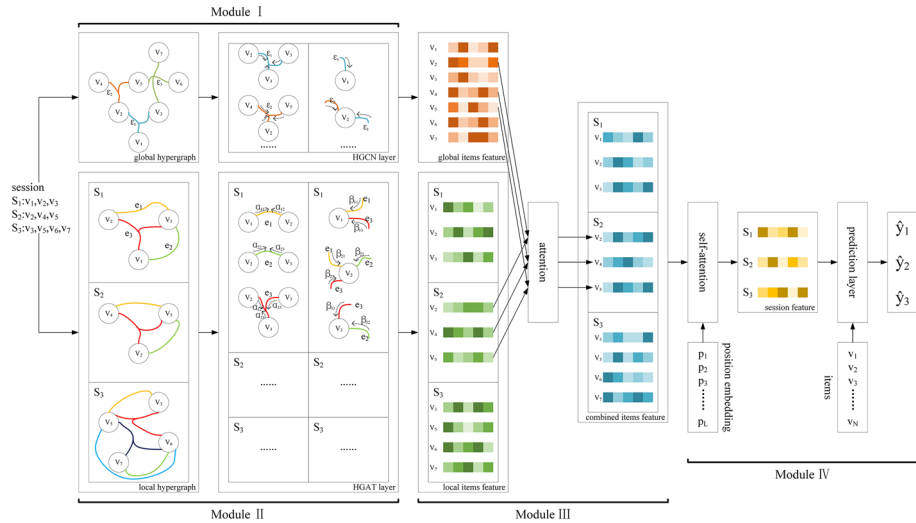
Hypergraph is an improved graph structure that can handle higher-order relationships between multiple nodes. In the hypergraph structure, each edge can connect two or more nodes and treat them as a hypernode. This special connection makes the hypergraph structure highly expressive in dealing with complex data relationships. DHCN [26] captures the relationship of all items in a session by modeling the session as a hyperedge and enhances the learning capability of the model by contrastive learning. SHARE [47] creates a hypergraph for each session that captures the relevance of items in the session. HyperS2Rec [27] improves the recommendation capability to some extent by combining the hypergraph structure with the sequential structure to learn the global and sequential relationships between session items, respectively. The FHGIGC proposed in this paper is a session-based recommendation model that combines global hypergraph and local hypergraph structures to effectively capture item consistency, relevant session information, and contextual interaction information of items in session data.

## 3 The proposed model

### 3.1 Problem setup

The set of all items is denoted as  $V = \{v_1, v_2, \dots, v_N\}$ ,  $N$  is the total number of items, and for the session  $m$  can be denoted as  $S_m = v_{m1}, v_{m2}, \dots, v_{mr}$ ,  $r$  is the length of this session. The purpose of session-based recommendation is to predict the next item of possible interest to the user based on such a session, i.e.,  $v_{m(r+1)}$ .

In the FHGIGC model proposed in this paper, sessions are modeled as global and local hypergraphs, respectively, and information is propagated through a hypergraph convolutional network and a hypergraph attention network. Figure 2 shows the overall framework of the proposed model, which consists of four main parts: the first part is the global item feature extraction module, which is used to capture the item consistency information of each session and the information in the related sessions; the second part is the item context interaction feature extraction module, which captures the context information of the items within a session through the hypergraph attention network; the third part is the item feature fusion module, which obtains the global and local item feature vectors through the attention. The fourth part is the session feature fusion and prediction module, in which the item feature



**Fig. 2** The overall framework of HGIGC model. Module I is the global hypergraph based on HGCN to learn the global item features. Module II is the local hypergraph based on HGAT to learn the local item features. Module III fusion of global and local item features to generate fused item features. Module IV incorporates an attention mechanism with position embedding to fuse item features into session features and make recommendations

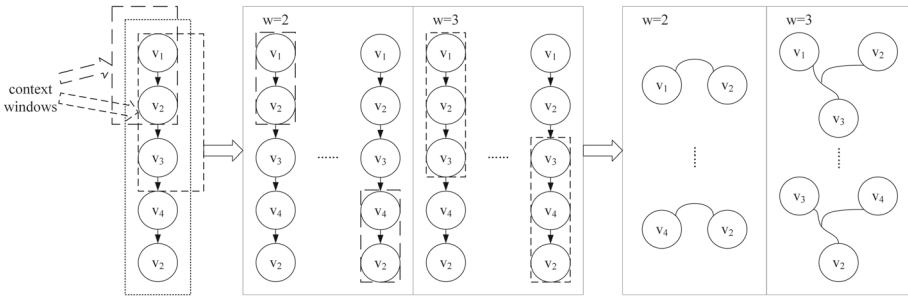
in a session are fused by the attention mechanism with position embedding to obtain the corresponding session feature, and thus calculate the probability of candidate items and recommend the higher ranked items to users as a recommendation list.

### 3.2 Global item feature extraction module

In order to capture the higher-order relationships of items in all sessions rather than between pairs of nodes, each session is treated as a hyperedge such that different sessions are constitutively connected to each other by sharing nodes, while the nodes in each session are connected to each other in the hyperedge. The hyperedges  $\epsilon_1, \epsilon_2 \epsilon_3$  constructed by sessions  $S_1, S_2, S_3$ , respectively, are shown in Fig. 2, and the global features of items are effectively captured by this structure.

