Multi-hypergraph Neural Network with Fusion of Location Information for Session-based Recommendation

Ziang Li, Jie Wu*, Guojing Han, Chi Ma, Yuenai Chen

Abstract—**Session recommendation is a technology that processes data referring to the user-anonymous session behavior analysis. This technique has a wide range of applications in e-commerce, videos, social networks, and other fields and it is a hot topic in current recommendation system research. Multi-hypergraph neural network with fusion of location information for session-based recommendation (MHGNN-LI) captures multi-dimensional interest features in the session through the multi-hypergraph representation and combines different contrastive learning methods to alleviate the problem of data sparsity. It also considers the dependence order of the recommended items to improve the recommendation effect. Concerning the methodology of work, the model first constructs a session hypergraph to obtain the interest features of multiple items in the session. Secondly, it decouples the interest features into global and local features within the session, and then it integrates the item features with location information into the local interest features. Finally, it introduces a line graph at the feature level between the different sessions and maximizes the mutual information between sessions and within them, through contrastive learning, to enhance the expression ability of session data. As for the validation, comparative experiments, with related models on two public datasets, Tmall and Diginetica, show that the MHGNN-LI model has improved on P@10 and MRR@10 by, at least, 15.47% and 20.51%, as well as 9.77% and 12.37% on the Tmall and Diginetica datasets, respectively.**

Index Terms—**session-based recommendation, hypergraph, feature disentangle, contrastive learning**

I. Introduction

In the era of overloaded internet information, recommendation systems have become an important tool to help users in quickly obtaining useful information from complex data. The original recommendation system relied on the user's personal information and past historical behavior data. However, most of this information is difficult to access, and just anonymous behavioral records during the ongoing session are available. Therefore, the Session-Based Recommendation (SBR) methods, based on sessions, have received widespread attention. As a definition, a session represents a behavioral record of the user interaction with an item over a continuous time [1], such as items purchased within an hour. Moreover, SBR methods do not require detailed analysis of the user identity information and the historical data; however, they only need to extract preference information from the interaction records in the current session to perform the required prediction.

In recent years, several studies have attempted to use neural networks to solve the problem of the user behavior time factors in SBR [2]; however, the most critical issue is to define the way to accurately capture the complex transitions between items when having limited information. Recurrent Neural Network (RNN) [3] is a neural network model that can learn sequence data and has a wide range of applications in various fields. Moreover, the Gated Recurrent Unit for Recommendation (GRU4REC) [4] is the first RNN-based recommendation model that monitors the user click behavior as a sequence and deploys the RNN's memory capabilities for a recommendation. Furthermore, the Neuro Affective Relational Model (NARM) [5] captures the user's current sequential behavior through the combination of RNN and attention mechanisms. In addition, to capture the complex transition relationship between items in the SBR task, some models adopt the Graph Neural Network (GNN) method [6], which constructs a session graph to represent the dependency relationship between items and uses GNN's powerful modeling ability to achieve the item embedding representation. However, these GNN-based recommendation models only adopt the user interaction records in short-term sessions and ignore the long-term historical behavioral data values.

Therefore, to fully utilize the long-term historical behavioral data, the Session-based Recommendation with Graph Neural Networks (SR-GNN) method [7] proposes a technique based on constructing a session graph. This later is also based on the user's historical session sequence and deploys the Gate Graph Neural Network (GGNN) [8] to effectively model the item transitions. Furthermore, the Self-Supervised Hypergraph Convolutional Networks for Session-based Recommendation (DHCN) [9] uses hypergraph convolutional networks for item transformation and captures high-order correlations between items. As for the HIDE model [10], it captures high-order relationships and latent interests in item transformations through hypergraphs and separates these interests at macro and micro levels. Currently, several studies use GNN and Hypergraph Neural Network (HNN) models to represent the item pattern transformations and achieve good results in SBR tasks. However, these models still show the following two shortcomings:

Manuscripts received April 3, 2023; revised September 7, 2023. This work is supported by Foundation of Guangdong Educational committee under Grant the No.2022ZDZX4052, 2021ZDJS082, No.2019KQNCX148.

Ziang Li is a postgraduate student of School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan, 114051, China (email:liziang0811@foxmail.com).

Jie Wu is an associate professor of School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan, 114051, China (*corresponding author to provide email:wujieaa@163.com).

Guojing Han is an associate senior engineer of Anshan Meteorological Service, Anshan 114004, China (email:314795969@qq.com).

Chi Ma is an associate professor of School of Computer Science and Engineering, Huizhou University, Huizhou 516007, China (email:machi@hzu.edu.cn).

Yuenai Chen is a professor of School of Computer Science and Engineering, Huizhou University, Huizhou 516007, China (email:cyn@hzu.edu.cn).

(1) They only optimize information between sessions and within sessions, respectively, without fully representing the learning outcomes from a global perspective;

(2) When extracting the user feature information from different sessions, they do not enhance the ability of SBR models through the construction of different session views.

A novel method (named MHGNN-LI) is proposed to solve such problems. Moreover, the hypergraphs are used as inputs to the encoder to convert high-order interests using three types of hyperedges. The proposed model uses two channels to describe the internal and the inter-session information, respectively. In addition, the contrastive learning is introduced to solve the problem of data sparsity in hypergraph modeling. Therefore, self-supervised learning maximizes the mutual information between both channels to obtain new information and improve their performance in different sessions. To sum up, the internal recommendation task of the session and the inter-session recommendation task are jointly optimized under the main and the auxiliary learning frameworks, respectively, which helps the recommendation task in achieving better performance indicators.

Finally, the main innovations of this model are as follows:

(1) Multi-hypergraph neural network with fusion of location information for session-based recommendation is proposed. This model is able to capture cross-session information beyond pairwise relationships between items in a single session through hypergraph modeling and dynamically optimize the internal and inter-session, in two different views, through contrastive learning;

(2) Considering the dependency relationship between the item sequences, the Gate Recurrent Unit (GRU) is deployed to learn the sequence representation of the items; moreover, it is fused with the user's interest features;

(3) Line graphs are introduced into self-supervised task network training to construct the inter-session contrastive learning in order to enhance the hypergraph modeling effect and alleviate data sparsity problems;

II. Related Work

A. Session-based Recommendation System

Traditional recommendation models, such as Collaborative Filtering (CF) based models [11], [12], originate the general preferences of users by decomposing the user-item matrix; however, this technique cannot capture the user's interest transfer. Moreover, SBR consists of understanding the subsequent behavior based on the continuous behavior of users in a given session. The whole sessions do not usually include the login information and are used to understand the user preferences through long-term historical behavior. For instance, Hidasi et al. proposed the GRU4REC model, which uses first the RNN method to solve session-based recommendation problems. The model uses RNN and GRUs to represent the session data and can learn session-level representations by given historical interactions of the sessions and then predict the user's next behavior. Moreover, NARM simulates the user sequential behavior and models user behavior through RNN and attention mechanisms, capturing the main intent of users in the current session. As for STAMP [13], it obtains the general interest of users by considering the short-term preference of the last click and the long-term preference of the session context. Finally, the Hierarchical RNN (HRNN) model [14] integrates a GRU layer to expand the RNN model sequence, and captures the dynamic preferences by transferring information across sessions.

