Journal of Network Intelligence
Taiwan Ubiquitous Information

ISSN 2414-8105 (Online)
Volume **10**, Number **1**, February **2025**

Multi Meta Information Fusion Graph Neural Networks for Session-based Recommendation

Yan Wen*

College of Computer Science and Engineering
Shandong University of Science and Technology, Qingdao 266590, China
wenyanxxxy@163.com

Hui-Zheng Sun

College of Computer Science and Engineering
Shandong University of Science and Technology, Qingdao 266590, China
1830176257@qq.com

Liang Liu

Shandong Port Group Co., Ltd, Qingdao 266000, China
392991408@qq.com

Ya-Kun Zhou

Geography and Environment Management-Faculty of Environment
Universoty of Waterloo, Waterloo N2L3G1, Canada
yakunzhou950518@gmail.com

Wei Bian

College of Computer Science and Engineering
Shandong University of Science and Technology, Qingdao 266590, China
1533884130@qq.com

Ming-Hai Yan

College of Computer Science and Engineering
Shandong University of Science and Technology, Qingdao 266590, China
1874789353@qq.com

*Corresponding author: Yan Wen

Received March 14, 2024, revised May 31, 2024, accepted August 25, 2024.

Abstract. *Session-based recommendation aims to predict the anonymous user's next click action based on his/her latest click sequence. Most state-of-the-art approaches use graph neural networks to model sessions as graphs to capture dynamic transitions between items within a session. Inspired by recent progress on hypergraph-based neural networks, we think it is a good candidate to model the commonly related items within a session, and it can provide a sound complement to the normal graph models. In addition, cross session information may also be helpful. But the excessive noise introduced by other sessions is a big challenge and is still not be well resolved by existing works. To this end, we propose a new approach, called Multi Meta Information Fusion Graph Neural Networks (MM-GNN) to infer user preferences for the current session in a more comprehensive way for the anonymous sessions. We fuse information from normal graph and hypergraph to fully exploit their capabilities of capturing sequential behavior and group intention within a session. In order to better use cross session information, we propose a global-level subsequent item injection module, which injects information of the most possible subsequent item of the last item in the current session in the global scope. Besides, to learn richer information on the hypergraph structure, we model the whole session as a hyperedge to complement sliding windows-based hyperedges when creating the hypergraph. We have evaluated the method on several benchmark datasets and the results show that our method outperforms state-of-the-art methods.*
Keywords: Session-based recommendation; Sequential behavior; Group intention

1. **Introduction.** In many cases, the rapid growth of information has brought us more opportunities, but at the same time it has caused us to suffer from information overload. Recommender systems play a crucial role in inferring user preferences or needs from their historical actions such as browsing and clicking. However, in many real-world scenarios, user profiles and long-term historical data are not available due to privacy policies or anonymous interactions. To solve this problem, session-based recommendation (SBR) is proposed to predict the next potential interacted items based on short-term and dynamic user behaviors in anonymous sessions.

Over the past few years, many methods have been proposed for SBR. Earlier research work modeled target sessions as decision Markov chains [1–4] to make the prediction from users' recent behavior. With the success of deep learning, many researches have focused on using deep neural networks to mine users' preferences. Recurrent neural networks (RNN), gated recurrent units (GRU), and long short-term memory (LSTM) are widely used to capture sequential correlations among items in a session.

In recent years, the rise of graph neural networks has brought new ideas for session-based recommendation, they can model target sessions as graphs and learn complex transition patterns between items, such as forward, backward, bidirectional and self-transitional relationships. In SR-GNN [5], the items within a session are modeled as a graph, and the complex transition patterns between items are learnt through graph structures. To learn richer item representations, many researchers have constructed GNNs using cross-session information, such as GCE-GNN [6], DAT-MDI [7] and GAG [8], which capture transitional relationships between items within a single session or across sessions. Meanwhile, these works mostly uses a normal graph as the main structure which takes items as nodes, and learns transition patterns between nodes in the graph. However, if we observe the session in a higher perspective, we can notice that in a session, there exists several group intentions [9], each of which represents a single objective of several consecutive interactions between the user and the system, and there also exists the phenomenon of intention transition within a session. Taking session A in Figure 1 as an example, if we use normal graph and only considers the transitions between each item, then the next most likely item to be visited after the last RAM would be CPU because there already exists a transition from RAM to CPU. But if we observe the session in a global way, we can discover that the first two items are mainly targeted to find a cell phone, while the last items are aimed to assemble a computer. So if we consider the last five items as a whole, then the next item to be visited might be a mainboard or a screen as they are parts of a computer. To obtain a better session representation, we can resort to hypergraph to be complementary to the normal graph structure. In hypergraph, the hyperedges are used to connect multiple nodes to represent that these nodes are commonly related, so as to cover richer association information. SHARE [10] was the first model to introduce sliding window into hypergraph to resolve the session-based recommendation problem, which capture the group intentions within a session, and achieved good result. Although hypergraph is very effective in capturing group intentions, there are still some cases where group intentions are not obvious, like the example of Session B in Figure 1. Obviously it is more effective to use the normal graph to learn the item transitions. Both of above situation are widely existed in real-world and should be paid attention to.