The global graph notation is denoted as  $G_g = (V, E)$ ,  $V$  is the set of nodes consisting of  $N$  items in all sessions, each edge contains two or more nodes, the total number of edges is  $M$ . Each edge corresponds to a single piece of session data, and the different hyperedges are connected to each other through shared nodes in the session. For such a hypergraph, its connectivity is represented by an association matrix  $H \in R^{N \times M}$  with  $H_{ij} = 1$  when node  $v_i \in \epsilon_j$  and  $H_{ij} = 0$  otherwise. Each hyperedge  $\epsilon_j$  is initialized with weight  $W_{jj}$  and all weights form a diagonal matrix  $W \in R^{M \times M}$ . The degrees of nodes and hyperedges are defined as  $D_{ii} = \sum_{j=1}^M W_{jj}H_{ij}$ ,  $B_{jj} = \sum_{i=1}^N H_{ij}$ , respectively, and all degrees form a diagonal matrix  $D \in R^{N \times N}$  and  $B \in R^{M \times M}$ .

By constructing the hypergraph structure in the above way, a global hypergraph constructed from all the session data in the dataset as hyperedges can be obtained. It has been demonstrated in study [48] that removing the nonlinear activation functions and convolutional layer parameter matrices between the layers of a graph convolutional network does



**Fig. 3** Constructing hyperedges through contextual windows

not adversely affect the downstream tasks of the model, and the computational complexity of the information propagation process can be greatly reduced in this way. In addition, since information can be transmitted across multiple nodes in a hyperedge, global dependencies can be better captured by hypergraph convolutional neural networks. Based on the above advantages, after obtaining the global hypergraph structure introduced above, the model propagates information through the hypergraph convolutional network (HGCN) [26, 49], and the propagation process is calculated as follows:

$$x_i^{(l+1)} = \sum_{t=1}^N \sum_{j=1}^M H_{ij} H_{tj} W_{jj} x_t^{(l)} \tag{1}$$

where  $x_i^{(l)}$ ,  $x_i^{(l+1)}$  are the input of item  $i$  at layer  $l$  and the output of item  $i$  at layer  $l+1$ , respectively.  $\sum_{t=1}^N H_{tj} W_{jj} x_t^{(l)}$  is the hidden representation of hyperedge  $j$  at layer  $l$ , which represents the hyperedge collects information from its connected nodes. The row normalization matrix of Equation (1) takes the form of:

$$X^{(l+1)} = D^{-1} H W B^{-1} H^T X^{(l)} \tag{2}$$

where  $X^{(l)}$ ,  $X^{(l+1)}$  are the input and output of the layer  $l$ , respectively, and the zeroth layer is obtained by random initialization, and the weight matrix is initialized to a unit matrix.

After that, the output of multilayer HGCN is obtained by averaging the final output of HGCN, which is calculated as follows:

$$X^G = \frac{1}{L+1} \sum_{l=0}^L X^{(l)} \tag{3}$$

### 3.3 Item context interaction feature extraction module

For each session, consider the contextual information that each item in the session can provide for that item. Inspired by Wang et al. [47], we set up context windows of different sizes in each session as the hyperedges of the local hypergraph, and the items covered by the windows are all the nodes connected by this hyperedge. The process of constructing multiple hyperedges through the set context window is illustrated in Fig. 3 to obtain the local hypergraph corresponding to the current session.

The local hypergraph notation is denoted as  $G_l = (V_l, E_l)$ ,  $V_l$  is the set of all items in the current session, and  $E_l$  is the set of all hyperedges obtained through the context window.

Although the hypergraph convolutional network has been introduced in 3.2, and the hypergraph structure has been simplified to improve the computational efficiency of the model, only some of the nodes in the same hyperedge in the local hypergraph may be able to provide information for this hyperedge, and each hyperedge may have different effects on the nodes. Therefore, we propagate information over a local hypergraph by incorporating a neural network of attention, the hypergraph attention network (HGAT) [25, 47, 49], in which, unlike the traditional graph structure in which information is propagated between neighboring nodes, the hyperedges are considered as a kind of hypernodes and the node representation learning process is decomposed into two steps.

### 3.3.1 Node to hyperedge

The information is first passed from the node to the hyperedge, and the propagation through the attention mechanism is calculated as follows:

$$e_j^{(l)} = \sum_{t \in \mathcal{N}_j} h_{t-j}^{(l)} \tag{4}$$

$$h_{t-j}^{(l)} = \alpha_{jt} W_1^{(l)} x_t^{(l-1)} \tag{5}$$

where  $\mathcal{N}_j$  denotes the set of all items connected by edge  $j$ ,  $e_j^{(l)}$  denotes the feature vector of edge  $j$  at layer  $l$ ,  $x_t^{(l-1)}$  denotes the input representation of item  $t$  at layer  $l$ ,  $h_{t-j}^{(l)}$  denotes the information passed from item  $t$  to edge  $j$ ,  $W_1^{(l)} \in R^{d \times d}$  is the trainable parameter matrix.  $\alpha_{jt}$  is the attention coefficient between hyperedge  $j$  and item  $t$ , which is calculated as follows:

$$\alpha_{jt} = \frac{S(\hat{W}_1^{(l)} x_t^{(l-1)}, u^{(l)})}{\sum_{f \in \mathcal{N}_j} S(\hat{W}_1^{(l)} x_f^{(l-1)}, u^{(l)})} \tag{6}$$

where  $\hat{W}_1^{(l)} \in R^{d \times d}$ ,  $u^{(l)} \in R^d$  are the trainable parameter matrix and attention parameters, respectively, and  $S(*)$  denotes the function for calculating the similarity score, which in the experiments is computed by scaling the dot product attention [50, 51] as follows:

$$S(a, b) = \frac{a^T b}{\sqrt{D}} \tag{7}$$

### 3.3.2 Hyperedge to node

After completing the first step of propagation, similarly, the propagation from the hyperedge to the node is calculated as follows:

$$x_t^{(l)} = \sum_{j \in \mathcal{Y}_t} h_{j-t}^{(l)} \tag{8}$$

$$h_{j-t}^{(l)} = \beta_{tj} W_2^{(l)} e_j^{(l)} \tag{9}$$

where  $\mathcal{Y}_t$  denotes all edges connected to item  $t$ ,  $h_{j-t}^{(l)}$  denotes the information passed from edge  $j$  to item  $t$ , and  $W_2^{(l)} \in R^{d \times d}$  is the trainable parameter matrix.  $\beta_{tj}$  denotes the attention coefficient between item  $t$  and hyperedge  $j$ , which is calculated as follows:

$$\beta_{tj} = \frac{S(\hat{W}_2^{(l)} e_j^{(l)}, W_3^{(l)} x_t^{(l-1)})}{\sum_{f \in \mathcal{Y}_t} S(\hat{W}_2^{(l)} e_f^{(l)}, W_3^{(l)} x_t^{(l-1)})} \tag{10}$$



where  $\hat{W}_2^{(l)}, \hat{W}_3^{(l)} \in R^{d \times d}$  is the trainable parameter matrix.

### 3.4 Item feature fusion module

Considering that global and local information has different importance for the target nodes, in this process, the importance of the learned representation in the global and local structures is captured and different weights are assigned to them for feature fusion by the attention mechanism [27]. The attention coefficients are calculated as follows:

$$\mu_i^G = q_1^T \tanh(W_4 x_i^G + b_1) \tag{11}$$

$$\mu_i^L = q_1^T \tanh(W_4 x_i^L + b_1) \tag{12}$$

where  $x_i^G, x_i^L$  denote the representation learned by item  $i$  in the global hypergraph and local hypergraph, respectively,  $q_1 \in R^d, W_4 \in R^{d \times d}, b_1 \in R^d$  are the attention parameters, trainable parameter matrix and bias vector,  $\tanh(*)$  is the activation function to prevent the gradient disappearance/explosion problem that may occur during training, the attention coefficients are normalized by the *softmax* function, and the normalized attention coefficients are expressed as:

$$\gamma_i^G = \frac{\exp(\mu_i^G)}{\exp(\mu_i^G) + \exp(\mu_i^L)} = 1 - \gamma_i^L \tag{13}$$

Finally, by linearly combining global and local information through the attention coefficients, the final representation can be calculated as follows:

$$x_i = \gamma_i^G x_i^G + \gamma_i^L x_i^L \tag{14}$$

### 3.5 Session feature fusion and prediction module

Session representation can be obtained by aggregating the features of items in a session. It has been shown [40, 52] that each item in a session is influenced by position and that introducing reverse positional embeddings in a session plays a more important role in capturing user preferences. Therefore, we introduce an attention mechanism with position embedding to capture the importance of items at different position when computing the session representation. The position embedding is initialized as  $P = \{p_1, p_2, \dots, p_m\}, p_1, p_1, p_2, \dots, p_m \in R^d$ , and  $m$  is the maximum session length. The initialized position embedding is connected to the item features and linearly transformed as follows:

$$x_{j,i}^* = \tanh(W_5[x_{j,i} || p_{m_j-i+1}] + b_2) \tag{15}$$

where  $x_{j,i}, x_{j,i}^*$  denote the representation before and after the transformation of item  $i$  in session  $j$ , respectively,  $W_5 \in R^{d \times 2d}, b_2 \in R^d$  are the trainable parameter matrix and the bias vector, respectively, and  $m_j$  denotes the length of session  $j$ . The  $||$  indicates a vector splice operation.

The attention coefficient is calculated as follows:

$$s'_j = \frac{1}{m_j} \sum_{i=1}^{m_j} x_{j,i} \tag{16}$$

$$\theta_{j,i} = q_2^T \sigma(W_6 s'_j + W_7 x_{j,i}^* + b_3) \tag{17}$$

where  $q_2 \in R^d$ ,  $W_6 \in R^{d \times d}$ ,  $W_7 \in R^{d \times d}$ ,  $b_3 \in R^d$  are the attention parameters, trainable parameter matrix and bias vector, respectively,  $\sigma(*)$  denotes the *sigmoid* activation function, and  $s'_j$  is the average of all items within the session.

According to the calculated attention coefficients, the importance of different items for session features can be distinguished, so that a linear combination of item features can be obtained for session features as:

$$s_j = \sum_{i=1}^{m_j} \theta_{j,i} x_{j,i} \quad (18)$$

The final score is calculated by multiplying the discourse representation  $s_j$  with the candidate item embedding  $x_i$  and outputting the probability through the *softmax* function:

$$\hat{z}_{j,i} = s_j^T x_i \quad (19)$$

$$\hat{y}_j = \text{softmax}(\hat{z}_j) \quad (20)$$

Finally, the  $K$  items with the highest scores in the results are created into a recommendation list and recommended to users.

For each session  $s_j$ , the loss function is defined as follows:

$$\mathcal{L} = -\sum_i^N (y_{j,i} \cdot \log(\hat{y}_{j,i}) + (1 - y_{j,i}) \cdot \log(1 - \hat{y}_{j,i})) \quad (21)$$

where  $\hat{y}_{j,i}$  denotes the probability that item  $i$  is in the recommendation list of session  $s_j$ .  $y_{j,i} = 1$  means that item  $i$  does exist in session  $s_j$ , otherwise  $y_{j,i} = 0$ .