B. Session Recommendation Based on Graph Neural Network

In recent years, some studies have introduced GNN into the recommendation systems [15], [16], [17] to learn the representation of graph-structured data; therefore, they designed different GNN models. For example, GCMC [15] uses a neural graph autoencoder to reconstruct the user-item rating graph. As for the Neural Graph Collaborative Filtering (NGCF) [16], it proposes constructing a user-item bipartite graph and capturing the collaborative signal between users and items using a multi-layer GNN. These methods represent some models that optimize the behavior of users between sessions separately. Unlike the previous SBR methods, GNN further identifies hidden information between items by modeling the session graph. Moreover, the SR-GNN model represents the session sequence into a graph and combines the sessions' general and current interests for better prediction. Furthermore, the Folklore Graph Neural Networks (FGNN) [18] considers the potential order of items in sessions and uses weighted graph attention layers as well as Readout functions to identify the potential information between items in sessions. Finally, the Target Attentive Graph Neural Network (TAGNN) model [19] extends SR-GNN by integrating the target attention networks into the model to capture different interests of users in sessions.

C. Hypergraph Learning

Hypergraphs represent a natural way to connect complex high-order relationships. With the development of Deep Learning (DL) technology, HNNs have also received widespread attention. Early research regarding HGNN [20] and HyperGCN [21] applied graph convolution to hypergraphs and proposed Dynamic Hypergraph Neural Networks (DHNNs) [22] and Linear Hypergraph Convolutional Networks (LHCNs) [23]. Meanwhile, some research combining hypergraph learning with recommendation systems, such as HyperRec [24], which uses hypergraphs to model short-term user preferences for predicting items. Such methods did not fully utilize the information between hyperedges and were not explicitly designed for session-based scenarios. Furthermore, SHARE [25] used a hypergraph attention network model to describe the current interest of users by capturing edge information on each item and on each hyperedge in the context window. Finally, DHCN uses hypergraphs to capture high-order relationships of item transitions whereas HIDE deployed hypergraphs to capture latent intentions in items.

D. Self-supervised Learning

Self-Supervised Learning (SSL) is a learning method that does not require manual annotation as it trains models by utilizing the intrinsic structure and data rules. In particular, the contrastive learning [26], [27] can enhance the robustness of user representations. Therefore, to improve the design of recommendation systems, some researchers have adopted

SSL methods to enhance the effectiveness of the representation learning [28], [29], [30]. For instance, reference [28] proposed a multi-task SSL framework to recommend large-scale items and designed a data augmentation method from the feature correlation perspective. Reference [29] used three types of data for expansion and identified the nodes automatically as self-supervised tasks in order to improve the representation learning. Finally, Reference [30] used a self-supervised graph co-training framework to learn the session representations based on the session views and the item views; in addition, the researchers used iterative pseudo-labels as information self-supervision examples to improve the recommendation performance.

III. Preparation Work

A. Problem Definition

Let $V = \{v_1, v_2, \ldots, v_n\}$ represent the set of items and $\hat{y}$ be the number of items. Each item v_n in V is encoded into a unified embedding space $h_n \in R^d$, where d is the dimension of the embedded item. Let $v_{s,k} \in V\,(1 \le k \le n)$ be the user interactions with items in session $s = [v_{s,1}, v_{s,2}, \ldots, v_{s,n}]$ of length n. The objective of SBR is to identify the user preferences in session $s = \{v_{s,1}, v_{s,2}, \ldots, v_{s,k}\}$ from time 1 to k and predict the item $v_{s,k+1}$ that is most likely to be interacting at time $k+1$. Therefore, the model inputs session sequence s and the outputs recommendation scores $\hat{y} = \{\hat{y_1}, \hat{y_2}, \ldots, \hat{y_n}\}$ for all possible items are captured. Finally, N items with the highest scores in $\hat{y}$ are recommended as candidate items for session s.

B. Hypergraph Construction

A hypergraph is constructed from items in a sequence, and the links between sessions are connected by hyperedges. As for the hypergraph $G = (V, E)$, it is defined by V and E that represent the sets of nodes and hyperedges, respectively. Each hyperedge $e \in E$ contains two or more nodes and is assigned a weight value W_{ee}. Moreover, W is a diagonal matrix regrouping all weight values. Therefore, the entire hypergraph can be represented by a matrix $H \in R^{M \times N}$, where $H_{ie} = \begin{cases} 1 & e \in E \\ 0 & e \notin E \end{cases}$. The degree of each vertex and hyperedge is defined as the degree matrix $B_{ii} = \sum_{i=1}^{M} H_{ie}$ and $D_{ii} = \sum_{e=1}^{N} W_{ee} H_{ie}$, respectively. Finally, both B and D are diagonal matrices. An example of constructing a hypergraph is shown in Figure 1.

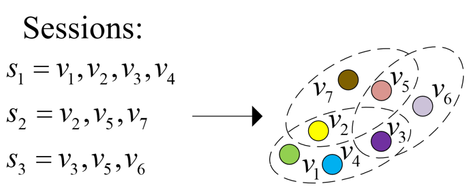

Fig. 1: Hypergraph Construction

C. Interest Embedding

The method, based on SBR, embeds item vectors into the same vector space to study the impact of the user interests on the effectiveness of the session recommendations. Inspired by reference [10], the item embedding of the model is divided into K parts. The interest embedding is initialized as $h_{v_i} = \left(h_{v_i}^1, h_{v_i}^2, \ldots, h_{v_i}^K\right)$ where $v_i \in V$ and $h_{v_i} \in R^d$ is the embedding vector of v_i with d being the size of embedding. Moreover, $h_{v_i}^k \in R^d$ represents the part of item v_i under the interest of k. In addition, the item embedding parts, for all K interests, are aggregated to construct the interest embedding vector, which is defined as $h_p^k = Mean\left(h_{v_i}^k | v_i \in V\right)$.

IV. MHGNN-LI Model

A. Overall Architecture of the Model

To sum up, the MHGNN-LI model consists of the following main modules, as shown in Figure 2:

(1) Session graph construction module: Construct a hypergraph based on the given sequence, then construct a hypergraph, within the session, based on the current session sequence; finally, construct a line graph of hypergraph between sessions based on the historical session sequence;

(2) Embedding location information extraction module: According to the given single session embedding into the gate recurrent unit, the item feature's representation with the position information is obtained;

(3) Intra-session information extraction module: Extract information from the constructed intra-session hypergraph using local and global feature disentangling to generate the intra-session feature representations;

(4) Inter-session information extraction module: Obtain the inter-session feature representations by averaging pooling and graph convolution of the hypergraph line graph to construct the inter-session and extract information to generate inter-session feature representations;

(5) Location information fusion module: Combine the feature representation relative to the position information with the features within the session, and use the reverse position embedding and the soft attention mechanism to obtain the final session feature representation with position information decoupled;

(6) Feature fusion module: The session feature representation with position information decoupled is subjected to a sum pooling operation with the inter-session feature representation; moreover, the session representation of the two views is identified through a self-supervised manner. Finally, the fused session representation with the position information is obtained;

(7) Prediction module: Combine linearly the learned session representations and the item session feature representations, obtained through self-supervised learning, to predict the next item.