Figure 1. Two sessions

To overcome the above limitations and make full use of the item-level and group-level features of sessions, we propose a graph neural network-based recommendation model called MM-GNN which combines the hypergraph and normal graph, and can better learn both the group intention transitions and item transitions within a session. First, we construct a hypergraph-based module which uses sliding windows to capture the group intentions of a session. Each window tries to capture the group intention within a certain region, and their combination may better learn the group intentions and their transitions in the whole session. Besides, we propose to model the whole session as a hyperedge, which can further enrich the expressiveness of the model. To compensate the loss of sequential information within a session, we also incorporate a normal GNN module which is constructed from the transitions between items and can better learn their first and higher order transitive relationships. Finally, in order to better use cross-session information, we build a global-level subsequent item injection module by computing the most likely next items for the last item in the current session from a global perspective, and then the candidate items are injected into the current session representation by a fusing function. This strategy not only enriches our model with global cross-session information, but also emphasize the importance of the last item and activate the position encoding in a session in a rather simple and effective way. All above modules are fused to get the final recommendation results. Experimental results show that our proposed method outperforms existing SBR-based methods on three widely used session recommendation datasets.

Contribution. In this paper, we propose a new scheme for session-based recommendation. The contribution of our paper can be summed up as the following four points:

(1) We effectively integrate hypergraph and normal graph, so that the advantages of the two graph structures can complement each other.

(2) We propose to use both sliding windows and the whole session to generate the hyperedge of a hypergraph, so that more precise group intentions and their transitions can be learnt.

(3) We build a global-level subsequent item injection module, which can effectively learn the subsequent item information of the last item in the current session from a global perspective. Through the inter session collaboration mechanism, we providing richer information for the final recommendation.

(4) We conducted extensive experiments on three real datasets and results show that MM-GNN outperforms multiple baselines.

The rest of this paper consists of the following sections: Section 2 reviews the related work. The proposed MM-GNN model is presented in Section 3, and experiments are performed in Section 4. Finally, the paper is concluded in Section 5.

2. **Realted work.** In recent years, Session-Based Recommendation (SBR) has received increasing attention. Session-based recommendation aims to predict the user's next action based only on his/her recent anonymous interactions which are regarded as a session. In this section, we discuss the recently proposed session-based recommendation methods in details, which can be divided into two main categories, i.e., RNN-based or GNN-based methods.

RNN-based methods: Classical matrix decomposition methods (e.g., probabilistic matrix decomposition [11] and matrix decomposition [12]) perform poorly due to limited representation capabilities. Therefore, RNN-based approaches [13–16] received much attention afterwards. The RNN-based approaches employ a loop and gated cell structure which makes it easy to capture user preference shifts when generating session representations. GRU4REC [13] introduces RNN for the first time in session-based recommender systems, using deep recurrent neural networks with gated recurrent units to model session data. Inspired by the great success of Transformer [17, 18], SASRec [19] proposes a self-attention-based recommendation method that tends to model long-term dependencies on dense data and focuses on recent actions on sparse data. SINE [16] proposes a novel sparse interest network that learns adaptively from a user's current interests by constructing an overall conceptual prototype matrix. Although RNN-based recommendation methods have gained some success, they demand huge data support and it is difficult for RNN models to learn the precise preference of users in the case of sparse session sequences.

GNN-based methods: Graphs have more powerful expressiveness in modeling complex structures than normal sequences, thus recent works mainly use GNN (Graph Neural Network) to learn item representations of SBR by constructing a session graph. Wu et al. [5] is the first to propose a recommendation model based on graph neural networks (SR-GNN) to extract short-term dynamic preferences of users in a session and predict their behaviors by establishing the transitive relationships between items through GNN. GCE-GNN [6] achieves more accurate recommendations by constructing a global graph and learning global co-occurrence information to complement the representation learning of item transitions in the current session. DHCN [20] realizes the sparseness of the data in the current session and learns inter-session information by taking each session as a node and uses comparative learning for item nodes and session nodes. HG-GNN [21] introduces user nodes to construct heterogeneous global graph neural network to better learn user preference information. MGS [22] uses attribute information on the session graph and utilizes an attribute-aware module to construct a mirror graph, which further guides the training process by propagating information between the session graph and the mirror graph. STAR [23] introduces time interval to provide more precise prediction for the next item. CORE [24] addresses the problem that session embeddings are usually not in the same representation space as item embeddings, and designs a representation-consistent encoder to ensure the consistency of the two representation spaces. Noting that previous models have ignored the fact that user preference is usually driven by some factor, Disen-GNN [25] introduces Dissociative Representation Learning (DRL) to learn independent factor-level embeddings and to explore more fine-grained session information.