## 4 Experiment

The model was experimented on three datasets, Tmall, Diginetica, and RetailRocket, respectively, and the following three questions were analyzed to verify the validity of the proposed FHGIGC model based on the experimental results:

- Whether FHGIGC outperforms the baseline model on real datasets.
- Whether the proposed FHGIGC can improve the recommended performance and what is the impact of each module on the model.
- How different hyperparameter settings affect the overall prediction effect of the model.

In this section, the dataset used in the experiment and how the data were processed are first introduced, then the evaluation metrics used and the parameters set in the experiment are given, and finally the experimental results are analyzed.

### 4.1 Dataset

**Tmall:** This dataset was first proposed in the IJCAI-15 competition and includes user shopping records from the Tmall shopping site from six months before the Double 11 to the day of the Double 11.

**Diginetica:** This dataset was proposed in the CIKM Cup 2016 and is session information extracted from e-commerce search engine logs, using only transaction data in this experiment.

**RetailRocket:** This dataset is a competition dataset published on kaggle and contains 4.5 months of data from Russian online retailers.

For all the above datasets, we follow the processing of previous work [35, 39], Deleting sessions that contain only one item and removing items with a total count of less than five,

**Table 1** Statistics of datasets

	Tmall	Diginetica	RetailRocket
Clicks	818,479	982,961	710,586
Training sessions	351,268	719,470	433,638
Test sessions	25,898	60,858	15,132
Items	40,728	43,097	36,968
Average length	6.69	5.12	5.43

and expanding the dataset by dividing it, As for session  $S = \{v_1, v_2, \dots, v_t\}$ , divide it into  $\{[v_1], v_2\}, \{[v_1, v_2], v_3\}, \dots, \{[v_1, v_2, v_3, \dots, v_{m-1}], v_m\}$ , where the last item of each sequence is used as the label. The statistics of all three datasets are given in Table 1.

### 4.2 Metrics

In order to observe the recommended effects of the model more intuitively, the FHGIGC model was evaluated and compared with the baseline model by two ranking-based evaluation metrics.

**HR@K (Hit Rate):** Which indicates the proportion of correctly recommended items in the  $K$  recommendation list. When the test dataset size is  $N$ , HR@K is calculated as shown below:

$$HR@K = \frac{Hit_n}{N} \tag{22}$$

where  $Hit_n$  indicates the number of correctly recommended item targets in the  $K$  recommendation list.

**MRR@K (Mean reciprocal rank):** Which means the inverse of the ranking of the correct recommendation item target in the  $K$  recommendation list, for example, the target appears in the first position is scored as 1, the second position is scored as 0.5, and if the target does not appear in the  $K$  recommendation list, it is scored as 0. When the size of the test dataset is  $N$ , the formula for  $MRR@K$  is as follows:

$$MRR@K = \frac{1}{N} \sum_{i=0}^N \frac{1}{Rank(i)} \tag{23}$$

where  $Rank(i)$  denotes the rank of correctly recommended item target in the  $K$  recommendation list.

### 4.3 Baselines

The ability of the FHGIGC model to improve the recommended performance was demonstrated by comparing the results with those of the following ten models.

**POP:** Recommending the  $K$  most frequently occurring items in the dataset.

**Item-KNN[5]:** Recommends similar products by calculating the cosine similarity between items.

**FPMC[8]:** Combines matrix decomposition with first-order Markov chain method to generate sequence-based recommendations.

**GRU4REC[13]:** Compute the session feature representation by gated recurrent network by treating the user’s current session as a sequence.

NARM[35]: Improves GRU4REC by adding an attention layer to capture important features in a session.

STAMP[36]: Replaces recurrent neural networks with multiple attention mechanisms and captures short-term preferences at the last node of the session.

SR-GNN[39]: Learns feature representations in graph neural networks by modeling multiple sessions as directed graphs, and fuses feature representations of multiple items in a session by an attention mechanism.

GC-SAN[43]: Combining graph neural networks with self-attention to learn long-term dependencies of items in a session.

SHARE[47]: Setting contextual windows on a single-session sequence to capture complex interactions between items within a session.

DHCN[26]: Constructing hypergraphs and line graphs to compute item as well as session feature representations through hypergraph neural networks, respectively, and learning the mutual information between them through a comparative learning approach.

#### 4.4 Parameter setting

To be fair, the embedding vector length as well as the batch size were set to 100 in the experiments following other research conventions. All learnable parameters are initialized by a Gaussian distribution with mean 0 and variance 0.1. In the optimizer settings, the Adam optimizer is selected and the learning rate is initialized to 0.001, decaying by 0.1 every three epochs, and trained for 30 epochs iteratively. Experiments show that the level of the hypergraph network and the size of the sliding window have different effects on the recommendation ability of the model under different datasets. For the baseline model, when both its evaluation metrics and datasets are the same as in this paper, the best results in the original experiment are used as a comparison; otherwise, the parameters are set to those of the best performance of the model for the experiment and comparison.

#### 4.5 Experiment result

The results are presented in Table 2 through the experiments, and it can be seen that the FHGIGC model proposed in this paper outperforms the baseline model on these three datasets. Specifically, we analyze the experimental results by answering the three questions posed above in 4.5, 4.6, and 4.7, respectively.

As a comparison, the POP in the traditional method only recommends the most frequently occurring items in the session data to the users without considering the user preferences, and has the worst effect. The Item-KNN method recommends by calculating the item similarity, but does not consider the sequential structure in the session data, so it does not achieve better results. FPMC, by combining matrix decomposition and Markov chain methods, has better results than the Item-KNN in both Tmall and RetailRocket datasets outperform Item-KNN, which proves to some extent the importance of sequential interaction information for obtaining user preferences.