B. Session Graph Construction Module

1) Session Hypergraph Construction: The sequence of conversations s can be modeled as a session hypergraph $G_s = (V_s, E_s)$, where $V_s \in V$ represents the set of items clicked by the user and E_s represents the set of hyperedges between the sessions and the items. Based on the idea in [9], three different hyperedges are used to capture higher-order relationships between items.

a. Transform Hyperedges E_s^t [10]. The time-order of item transformation is a key factor of SBR [31]. As shown in the

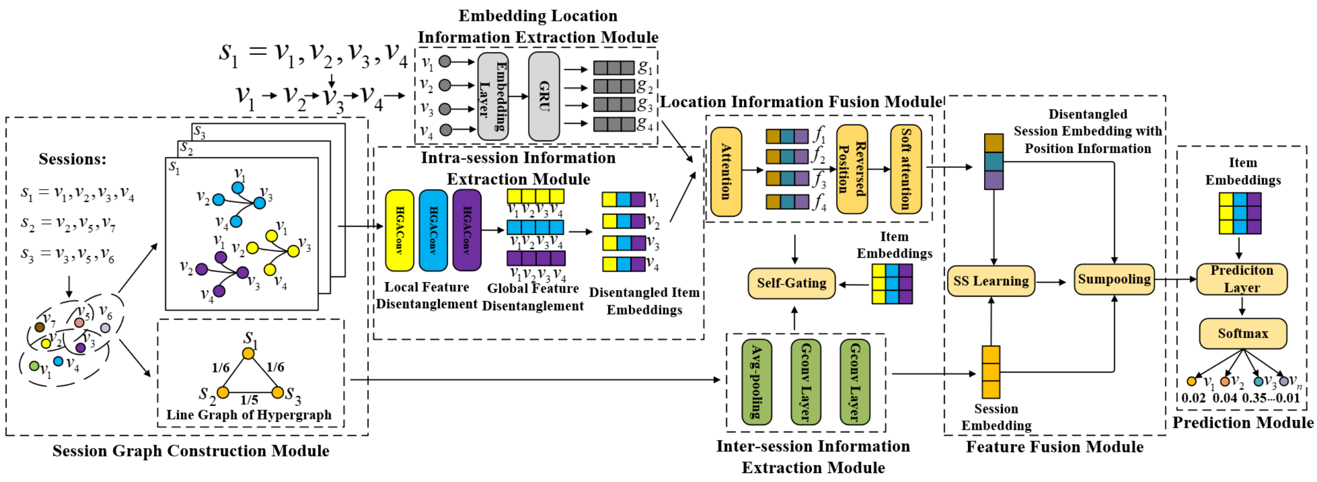

Fig. 2: MHGNN-LI Model Structure Framework

example of Figure 3(a), the transformation order of items in the session is maintained, and item v_2 is passed into the set of items $\{v_1, v_2, v_3\}$ and connected to the hyperedges, reflecting the higher-order correlations between items;

b. Context Hyperedges E_s^c [10]. The context order can reflect the user's potential interests. First, the sliding window w size is performed on the item sequence to capture the user's local interests. Then, hyperedges are used to connect the items in the window, represented as $e = \{v_1, v_2\} \in E_{s_2}^c$, where w is the size of the sliding window. Referring to the example of Figure 3(b), for different window sizes, user interests can be obtained from several items. Finally, hyperedges are collected from different sliding windows to represent $E_c^s = \bigcup_{w=1}^{W} E_{s_w}^c$;

c. Interest Hyperedges E_s^i. The interest of users who click on the same item will have unique similarities and can be built based on capturing correlations between interest items. As shown in the example of Figure 3(c), the cosine similarity between the kth interest and item $v_i \in V$, represented as $S_{ki} = cos\left(h_p^k, h_{v_i}^k\right)$ is calculated first. Then, each interest is represented as a hyperedge connecting the top-εn items with S_{ki}, where n shows the number of items in the session and ε represents the sparsity of the hyperedges. A set of hyperedges is generated for session s based on these three hyperedges types, represented as $E_s = E_s^t \bigcup E_s^c \bigcup E_s^i$.

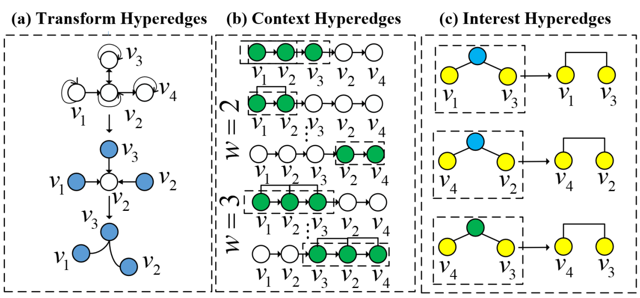

Fig. 3: Session Hypergraph Construction

2) Line Graph of Hypergraph Construction: According to the session graph, the line graph of hypergraph $L(G)$ is derived. In general, the line graph of a hypergraph is a simple graph in which two nodes of $L(G)$ correspond to, at least, one common hyperedge node. Nodes serve as the connecting points for each session, while shared nodes enable the interconnection between multiple sessions. Compared to the hypergraph of higher-order relationships within sessions, the line graph of the hypergraph describes the relationships between sessions. Finally, define $L(G) = (V_L, E_L)$, where $V_L = \{v_e | v_e \in E\}$, $E_L = \left\{(v_{e_x}, v_{e_y}) | e_x, e_y \in E, |e_x \bigcap e_y| \geq 1\right\}$, and assign a weight value $W_{x,y}$ to each edge (v_{e_x}, v_{e_y}), where $W_{x,y} = |e_x \bigcap e_y| / |e_x \bigcup e_y|$.

C. Embedding Location Information Extraction Module

In terms of sequence modeling, RNN has significant advantages. Moreover, GRU, as a variant of RNN, can effectively alleviate the problem of long-term dependence as it usually has fewer parameters and faster training speed than LSTM [32]. Therefore, a module, called embedding location information extraction module, is designed based on GRU to effectively capture the sequential dependence among the items in the session.

Let the hidden state be g_i and set the initial value to null. For the initial representation of item v_i c_i, the GRU updates the hidden state g_i as follows:

$$r_i = \sigma\left(W_r c_i + U_r g_{i-1}\right) \tag{1}$$

$$a_i = \sigma\left(W_a c_i + U_a g_{i-1}\right) \tag{2}$$

$$\tilde{g}_i = tanh\left(W_{\tilde{g}} c_i + U_{\tilde{g}}\left(r_i \odot g_{i-1}\right)\right) \tag{3}$$

$$g_i = (1 - a) \odot g_{i-1} + a_i \odot \tilde{g}_i \tag{4}$$

where W_r, W_a, $W_{\tilde{g}} \in R^{d \times d}$, U_r, U_a, and $U_{\tilde{g}} \in R^{d \times d}$ are learnable parameters. $\sigma(\cdot)$ is the sigmoid function. $\odot$ is an element-wise multiplication operation. The reset gate r_i represents the combination of current and previous information. The update gate a_i controls the proportion of previous information to current information. $\tilde{g}_i$ records the current state. By using GRU, we obtain a feature representation of items based on sequential embedding X_g.