Through learning from the above fields and being inspired by other fields [26–28], we have made new progress in our work. Firstly, we combine normal graph and hypergraph, there are few studies combining the two, and this is the research gap we try to fill in this work. Second, we model the whole session as a hyperedge to complement sliding windows-based hyperedges when creating the hypergraph. Third, we introduce subsequent sessional information from a global perspective to gain better recommendation results.

3. Methods.

3.1. Problem statement.

Session-based recommendations are designed to predict a user's next item of interest based on his/her recently visited items. Let $V = \{v_1, v_2, ..., v_m\}$ denotes the set of unique items in all sessions, where $m = \|V\|$ is the number of items. A session $s = \{v_1, ...v_i, ..., v_l\}$ is defined as a sequence of items, where denotes the ith item visited by the user in session and is the length of session . The session-based recommendation model outputs the probability for each item to be selected as the next item i.e., $\hat{y} = \{\hat{y}_1, \hat{y}_2, ..., \hat{y}_m\}$, and top k items will be recommended to the user.

3.2. Network architecture.

To obtain more comprehensive and reliable session representations, we propose a new model called Multi Meta Information Fusion Graph Neural Networks (MM-GNN) for session-based recommendation. MM-GNN models the session in two forms to obtain better representations for the items and sessions, one is in the form of the normal graph, the other is in the form of the hypergraph which is generated by applying sliding windows on a session, and it also incorporates information of possible subsequent items from the global perspective to better utilize cross-session information. Figure 2 presents the architecture of MM-GNN, which consists of five main components. 1) Normal Graph-based Learning Module. It uses a GNN model on the session graph to learn item embeddings in the current session. 2) Hypergraph-based Learning Module. It uses a GNN model on the hypergraph to learn item and hyperedge representations bi-directionally, i.e., from nodes to hyperedges and from hyperedges to nodes. 3) Subsequent Item Injection Module. It learns the sequential pattern from all sessions for the last item of the current session. 4) Fusion module. This module fuses the information learnt in the previous modules. 5) Prediction module. It generates the predicted probability of the recommended candidate items. Next, we will introduce these five components in detail.

3.3. Normal graph-based learning module.

Let G_s denotes a session graph for session s, the edges are the transitions between the items in session s. The vector $h_{v_i} \in R^d$ represents the initial d-dimensional embedding for item v_i, where $v_i \in V$.

Since the neighbors of v_i in the session graph have different importance and similarities to v_i, we use the attention mechanism to learn different weights for its neighbors. The attention coefficients can be obtained by element-wise production and nonlinear transformation:

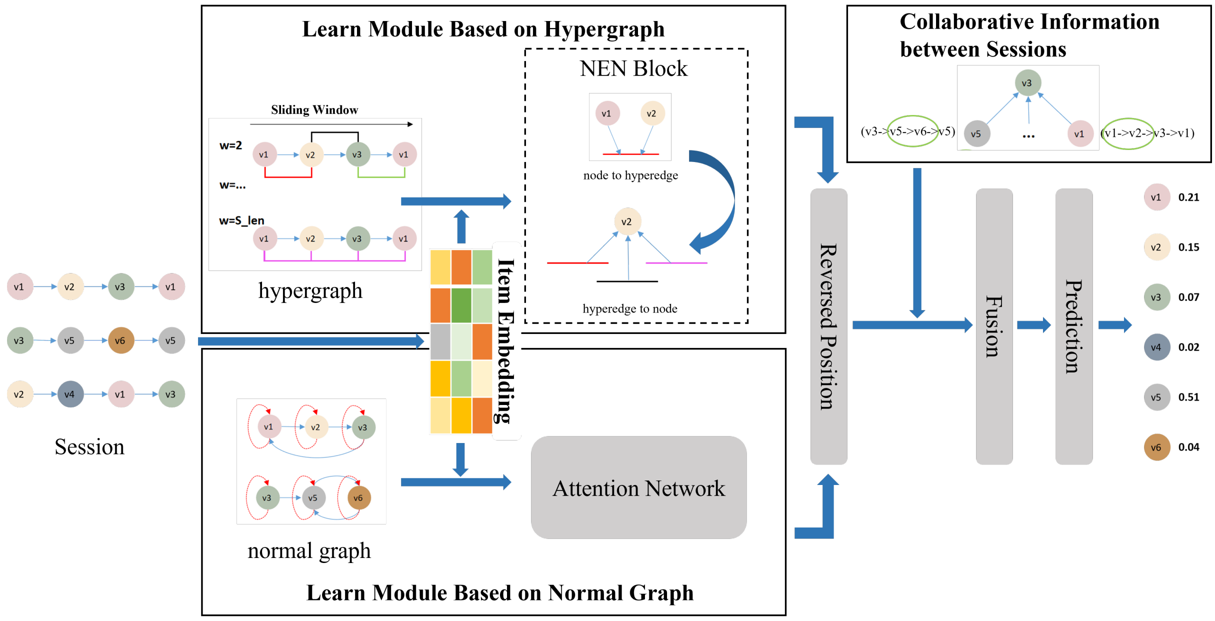

FIGURE 2. The pipeline of the proposed MM-GNN model

$$f_{ij} = LeakyReLU\left(a_b^T\left(h_{v_i} \bullet h_{v_j}\right)\right) \tag{1}$$

Where we use LeakyReLU as the activation function, f_{ij} denotes the importance of the node v_j to the node v_i, $v_j \in N_{v_i}^s$, $N_{v_i}^s$ is the set of the first-order neighbors of v_i. $a_b \in R^d$ is the weight vector, can be either of (a_{in}, a_{out}, a_{in-out}, a_{self}) according to different transition directions. a_{in}, a_{out}, a_{in-out} and a_{self} are trainable weight vector. For the edge (v_i, v_j), a_{in} corresponds to the transition(edge) from v_i to v_j only, a_{out} corresponds to transition(edge) from v_j to v_i only, a_{in-out} corresponds to the transition between v_i and v_j in both direction, a_{self} corresponds to the self-transition edge.

In order to make the coefficients comparable between different nodes, we normalize the attention scores by the softmax function:

$$\alpha_{ij} = \frac{\exp\left(f_{ij}\right)}{\sum_{v_k \in N_{v_i}^s} \exp\left(f_{ik}\right)} \tag{2}$$

The output representations of each node can be obtained by a linear combination of the representations of its neighbor nodes, through the following formula:

$$h_{v_i}^n = \sum_{v_j \in N_{v_i}^s} \alpha_{ij} h_{v_j} \tag{3}$$

3.4. **Hypergraph-based learning module.** Hypergraph is a special type of graph structure that extends the traditional understanding of graphs. In a normal graph, an edge can usually only connect two nodes, which is limited in describing many complex systems. The emergence of hypergraphs is precisely to make up for this deficiency. In a hypergraph, an edge can connect multiple nodes without being limited by the number of nodes, making it more flexible and accurate in describing complex relationships and dependency structures in the real world.

In the hypergraph-based learning module, we use the sliding windows to capture transferring group intention of a session. We set different window size to capture different granularity of interest. For example, for a session $\{v_1, v_2, v_3, v_4\}$, we can set the windows size to be 2 or 3, which can capture the group intention of 2 or 3 consecutive items. For example, if the window size is set to 2, then there can be 3 hyperedges which are $e_1 = \{v_1, v_2\}$, $e_2 = \{v_2, v_3\}$, $e_3 = \{v_3, v_4\}$; if the window size is set to 3, then there can be 2 hyperedges which are $e_1 = \{v_1, v_2, v_3\}$, $e_2 = \{v_2, v_3, v_4\}$. Besides, we have added the whole session as a hyperedge to in the hypergraph, i.e., the length of the hyperedge is entirely determined by the length of the session.

We learn nodes and hyperedges representations in the hypergraph through a number of NEN blocks which contains two propagation steps with attention mechanism, i.e., node to hyperedge propagation and hyperedge to node propagation.