GRU4REC is the first model to apply RNN to the field of session-based recommendation, but the improvement of its recommendation accuracy is quite limited because GRU4REC only considers the sequential relationships in a session and does not fully take into account the preference changes at different positions in the sequence. NARM and STAMP integrate attention mechanisms into RNN methods in order to more accurately capture shifts in user preferences. Among them, STAMP demonstrates the importance of short-term preferences

**Table 2** Experimental results of FHGIGC and comparison model, where FHGIGC is indicated by **bolded**

Method	Tmall			Diginetica			RetailRocket					
	H10	H20	M10	M20	H10	H20	M10	M20	H10	H20	M10	M20
POP	1.67	2.00	0.88	0.90	0.76	1.18	0.26	0.28	1.59	1.97	0.44	0.75
Item-KNN	6.65	9.15	3.11	3.31	25.07	35.75	10.77	11.57	22.48	28.97	10.43	10.98
FPMC	13.10	16.06	7.12	7.32	15.43	22.14	6.20	6.66	25.99	32.37	13.38	13.82
GRU4REC	9.47	10.93	5.78	5.89	17.93	30.79	7.73	8.22	38.35	44.01	23.27	23.67
NARM	19.17	22.30	10.42	10.70	35.44	48.32	15.13	16.00	42.07	50.22	24.59	24.88
STAMP	22.63	26.47	13.12	13.36	33.98	46.62	14.26	15.13	42.95	50.96	24.61	25.17
SR-GNN	23.41	27.57	13.45	13.72	38.42	51.26	16.89	17.78	43.21	50.32	26.07	26.57
GC-SAN	15.75	19.28	8.64	8.89	37.86	50.84	16.89	17.79	44.1	51.18	26.92	27.4
SHARE	25.04	29.69	13.98	14.24	39.52	52.73	17.12	18.05	46.47	54.16	26.59	27.11
DHCN	25.16	30.46	13.92	14.28	39.78	53.09	17.44	18.35	47.93	55.75	28.28	28.82
FHGIGC	27.31	32.79	15.53	15.89	40.32	53.71	17.54	18.46	48.36	56.13	28.93	29.46
Improve	8.55%	7.65%	11.57%	11.27%	1.36%	1.17%	0.57%	0.60%	0.90%	0.68%	2.30%	2.22%

in predicting users' next clicked item by replacing the RNN layer with multiple attention mechanisms and using the last item in the session as the user's short-term preference.

With the results of SR-GNN and GC-SAN, we can observe that the graph neural network approach outperforms the traditional and deep learning methods in terms of recommendations. This is because the graph structure approach is more conducive to capturing the transformation relationships of pairs of items between sessions. However, the traditional graph structure cannot effectively capture the higher-order relationships between multiple nodes. SHARE and DHCN model the session data through hypergraphs, and the experimental results fully demonstrate the advantages of hypergraph structure in session-based recommendation.

The FHGIGC model proposed in this paper combines global and local information based on all sessions to improve the accuracy of recommendations. In contrast to SHARE, the model effectively learns item consistency and relevant session information from the full session. In addition, unlike DHCN, the model learns the contextual features of items in the current session through the local hypergraph structure.

#### 4.6 Ablation experiment

In order to verify the effect of different modules in the proposed FHGIGC model on the recommended results, ablation experiments were performed by designing four variants of the model, and the four variants are described as follows.

**FHGIGC-G:** In this variant, the global item feature extraction module is removed from the model, and the item features learned in the item context interaction feature extraction module are used as the final item features, and this variant is used to verify the effectiveness of the global item feature extraction module.

**FHGIGC-L:** In this variant, the item context interaction feature extraction module is removed from the model, and the item features learned in the global item feature extraction module are used as the final item features, and this variant is used to verify the effectiveness of the item context interaction feature extraction module.

**FHGIGC-C:** The item feature fusion module is removed from the model in this variant to combine the global hypergraph with the local hypergraph by direct summation, and this module verifies that global features have different importance from local features.

**FHGIGC-P:** In this variant, the positional embedding of the session feature fusion module is removed, and the session features are obtained by directly fusing the item features in the session, and this module verifies the different effects of items in different position in the session on the session features.

The experimental results of the four variants of the model are shown in Fig. 4, and according to the obtained experimental results, it can be observed that the proposed FHGIGC achieves the best results. FHGIGC-G and FHGIGC-L show a significant decrease in model performance compared to FHGIGC, which indicates the effectiveness of learning item features jointly through multiple structures for prediction tasks. Meanwhile, the performance degradation of FHGIGC-G is more significant relative to FHGIGC-L, which indicates that the relevant sessions can provide more accurate and rich information for the recommendation task. Comparing FHGIGC with FHGIGC-C, it can be concluded that the information learned from the two structures does not have the same degree of influence on the recommended characters. The FHGIGC results outperformed the FHGIGC-P, which illustrates that items in different position have different degrees of influence on the session as a whole,

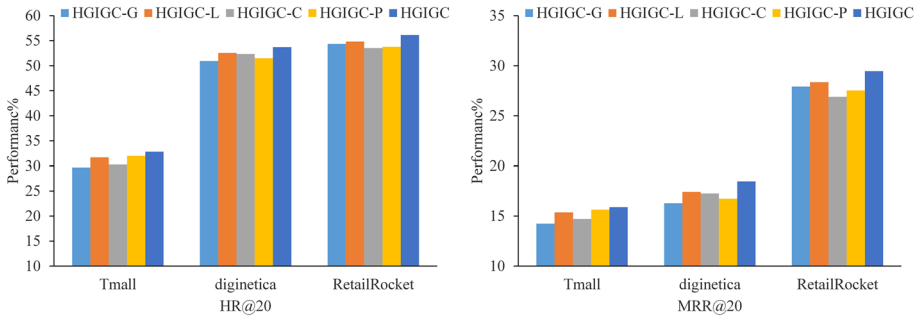


Fig. 4 Comparison of experimental results of the FHGIGC model and the proposed multiple variants