D. Intra-session Information Extraction Module

The session data in SBR contains a lot of noise signals, which can affect the recommendation model accuracy. The items that the user clicks on during each session reflect the user's different interests. Therefore, this feature can be used to filter out noise. To better capture the user's interests, several parallel interest-aware encoders are used in the local feature encoding to separate the user's diverse interests; moreover, each encoder specifically captures specific interests. To make the encoder capture only a single interest, propagation is performed on a particular session hypergraph with the same interest, and all transitions and context hyperedges as well as interest hyperedges, contained in the hypergraph G_s^k under k interests in each session s, are used as inputs to represent the node features. According to the method presented in literature [10], HyperGraph Attention Convolution (HGAConv) is used to study the interest of items on the interest hypergraph G_s^k, which includes node-to-hyperedge (ne) and hyperedge-to-node (en) propagation. Both propagations are defined below:

(1) Node-to-hyperedge (ne) propagation. Nodes connected by hyperedges represent user interests; however, other nodes may be noisy. Therefore, an attention mechanism is used to aggregate nodes v_o in order to obtain the corresponding hyperedge features f_j^k for the k-th interest, represented as:

$$f_j^k = AGG_{ne}\left(\alpha_{j_o}^k h_{v_0}^k | v_o \in e_j\right) \tag{5}$$

where AGG_{ne} is an aggregation function and $\alpha_{j_o}^k$ represents the attention weight of node v_o in the hyperedge e_j. Assuming that nodes are connected through hyperedges e_j, they can be regarded as a set of items. Then, the average value of this item set is calculated and denoted as $h_{c_j}^k = Mean\left(h_{v_o}^k | v_o \in e_j\right)$. Moreover, the closer the node is to the clustering center, the closer it is to the user's interest. Thus, the $softmax$ function is used to normalize it and get the calculated attention weight as follows:

$$\alpha_{j_o}^k = \frac{exp\left(LeakyReLU\left(q_1^{k^{\mathrm{T}}}\left(h_{c_j}^k \circ h_{v_o}^k\right)\right)\right)}{\sum_{v_{0'} \in e_j} exp\left(LeakyReLU\left(q_1^{k^T}\left(h_{c_j}^k \circ h_{v_{o'}}^k\right)\right)\right)} \tag{6}$$

where $q_1^k \in R^{\frac{d}{K}}$ is the attention vector under the k-th interest and $\circ$ represents the Hadamard product.

(2) Hyperedge-to-node (en) propagation. Given the hyperedge features, the partial embedding for the k-th interest can be updated as $h_{v_i}^{k'} = AGG_{en}\left(\beta_{ij}^k f_j^k | e_j \in \varepsilon_{s_{v_i}}\right)$, where β_{ij}^k is the output feature of node v_i and h is the attention coefficient of hyperedge $E_{s_{v_i}}$ on node v_i under the k-th interest. Furthermore, e_j is the set of hyperedges connected to v_i. The embedding of each node v_i is represented as $h_{q_{v_i}}^k = h_{v_i}^k + s^k$, where s^k is the average k-th interest feature of items in session s. Since matching the current session with interest click on the current item is closer, the query-aware attention score is calculated as follows:

$$\beta_{ij}^k = \frac{exp\left(LeakyReLU\left(q_2^{k^T}\left(h_{q_{v_i}}^k \circ f_j^k\right)\right)\right)}{\sum_{e_{j'} \in \varepsilon_{s_{v_i}}} exp\left(LeakyReLU\left(q_2^{k^T}\left(h_{q_{v_i}}^k \circ f_{j'}^k\right)\right)\right)} \tag{7}$$

where $q_2^k \in R^{\frac{d}{K}}$ is the attention vector of the k-th interest.

Additionally, using an interest-aware encoder, specific interest parts $\left(h_{v_i}^{1'}, h_{v_i}^{2'}, ..., h_{v_i}^{K'}\right)$ can be obtained by decoupling local features in any given session.

1) Global Feature Disentanglement within a Session: An auxiliary classification task was designed to decouple the feature vectors of interest on local features within a session to capture semantic information better. The interest-aware part of item $\left\{h_{v_i}^{k'} | v_i \in V_s\right\}$ in session s was used to predict interest classes, represented as:

$$\hat{y}_s^p = softmax\left(MLP\left(\left\{h_{v_i}^{k'} | v_i \in V_s\right\}\right)\right) \tag{8}$$

where $\hat{y}_s^p$ represents the predicted probability of all interests, and MLP is a single-layer multi-layer perceptron. The loss function for interest classification is represented as:

$$\varphi_d^s = -\sum_{k=1}^K 1_{p=k} \log\left(\hat{y}_{sk}^p\right) \tag{9}$$

where $1_{p=k}$ is an indicator function that takes 1 when the predicted interest label is correct.

E. Inter-session Information Extraction Module

The line graph channel is responsible of encoding the line graph, which can integrate the cross-session information and represent the connection relationship of the hyperedges. Before performing the convolutional operations, the method in [8] passes the item embeddings $X^{(0)}$ through filters with Self-Gating Units (SGUs) to obtain line graph-specific item embeddings $X_l^{(0)}$ for the hypergraph. The line graph does not involve other item embeddings; therefore, the items of each session are looked up, and the corresponding item embeddings in $X_l^{(0)}$ are averaged to initialize the session-specific embeddings $h_l^{(0)}$. In addition, the adjacency matrix of $L(G)$ is defined as $A \in R^{M \times M}$, where M is the number of nodes in the line graph. Let $A_{x,y} = W_{x,y}$ and set $\hat{A} = A+1$, where I is an identity matrix and $\hat{D} \in R^{M \times M}$ is a diagonal matrix where $\hat{D}_{x,x} = \sum_{y=1}^m \hat{A}_{x,y}$. Thus, the line graph convolution is defined as follows:

$$h_l^{(l+1)} = \hat{D}^{-1}\hat{A}h^{(l)}Q^{(l)} \tag{10}$$

where $Q^{(l)} \in R^{d \times d}$ is a weight matrix. In each convolutional layer, sessions collect neighbor information from learned h to capture inter-session information. Additionally, by passing $h_l^{(l)}$ through line graph convolutional layers and then averaging session embeddings at each layer, we obtain the final session embedding:

$$h_l = \frac{1}{L+1}\sum_{l=0}^L h_l^{(l)} \tag{11}$$

F. Location Information Fusion Module

The attention mechanism represents relatively important information by assigning different weights. The weights it produces can be continuously updated to select important information based on the circumstances. Referring to the above advantages, attention mechanisms are used respectively to learn item feature representations with position information decoupled whereas soft attention mechanisms are applied to obtain session representations.

$$\chi_{h_{v_i}^k}^i = q_1^T tanh\left(W_1 h_1 + b_i\right) \tag{12}$$

$$\chi_g^i = q_1^T tanh\left(W_1 g_1 + b_i\right) \tag{13}$$

where h_i and g_i are the feature representations of item $v_{s,i}$ based on HGCN and the feature representation of sequence information embedding respectively. $q_1 \in R^d$, $W_1 \in R^{d\times d}$ and $b_1 \in R^d$ are learnable attention vectors, weight matrices and bias vectors.

Use the $softmax$ function to convert importance into weight coefficients:

$$\alpha_{h_{v_i}^k}^i = \frac{exp\left(\chi_{h_{v_i}^k}^i\right)}{exp\left(\chi_{h_{v_i}^k}^i\right) + exp\left(\chi_g^i\right)} = 1 - \alpha_g^i \tag{14}$$

where $\alpha_{h_{v_i}^k}^i$ and α_g^i are the weight coefficients of the item feature representation based on HGCN and the sequence information embedding respectively.

Finally, the session-internal embedding with decoupled position information is represented as:

$$h_{v_{si}}^{g_i} = \alpha_{h_{v_i}^k}^i h_i + \alpha_g^i g_i \tag{15}$$

Given a conversation interest sequence $s = [v_{s,1}, v_{s,2}, \ldots, v_{s,n}]$, the interest embedding vector part is represented as $\left(h_{v_{si}}^{1'}, h_{v_{si}}^{2'}, \ldots, h_{v_{si}}^{K'}\right)$. Using the GCE-GN model [33], reverse position embedding is introduced to study the corresponding interest weight γ_i^k after fusion information. The soft attention mechanism is used to calculate the item weight under each interest as follows:

$$z_i^k = \tanh\left(W_1^k \left[h_{v_{si}}^k \| p_{n-i+1}\right]\right) \tag{16}$$

$$h_{s^*}^k = W_2^k \left[h_{v_{si}}^k \| h_{v_{sn}}^k{}'\right] \tag{17}$$

$$\gamma_i^k = q^{k\mathrm{T}} \sigma\left(\left(W_3^k \left[h_{v_{si}}^k \circ h_{s^*}^k\right]\right) \| \left(W_4^k h_{s^*}^k\right) \| \left(W_5^k z_i^k\right) + b^k\right) \tag{18}$$

where p_{n-i+1} and $h_{v_i}^k{}'$ are reverse position embedding and the k-th interest part of item $v_{s,n}$.W_1^k,$W_2^k \in R^{\frac{d}{K}\times\frac{2d}{K}}$,$W_i^k\big|_{i=3}^5 \in R^{\frac{d}{K}\times\frac{2d}{K}}$ and b^k,$q^k \in R^{\frac{3d}{K}}$ are learnable parameters under the k-th interest. Then, aggregate the learned k-th item's interest part in session s to generate the fused interest's session representation h_s^k as follows:

$$h_s^k = \sum\nolimits_{i=1}^n \gamma_i^k \cdot h_{v_{si}}^k{}' \tag{19}$$

G. Feature Fusion Module

Given a session interest sequence $s = [v_{s,1}, v_{s,2}, \ldots, v_{s,n}]$, according to the session sequence s, both the session-internal information representation, with position information, and the session-inter information representation are aggregated, and the two parts of information are fused by pooling summation.

$$h_s^{g_k l} = SumPooling(h_s^{g_k}, h_s^{(l)}) \tag{20}$$

where $h_s^{g_k l}$ is the final vector representation of an item in a session s after extracting the session-inter information and the session-internal information with position information. Therefore, the entire session can be represented as $s = \left\{h_{s_1}^{g_k l}, h_{s_2}^{g_k l}, \ldots, h_{s_m}^{g_k l}\right\}$.

H. Prediction Module

First, calculate the preference score for each candidate item v_i under each interest of session s, and combine the scores between all fused interests to obtain the final preference score, denoted as:

$$p_{si} = \sum\nolimits_{k=1}^K h_s^{g_k l^{\mathrm{T}}} h_{v_i}^{g_k} \tag{21}$$

For session s, $\hat{p}_s = [\hat{p}_{s_1}, \hat{p}_{s_2}, \ldots, \hat{p}_{s_m}]$ is the score vector containing the predicted scores of all m candidate items. Use the $softmax$ function to calculate the probability that each item becomes the next click item under the session, denoted as:

$$\hat{y}_s = softmax(\hat{p}_s) \tag{22}$$

Then use the cross-entropy loss function for each session s, which is defined as follows:

$$\varphi_r^s = -\sum\nolimits_{i=1}^m y_{si} \log\left(\hat{y}_{si}\right) + \left(1 - \hat{y}_{si}\right) \log\left(1 - \hat{y}_{si}\right) \tag{23}$$

where y_s represents one-hot encoding. Finally, unify the auxiliary interest prediction task with the recommendation task. The total loss of session s is defined as:

$$\varphi^s = \varphi_r^s + \lambda \varphi_r^d \tag{24}$$

where λ represents the weight that balances the two tasks.

The optimization of the model was performed from the perspective of session-level representation, and SSL was applied to the network for hypergraph modeling. Moreover, self-supervised signals were used as auxiliary tasks that were beneficial to perform the recommendation task. Firstly, the self-supervised signals were created. In MHGNN-LI, the hypergraph structural information within and between sessions was encoded by learning two sets of session embeddings. Secondly, a contrastive learning method was used to compare the session embeddings learned from two views. Both channels in MHGNN-LI were regarded as two views at the session level, and the InfoNCE [34] was used as the objective function to learn the task. A noise-contrastive function, with a standard binary cross-entropy loss between positive and negative samples, was also deployed as the learning target for self-supervised learning, which is defined as follows:

$$\eta_s = -\log\sigma\left(f_D\left(h_s^{g_k l}, h_{v_i}^{g_k}\right)\right) - \log\sigma\left(1 - f_D\left(\tilde{h}_s^{g_k l}, h_{v_i}^{g_k}\right)\right) \tag{25}$$

where $\tilde{h_s^{g_k l}}$ represents the negative samples obtained by shuffling the rows and the columns, and $f_D(\cdot) : R^d \times R^d \to R$ indicates a discriminator function that calculates the consistency between both input vectors. Using the discriminator to determine the dot product between two vectors can maximize the mutual information between the session embeddings learned from different views through convolutional operations to pass information between them. Finally, the hypergraph neural network recommendation model is integrated with the SSL into one learning framework for joint optimization. The form of the final learning objective is defined as follows:

$$\mu = \varphi^s + \omega \eta_s \tag{26}$$

where ω is the weight coefficient of the self-supervised learning loss function.

V. Experiments

A. Experimental Settings

1) Datasets: The effectiveness of the proposed model was verified by experiments using the two publicly available Tmall and Diginetica datasets. According to the method proposed in reference [35], all sessions containing only one item were removed from the dataset, and items that appeared less than five times were also removed. In the Tmall dataset, the session data of the previous week was set as the test set whereas the remaining session data was considered as the training set. As for the Diginetica dataset, the test set consisted of the sessions in subsequent weeks. The dataset was expanded and labeled using the sequence segmentation. For each sequence $s = [i_{s,1}, i_{s,2}, \ldots, i_{s,m-1}, i_{s,m}]$, this method can generate multiple tags with corresponding labels $([i_{s,1}], i_{s,2})$, $([i_{s,1}, i_{s,2}], i_{s,3})$, $([i_{s,1}, i_{s,2}, \ldots, i_{s,m-1}], i_{s,m})$. The label of each sequence is the last clicked item in it. The statistical data of the dataset is shown in Table 1.