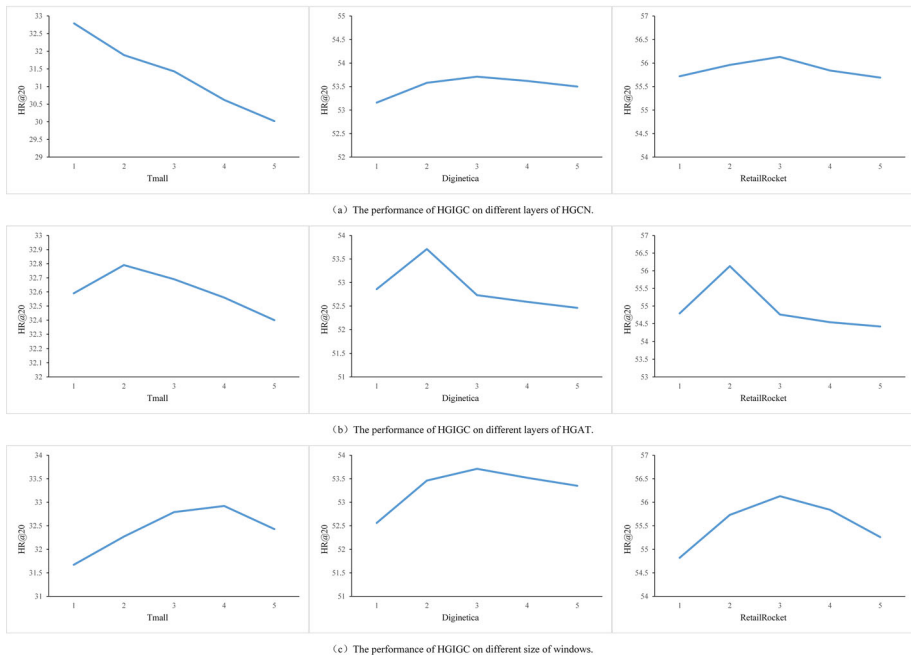


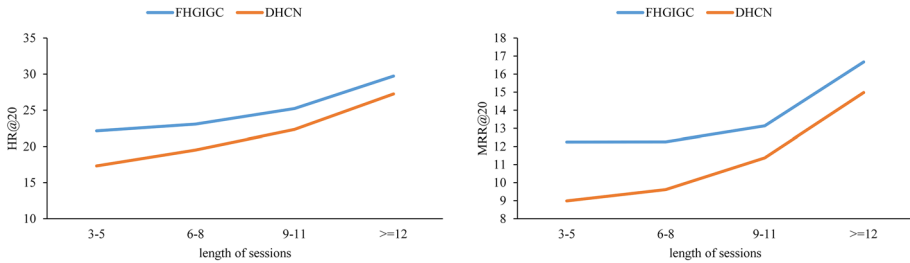
Fig. 5 Comparison of experimental results obtained by FHGIGC model when setting different parameters

validating the role of position embedding in capturing the importance of items in different position.

### 4.7 Parameter experiment

To explore the best performance of the FHGIGC model parameter experiments were conducted on each of the three datasets to find the best parameters by grid search, and the experimental results are shown in Fig. 5.

Number of layers of HGICN: The layers of the hypergraph convolutional neural network are set to 1, 2, 3, 4, 5, respectively. From the results in Fig. 5a, it can be observed that on the Tmall dataset, the best results are obtained when the number of layers is 1, and the model



**Fig. 6** Experimental results of the model with session samples of different lengths

becomes less and less effective when the number of layers rises. And on the Diginetica and RetailRocket datasets, the best results can be achieved when the hypergraph convolutional neural network level is set to 3. For the above results, the possible reason for the analysis is that in the Tmall dataset, where the average length of sessions is longer, a single-layer neural network can better capture the session consistency in sessions. For the Diginetica and RetailRocket datasets with shorter average lengths, a deeper layer of neural network needs to be set up so that the model can add information from the relevant sessions.

**Number of layers of HGAT:** The hypergraph attention network layers are set to 1, 2, 3, 4, 5, respectively. The results in Fig. 5b show that the best results can be obtained when the attention network is set to two layers, and the final prediction of the model decreases instead of rising when the number of attention layers is continued to be raised. This is because the deepening of the graph neural network layers may make the information between nodes too smooth.

**Size of context window:** The context windows are set to 1, 2, 3, 4, 5, respectively. The results are displayed in Fig. 5c. When the maximum window size is set to 1, each hyperedge contains only one node, so that there is no information propagation between nodes. And when the maximum window is set to 2, each hyperedge contains two nodes, in which case the hypergraph structure degenerates to an ordinary graph structure. From the results, it can be seen that larger window sizes achieve better results on the Tmall dataset, possibly because the average session length in the Tmall dataset is larger, allowing multiple hyperedges to be constructed with larger window sizes.

## 4.8 Case study

In this section, we further explore the ability of the proposed method to capture user preferences in sessions of different lengths. On the Tmall dataset, the dataset is firstly categorized into four groups according to their lengths, and the session lengths of each group are 3–4, 5–8, 9–11, and greater than 12. Experiments are conducted on each of the four groups for FHGIGC and DHCN, and the results of the experiments are displayed in Fig. 6. From the results in the Fig. 6, it can be seen that FHGIGC performs more stably in sessions of different lengths, and outperforms DHCN in the short-session experiments, which indicates that the proposed method of fusing global and contextual information can effectively improve the recommendation performance. In addition, the experimental results respond that the model recommendation performance is higher when the session data are longer, which is because the items in the global graph are more tightly connected to learn richer information when the session length increases, and more hyperedges can be constructed for the current session for feature propagation through the context window set in the local graph. Therefore, our



proposed method can effectively capture the user preferences reflected by the session for recommendation tasks.