TABLE I: Dataset Statistics

Dataset	Tmall	Diginetica
# train sessions	351,268	719,470
# test sessions	25,898	60,858
# of items	40,728	43,097
Average lengths	6.69	5.12

2) Evaluation Metrics: The precision P@K and mean reciprocal rank MRR@K are used as evaluation metrics for experimental results.

3) Baseline Methods: To verify the performance of the model, MHGNN-LI was compared with the following representative methods:

- **FPMC [34]:** A sequence modeling algorithm for a personalized recommendation based on the Markov chain model and matrix factorization.
- **GRU4REC [4]:** Uses Gated Recurrent Unit (GRU) to capture time dependence in the user's historical behavior sequence and calculate the interest level of each item.
- **NARM [5]:** An RNN-based model that models sequence behavior to generate recommendations.
- **STAMP [13]:** Uses attention mechanism to capture time dependence in the user's historical behavior sequence and user interest preferences to enhance session-based recommendations.
- **SR-GNN [7]:** Applies Graph Neural Network (GNN) to capture relationships in user's historical behavior sequence.
- **FGNN [18]:** Formulates the next item recommendation within a session as a graph classification problem.
- **DHCN [8]:** A session recommendation algorithm based on a dual-channel hypergraph convolutional neural network that improves recommendation effectiveness through self-supervised learning.
- **HIDE [9]:** A session recommendation algorithm based on hypergraph neural network that improves recommendation effectiveness by introducing an interest separation mechanism.

4) Hyper-parameters Settings: To have a fair comparison, all reference baseline papers reported the best parameter settings in the article. In this experiment, the hidden vector dimension is set to 100, the training batch size is set to 256, and L_2 regularization is set to 10^{-5}. As for the MHGNN-LI model, it adopts a two-layer structure with an initial learning rate of 0.01. All trainable parameters are initialized using a Gaussian distribution with a mean of zero and a standard deviation of 0.1. The training iteration is set to 20.

B. Comparison and Analysis of Model Performance

We have highlighted the best results for each column in Table 2 and drawn the following conclusions through analysis:

(1) Models based on HNN (DHCN, SHARE, and HIDE) demonstrate superior performance compared to models based on RNN, indicating the effectiveness of hypergraph neural network models. However, these models still fall short in terms of performance compared to MHGNN-LI;

(2) The proposed MHGNN-LI significantly outperforms other baseline models on all datasets, particularly exhibiting a remarkable improvement when dealing with the Tmall dataset.

By analyzing the Tmall dataset, it was found that many items co-occur in different sessions in the form of frequent item sets. Compared to SHARE and HIDE, the model has two main advantages:

(1) Using HNNs to capture relationships between multiple session hypergraphs, separating session hypergraphs of interest into multiple specific session hypergraphs for modeling, and then using fully connected items with hyperedges;

(2) Considered mutual information between different sessions.

At the same time, the improvement of MHGNN-LI on MRR is more significant than that regarding the accuracy, which indicates that MHGNN-LI can not only accurately represent items interacted with users, but it can also improve their ranking in top-K recommendation lists. The introduction of the line graphs for contrastive learning brings significant performance improvements for inter-session and intra-session comparisons. More specifically in both datasets with shorter average session lengths, contrastive learning plays an important role. The sparsity of the session data may hinder the benefits of hypergraph interest attention convolutional network modeling whereas maximizing the mutual information between two different session perspectives in MHGNN-LI can solve this problem.

C. Ablation Study

This experiment analyzed the performance of the MHGNN-LI model on the two benchmark datasets presented previously (i.e., Tmall and Diginetica). To determine the contribution of each module in MHGNN-LI, three variants of MHGNN-LI were set up: (1) MHGNN-LI-H, (2) MHGNN-LI-L, and (3) MHGNN-LI-AT, and they were compared with the complete MHGNN-LI. The study showed that the hypergraph attention convolutional network channel, the line graph channel, and the attention mechanism contributed altogether to the final performance. As shown in Figures 4 and 5, the contribution of the hypergraph attention convolutional

TABLE II: Performance of All Models on Two Datasets

Methods	Tmall				Diginetica			
	P@10	M@10	P@20	M@20	P@10	M@10	P@20	M@20
FPMC	13.05	7.11	16.08	7.34	15.43	6.20	26.53	6.95
GRU4REC	9.50	5.75	10.98	5.92	17.93	7.33	29.45	8.33
NARM	19.21	10.39	23.35	10.68	35.44	15.13	49.70	16.17
STAMP	22.64	13.08	26.44	13.35	33.98	14.26	45.64	14.32
SR-GNN	23.49	13.47	27.65	13.76	36.86	15.52	50.73	17.59
FGNN	20.64	10.05	25.27	10.41	37.72	15.95	51.36	18.47
DHCN	26.24	14.63	31.51	15.08	39.87	17.53	53.18	18.44
SHARE	25.14	14.13	30.46	14.57	40.03	17.22	53.21	18.23
HIDE	31.10	16.77	37.12	17.19	40.14	17.38	53.26	18.30
MHGNN-LI	**35.91**	**20.21**	**41.56**	**20.14**	**44.06**	**19.53**	**56.88**	**20.44**

network channel was the largest. When this channel was only used, the network performance was significantly better than the baseline performance. In contrast, using only the line graph channel resulted in a significant drop in the performance for both datasets. However, concerning Diginetica, the MHGNN-LI-L method still achieved baseline levels. Removing the attention mechanism in the hypergraph attention convolutional network channels also led to a significant performance drop on both datasets.

Therefore, these findings indicate that the SBR model needs to consider the temporal factor and the higher-order correlations item. Therefore, the session recommendation models should consider the impact of time factors as well as the high-order item correlations and use hypergraph attention convolutional networks to model high-order correlations.

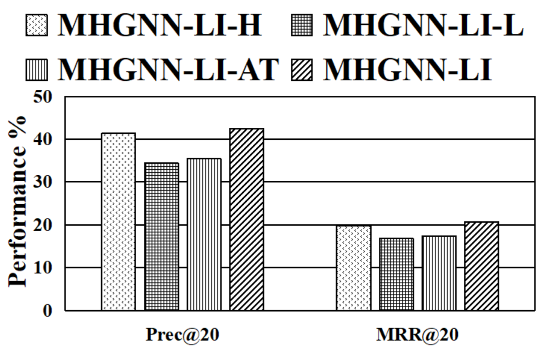

Fig. 4: Performance of two indicators on Tmall dataset

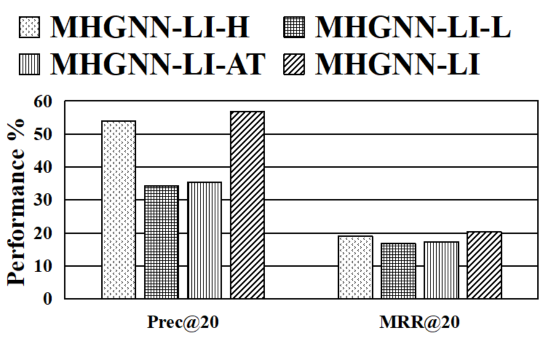

Fig. 5: Performance of two indicators on Diginetica dataset

D. Impact of Different Session Lengths

This experiment studied the stability performance of MHGNN-LI under different session lengths. First, the sessions of Tmall and Diginetica were divided into two groups: short sessions and long sessions, where the former contained sessions with lengths less than or equal to five samples whereas the later consisted of sessions with lengths greater than five. Therefore, the value of five was the cutoff point because it was the most common length among all sessions. The short-term and long-term performance of MHGNN-LI and DHCN, SHARE, and HIDE were compared in terms of P@20. As shown in Figures 6 and 7, MHGNN-LI outperformed all baseline models for both long and short sessions. More specifically, in long sessions, MHGNN-LI's performance was much better than that of the baseline models. Therefore, the experimental results demonstrate the universality of MHGNN-LI in the recommendation.

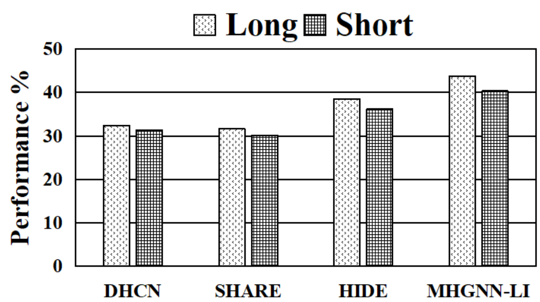

Fig. 6: Comparison of long and short session results of Prec@20 on Tmall dataset

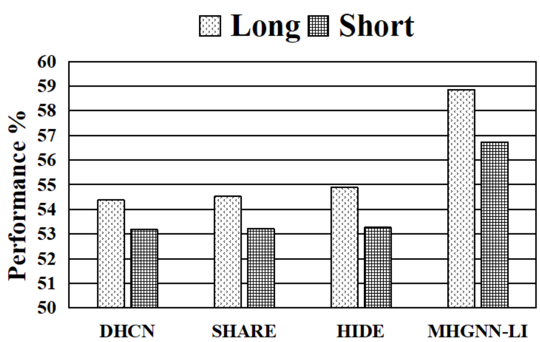

Fig. 7: Comparison of long and short session results of Prec@20 on Diginetica dataset

E. Impact of Model Depth

To study the impact of the depth of the model on the session-based recommendation, the number of layers of the network was limited to $\{1, 2, 3, 4, 5\}$. As shown in Figures 8 and 9, MHGNN-LI was not very sensitive to the number of layers when dealing with the Tmall dataset, and a one-layer setting has the best performance. However, concerning the Diginetica dataset, the performance of a three-layer network is the best. In addition, as the number of layers increases, the performance of MRR@20 decreases. One possible reason is that the performance of items becomes too smooth as the number of layers grows.

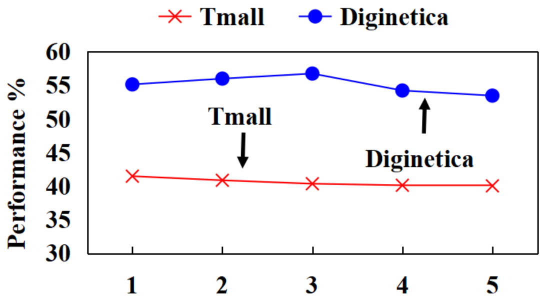

Fig. 8: Impact of model layers on Prec@20

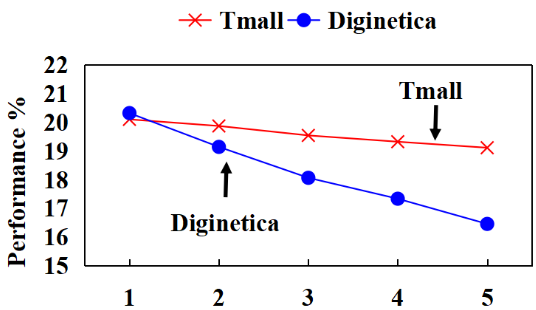

Fig. 9: Impact of model layers on MRR@20

F. Impact of Self-Supervised Learning

A hyperparameter α was introduced to control the impact of self-supervised learning on the model. A set of α values $\{0.01, 0.1, 0.5, 1, 5, 50\}$ was set up to report the performance of MHGNN-LI. As shown in Figures 10 and 11, when the self-supervised tasks were used for optimization, the recommendation task got good benefits. Moreover, α can improve the Prec@20 and MRR@20 values on both datasets. However, as α increases, both indicators gradually decrease regarding the Diginetica dataset whereas very small changes are encountered on Tmall. It is unclear why there is not little in the performance of both indicators on Tmall, and it is hoped that this issue will be addressed in future work. Therefore, there exists a profound interrelation between the item's hit rate and its rating level, and they vary correspondingly with the fluctuations in the value of parameter α.

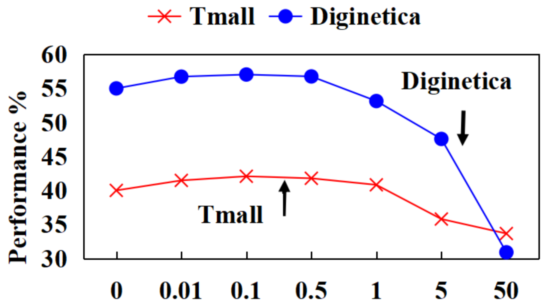

Fig. 10: Impact of self-supervised learning on Prec@20

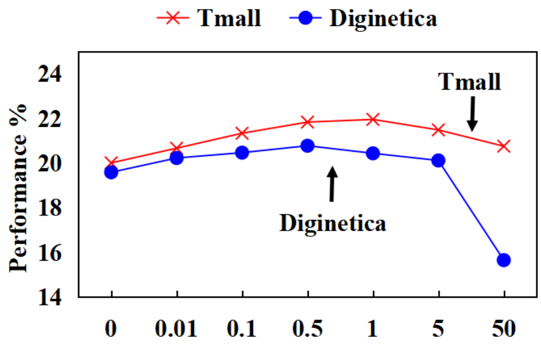

Fig. 11: Impact of self-supervised learning on MRR@20

VI. Conclusion

This paper introduces MHGNN-LI, Multi-hypergraph neural network with fusion of location information for session-based recommendation. The model constructs the line graphs for intra-session and inter-session hypergraphs, enabling the generation of session-level and item-level feature representations with attention mechanisms fused along with location information. Subsequently, sessions are generated using reverse position embedding and soft attention mechanisms, combining the fused interest features with the inter-session feature representations generated by GCN. Finally, a SSL approach is employed to compute the final session representation, facilitating the calculation of the ranking scores for each recommended candidate item.