## 5 Conclusion

In this paper, we propose a hypergraph neural network model FHGIGC for the session-based recommendation task, which can learn relevant session information as well as item consistency information from the global hypergraph and capture the contextual features of items in a session from the local hypergraph, respectively. The learning item features are combined through both structures and these features are integrated through an attention layer. The experiments show that the proposed model outperforms the comparison model on all three datasets, demonstrating that the FHGIGC model can effectively solve the problem of missing item features within a session in existing session-based recommendation models and mitigate the negative impact of information scarcity in short sessions on the recommendation effect. Current research in session-based recommendation relies exclusively on session data for recommendation. In future work, we consider constructing item as well as spatiotemporal information in session data through heterogeneous graphs, and introducing generative adversarial networks to capture dynamic feature changes in these data.

**Author Contributions** XC contributed to conceptualization; methodology; and validation; XC and TL contributed to writing-original draft preparation; XC, XH and MZ contributed to writing-review and editing; and XH and MZ supervised the study. All authors have read and agreed to the published version of the manuscript.

**Funding** Not applicable.

**Data Availability Statement** All datasets used in this article are open source and can be accessed from the following links: <https://tianchi.aliyun.com/dataset/42> <http://cikm2016.cs.iupui.edu/cikm-cup/> <https://www.kaggle.com/datasets/retailrocket/ecommerce-dataset>.

## Declarations

**Conflict of interest** The authors declare that they have no competing interest.

**Ethics approval** Not applicable.

## References

1. Sarwar B, Karypis G, Konstan J, Riedl J (2000) Analysis of recommendation algorithms for e-commerce, pp 158–167
2. He C, Parra D, Verbert K (2016) Interactive recommender systems: a survey of the state of the art and future research challenges and opportunities. *Expert Syst Appl* 56:9–27
3. Bian Z, Zhou S, Fu H, Yang Q, Sun Z, Tang J, Liu G, Liu K, Li X (2021) Denoising user-aware memory network for recommendation, 400–410
4. Chang J, Gao C, Zheng Y, Hui Y, Niu Y, Song Y, Jin D, Li Y (2021) Sequential recommendation with graph neural networks, 378–387
5. Sarwar B, Karypis G, Konstan J, Riedl J (2001) Item-based collaborative filtering recommendation algorithms, 285–295
6. Schafer JB, Frankowski D, Herlocker J, Sen S (2007) Collaborative filtering recommender systems. *The adaptive web: methods and strategies of web personalization*, 291–324
7. Park C, Kim D, Oh J, Yu H (2017) Do "also-viewed" products help user rating prediction? In: *Proceedings of the 26th international conference on world wide web*, pp 1113–1122

8. Rendle S, Freudenthaler C, Schmidt-Thieme L (2010) Factorizing personalized markov chains for next-basket recommendation, pp 811–820
9. Zimdars A, Chickering DM, Meek C (2013) Using temporal data for making recommendations. arXiv preprint [arXiv:1301.2320](https://arxiv.org/abs/1301.2320)
10. Mandal S, Maiti A (2018) Explicit feedbacks meet with implicit feedbacks: a combined approach for recommendation system. In: International conference on complex networks and their applications, pp 169–181. Springer
11. Mandal S, Maiti A (2020) Explicit feedback meet with implicit feedback in gpmf: a generalized probabilistic matrix factorization model for recommendation. *Appl Intell* 50(6):1955–1978
12. Mandal S, Maiti A (2021) Rating prediction with review network feedback: a new direction in recommendation. *IEEE Trans Comput Soc Syst* 9(3):740–750
13. Hidasi B, Karatzoglou A, Baltrunas L, Tikk D (2015) Session-based recommendations with recurrent neural networks. arXiv preprint [arXiv:1511.06939](https://arxiv.org/abs/1511.06939)
14. Tan YK, Xu X, Liu Y (2016) Improved recurrent neural networks for session-based recommendations, 17–22
15. Yin H, Wang W, Wang H, Chen L, Zhou X (2017) Spatial-aware hierarchical collaborative deep learning for poi recommendation. *IEEE Trans Knowl Data Eng* 29(11):2537–2551
16. Mandal S, Maiti A. Deep collaborative filtering with social promoter score-based user-item interaction: a new perspective in recommendation. *Appl Intell*, 1–26
17. Fan W, Ma Y, Li Q, He Y, Zhao E, Tang J, Yin D (2019) Graph neural networks for social recommendation. In: The world wide web conference, pp 417–426
18. Qiu R, Yin H, Huang Z, Chen T (2020) Gag: global attributed graph neural network for streaming session-based recommendation. In: Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval, pp 669–678
19. Wu Z, Pan S, Chen F, Long G, Zhang C, Philip SY (2020) A comprehensive survey on graph neural networks. *IEEE Trans Neural Netw Learn Syst* 32(1):4–24
20. Zhou J, Cui G, Hu S, Zhang Z, Yang C, Liu Z, Wang L, Li C, Sun M (2020) Graph neural networks: a review of methods and applications. *AI Open* 1:57–81
21. Mandal S, Maiti A (2021) Graph neural networks for heterogeneous trust based social recommendation. In: 2021 international joint conference on neural networks (IJCNN), pp 1–8. IEEE
22. Pang Y, Wu L, Shen Q, Zhang Y, Wei Z, Xu F, Chang E, Long B, Pei J (2022) Heterogeneous global graph neural networks for personalized session-based recommendation, 775–783
23. Deng ZH, Wang CD, Huang L, Lai JH, Philip SY (2022) G<sup>+</sup> 3sr: Global graph guided session-based recommendation. *IEEE Trans Neural Netw Learn Syst* (2022)
24. Wang H, Zeng Y, Chen J, Zhao Z, Chen H (2022) A spatiotemporal graph neural network for session-based recommendation. *Expert Syst Appl* 202:117114
25. Feng Y, You H, Zhang Z, Ji R, Gao Y (2019) Hypergraph neural networks. *Proc AAAI Conf Artif Intell* 33(01):3558–3565
26. Xia X, Yin H, Yu J, Wang Q, Cui L, Zhang X (2021) Self-supervised hypergraph convolutional networks for session-based recommendation. *Proc AAAI Conf Artif Intell* 35(5):4503–4511
27. Ding C, Zhao Z, Li C, Yu Y, Zeng Q (2023) Session-based recommendation with hypergraph convolutional networks and sequential information embeddings. *Expert Syst Appl* 223:119875
28. He X, Liao L, Zhang H, Nie L, Hu X, Chua TS (2017) Neural collaborative filtering. In: Proceedings of the AAAI conference on artificial intelligence 173–182 (2017)
29. Chen W, Cai F, Chen H, Rijke MD (2019) Joint neural collaborative filtering for recommender systems. *ACM Trans Inf Syst (TOIS)* 37(4):1–30
30. Shani G, Heckerman D, Brafman RI, Boutilier C (2005) An mdp-based recommender system. *J Mach Learn Res* 6(9)
31. Wang S, Hu L, Wang Y, Cao L, Sheng QZ, Orgun M (2019) Sequential recommender systems: challenges, progress and prospects. arXiv preprint [arXiv:2001.04830](https://arxiv.org/abs/2001.04830)
32. Quadrana M, Karatzoglou A, Hidasi B, Cremonesi P (2017) Personalizing session-based recommendations with hierarchical recurrent neural networks, 130–137
33. Jannach D, Ludewig M (2017) When recurrent neural networks meet the neighborhood for session-based recommendation, 306–310
34. Song J, Shen H, Ou Z, Zhang J, Xiao T, Liang S (2019) Islf: Interest shift and latent factors combination model for session-based recommendation 5765–5771
35. Li J, Ren P, Chen Z, Ren Z, Lian T, Ma J (2017) Neural attentive session-based recommendation, 1419–1428
36. Liu Q, Zeng Y, Mokhosi R, Zhang H (2018) Stamp: short-term attention/memory priority model for session-based recommendation, 1831–1839

37. Peng D, Yuan W, Liu C (2019) Harsam: a hybrid model for recommendation supported by self-attention mechanism. *IEEE Access* 7:12620–12629
38. Zhang P, Guo J, Li C, Xie Y, Kim JB, Zhang Y, Xie X, Wang H, Kim S (2023) Efficiently leveraging multi-level user intent for session-based recommendation via atten-mixer network, 168–176
39. Wu S, Tang Y, Zhu Y, Wang L, Xie X, Tan T (2019) Session-based recommendation with graph neural networks. *Proc AAAI Conf Artif Intell* 33(01):346–353
40. Qiu R, Huang Z, Chen T, Yin H (2021) Exploiting positional information for session-based recommendation. *ACM Trans Inf Syst (TOIS)* 40(2):1–24
41. Qiu R, Li J, Huang Z, Yin H (2019) Rethinking the item order in session-based recommendation with graph neural networks, 579–588
42. Qiu R, Huang Z, Li J, Yin H (2020) Exploiting cross-session information for session-based recommendation with graph neural networks. *ACM Trans Inf Syst (TOIS)* 38(3):1–23
43. Xu C, Zhao P, Liu Y, Sheng VS, Xu J, Zhuang F, Fang J, Zhou X (2019) Graph contextualized self-attention network for session-based recommendation. *IJCAI* 19:3940–3946
44. Wu S, Zhang Y, Gao C, Bian K, Bian B (2020) Garg: anonymous recommendation of point-of-interest in mobile networks by graph convolution network. *Data Sci Eng* 5:433–447
45. Chen T, Wong RCW (2020) Handling information loss of graph neural networks for session-based recommendation. In: *Proceedings of the 26th ACM SIGKDD International conference on knowledge discovery and data mining*, 1172–1180 (2020)
46. Xia X, Yin H, Yu J, Shao Y, Cui L (2021) Self-supervised graph co-training for session-based recommendation. In: *Proceedings of the 30th ACM international conference on information & knowledge management*, pp 2180–2190
47. Wang J, Ding K, Zhu Z, Caverlee J (2021) Session-based recommendation with hypergraph attention networks, 82–90 (2021). *SIAM*
48. Wu F, Souza A, Zhang T, Fifty C, Yu T, Weinberger K (2019) Simplifying graph convolutional networks, 6861–6871 (2019). *PMLR*
49. Bai S, Zhang F, Torr PH (2021) Hypergraph convolution and hypergraph attention. *Pattern Recogn* 110:107637
50. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I (2017) Attention is all you need. *Adv Neural Inf Process Syst* 30 (2017)
51. Kang WC, McAuley J (2018) Self-attentive sequential recommendation, 197–206. *IEEE*
52. Wang Z, Wei W, Cong G, Li XL, Mao XL, Qiu M (2020) Global context enhanced graph neural networks for session-based recommendation, 169–178

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

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.