As for the future research directions, they will involve exploring alternative approaches to jointly optimize user information within and between sessions, aiming to supplementary enhance the recommendation performance.

References

[1] W. Pan and K. Yang, "Enhanced multi-head self-attention graph neural networks for session-based recommendation," *Engineering Letters*, vol. 30, no. 1, pp. 37–44, 2022.

[2] F. Cai, Z. Pan, C. Song, and X. Zhang, "Exploring latent connections in graph neural networks for session-based recommendation," *Information Retrieval Journal*, vol. 25, no. 3, pp. 329–363, 2022.

[3] D. Zhao and L. Feng, "A two-stage machine-learning-based prognostic approach for bearing remaining useful prediction problem," *IAENG International Journal of Computer Science*, vol. 48, no. 4, pp. 1028–1034, 2021.

[4] S. Wang, L. Cao, Y. Wang, Q. Z. Sheng, M. A. Orgun, and D. Lian, "A survey on session-based recommender systems," *ACM Computing Surveys (CSUR)*, vol. 54, no. 7, pp. 1–38, 2021.

[5] J. Yuan, Z. Song, M. Sun, X. Wang, and W. X. Zhao, "Dual sparse attention network for session-based recommendation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 5, 2021, pp. 4635–4643.

[6] S. Wu, F. Sun, W. Zhang, X. Xie, and B. Cui, "Graph neural networks in recommender systems: a survey," *ACM Computing Surveys*, vol. 55, no. 5, pp. 1–37, 2022.
[7] S. Wu, Y. Tang, Y. Zhu, L. Wang, X. Xie, and T. Tan, "Session-based recommendation with graph neural networks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 346–353.
[8] W.-C. Huang, C.-T. Chen, C. Lee, F.-H. Kuo, and S.-H. Huang, "Attentive gated graph sequence neural network-based time-series information fusion for financial trading," *Information Fusion*, vol. 91, pp. 261–276, 2023.
[9] X. Xia, H. Yin, J. Yu, Q. Wang, L. Cui, and X. Zhang, "Self-supervised hypergraph convolutional networks for session-based recommendation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 5, 2021, pp. 4503–4511.
[10] Y. Li, C. Gao, H. Luo, D. Jin, and Y. Li, "Enhancing hypergraph neural networks with intent disentanglement for session-based recommendation," in *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2022, pp. 1997–2002.
[11] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proceedings of the 10th International Conference on World Wide Web*, 2001, pp. 285–295.
[12] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, "Evaluating collaborative filtering recommender systems," *ACM Transactions on Information Systems (TOIS)*, vol. 22, no. 1, pp. 5–53, 2004.
[13] Q. Liu, Y. Zeng, R. Mokhosi, and H. Zhang, "Stamp: short-term attention/memory priority model for session-based recommendation," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 1831–1839.
[14] M. Quadrana, A. Karatzoglou, B. Hidasi, and P. Cremonesi, "Personalizing session-based recommendations with hierarchical recurrent neural networks," in *Proceedings of the Eleventh ACM Conference on Recommender Systems*, 2017, pp. 130–137.
[15] Y. Fan, M. Chen, and X. Pan, "Gcrflda: scoring lncrna-disease associations using graph convolution matrix completion with conditional random field," *Briefings in Bioinformatics*, vol. 23, no. 1, p. bbab361, 2022.
[16] X. Wang, X. He, M. Wang, F. Feng, and T.-S. Chua, "Neural graph collaborative filtering," in *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2019, pp. 165–174.
[17] F. Liu, H. Chen, Z. Cheng, A. Liu, L. Nie, and M. Kankanhalli, "Disentangled multimodal representation learning for recommendation," *IEEE Transactions on Multimedia*, pp. 1–11, 2022.
[18] R. Qiu, J. Li, Z. Huang, and H. Yin, "Rethinking the item order in session-based recommendation with graph neural networks," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2019, pp. 579–588.
[19] F. Yu, Y. Zhu, Q. Liu, S. Wu, L. Wang, and T. Tan, "Tagnn: Target attentive graph neural networks for session-based recommendation," in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2020, pp. 1921–1924.
[20] Y. Feng, H. You, Z. Zhang, R. Ji, and Y. Gao, "Hypergraph neural networks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 3558–3565.
[21] N. Yadati, M. Nimishakavi, P. Yadav, V. Nitin, A. Louis, and P. Talukdar, "Hypergcn: A new method for training graph convolutional networks on hypergraphs," in *Advances in Neural Information Processing Systems*, vol. 32, 2019.
[22] J. Wang, Y. Zhang, Y. Wei, Y. Hu, X. Piao, and B. Yin, "Metro passenger flow prediction via dynamic hypergraph convolution networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 12, pp. 7891–7903, 2021.
[23] S. Bandyopadhyay, K. Das, and M. N. Murty, "Line hypergraph convolution network: Applying graph convolution for hypergraphs," *CoRR*, vol. abs/2002.03392, 2020.
[24] J. Wang, K. Ding, L. Hong, H. Liu, and J. Caverlee, "Next-item recommendation with sequential hypergraphs," in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2020, pp. 1101–1110.
[25] J. Wang, K. Ding, Z. Zhu, and J. Caverlee, "Session-based recommendation with hypergraph attention networks," in *Proceedings of the 2021 SIAM International Conference on Data Mining (SDM)*, 2021, pp. 82–90.
[26] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proceedings of the 37th International Conference on Machine Learning*, vol. 119, 2020, pp. 1597–1607.
[27] T. Yao, X. Yi, D. Z. Cheng, F. Yu, T. Chen, A. Menon, L. Hong, E. H. Chi, S. Tjoa, J. J. Kang, and E. Ettinger, "Self-supervised learning for large-scale item recommendations," in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 2021, p. 4321–4330.
[28] J. Wu, X. Wang, F. Feng, X. He, L. Chen, J. Lian, and X. Xie, "Self-supervised graph learning for recommendation," in *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2021, pp. 726–735.
[29] X. Xia, H. Yin, J. Yu, Y. Shao, and L. Cui, "Self-supervised graph co-training for session-based recommendation," in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 2021, pp. 2180–2190.
[30] Q. Gao, W. Wang, K. Zhang, X. Yang, C. Miao, and T. Li, "Self-supervised representation learning for trip recommendation," *Knowledge-Based Systems*, vol. 247, p. 108791, 2022.
[31] G. Tang, X. Zhu, J. Guo, and S. Dietze, "Time enhanced graph neural networks for session-based recommendation," *Knowledge-Based Systems*, vol. 251, p. 109204, 2022.
[32] F. L. Abba Suganda Girsang and D. Tanjung, "Stock price prediction using lstm and search economics optimization," *IAENG International Journal of Computer Science*, vol. 47, no. 4, pp. 758–764, 2020.
[33] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *ArXiv Preprint ArXiv:1807.03748*, 2018.
[34] S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme, "Factorizing personalized markov chains for next-basket recommendation," in *Proceedings of the 19th International Conference on World Wide Web*, 2010, pp. 811–820.
[35] Z. Wang, W. Wei, G. Cong, X.-L. Li, X.-L. Mao, and M. Qiu, "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*, 2020, pp. 169–178.