(1) Node to hyperedge propagation. Among the nodes connected by one hyperedge, some play more important roles than others as they can reveal more informative preferences contained in this window, thus they are key factors for the hyperedge representation learning and should be paid more attention to. So when performing aggregation, we give different weights to these nodes by using the attention mechanism to obtain the hyperedge feature t_j. We assume that the nodes connected by the hyperedge e_j can form a cluster, which has a virtual center and can be calculated as the average value of the nodes representations of the hyperedge, i.e., $h_{e_j m} = \text{Mean}\,(h_{v_i} \mid v_i \in e_j)$. Nodes near the cluster center are more likely to reflect core preferences, so we consider $h_{e_j m}$ as a factor that determines whether the current node v_i is more important for hyperedge e_j and use it to calculate the node weights. Then we obtain the feature t_j of hyperedge e_j by the following formula:

$$\alpha_{ji} = \frac{\exp\left(LeakyReLU\left({W_1}^T\left(h_{e_j m} \bullet h_{v_i}\right)\right)\right)}{\sum_{v_x \in e_j} \exp\left(LeakyReLU\left({W_1}^T\left(h_{e_j m} \bullet h_{v_x}\right)\right)\right)} \tag{4}$$

$$t_j = \sum_{v_x \in e_j} \alpha_{jx} h_{v_x} \tag{5}$$

Where $W_1 \in R^{d \times 1}$ is a trainable parameter, and $\bullet$ denotes the Hadamard product, α_{ji} denotes the attention weight of hyperedge e_j on node v_i.

(2) Hyperedge to node. When updating the embedding of a node, we need to aggregate the information from all the hyperedges connected to that node. Similar to the node to hyperedge, we also use the attention mechanism to aggregate the hyperedges to obtain a node representation $h_{v_i}^{\#}$. And we regard the average of all items contained in the current session as h_{v_m}, which determines whether the hyperedge e_j is more important for current node v_i, i.e., $h_{v_m} = \text{Mean}\left(h_{v_p} \mid v_p \in s\right)$. Then we obtain the node feature $h_{v_i}^{\#}$ by the following formula:

$$\beta_{ij} = \frac{\exp\left(LeakyReLU\left({W_2}^T(h_{v_m} \bullet t_j)\right)\right)}{\sum_{t_x \in E_{v_i}} \exp(LeakyReLU({W_2}^T(h_{v_m} \bullet t_x)))} \tag{6}$$

$$h_{v_i}^{\#} = \sum_{t_x \in E_{v_i}} \beta_{ix} t_x \tag{7}$$

Where $W_2 \in R^d$ is a trainable parameter. E_{v_i} is the set of all hyperedges to which v_i belongs in the current session, and β_{ij} denotes the attention score of node v_i to hyper-edge e_j.

Based on the above two-way propagation process, i.e., node to hyperedge propagation and hyperedge to node propagation, we can have the hyperedge representation and node representation as $t_j, h_{v_i}^{\#} = NEN\left(h_{v_i}, t_j\right)$.

Here, we let the hyperedge obtained from the whole session be e_d, and its hyperedge feature t_d is obtained through the above process NEN. t_d is used in the computation of the final item representation, because it reflects the intention of the whole session. The item representation is obtained by:

$$h_{v_i}^{h} = \sigma\left(t_d \bullet h_{v_i}^{\#}\right) \bullet h_{v_i}^{\#} + h_{v_i}^{\#} \tag{8}$$

Where σ is the sigmoid activation function.

3.5. **Subsequent item injection module.** The subsequent item injection module aims to capture global-level sequential patterns of an item. For the current session, the last item to some extent reflects the user's recent interests and is a clue to the user's current intention, which has great relevance to the selection of the next item. The subsequent operations on the last item in the current session in other sessions reflect more general item relevance, so this information can be used to learn important influencing factors for recommending the next item.

Specifically, the global graph is constructed based on the set of items in the t-hop range of all sessions in the training set. For example, assume there are two sessions $s_1 = \{v_1, v_2, v_3, v_4, v_5\}$ and $s_2 = \{v_6, v_7, v_8, v_2, v_9\}$ in the training set, when the hop range $t = 2$, the subsequent items of v_2 is v_3, v_4, v_9. Since there are tens of thousands of sessions in the training set, there will be many subsequent items for each item, and we select the top r as the final set based on their co-occurrences. During the testing phase, we do not dynamically update the topology of the global graph for efficiency reasons.

For the current session s, it can be represented by the subsequent items of the last item v_l. We define $v_x \in N_{\text{subsq}}^{l}$ as a global-level possible subsequent item of the last item v_l in the current session s. We select r items N_{rsubsq}^{l} from all subsequent items based the number of co-occurrence between last item v_l and its subsequent items for aggregation.

We aggregate the subsequent information from two aspects. One is to learn global-level item representation by attention mechanism, and the other is the simple and effective averaging strategy.

(1) Attention based. First, we compute the attention k_{lj} of last item v_l on subsequent item v_j. The formula is as follows:

$$k_{lj} = \frac{\exp\left(W_4 LeakyReLU\left(W_3{}^{T}\left[\left(h_{v_l} \bullet h_{v_j}\right) \| u_{lj}\right]\right)\right)}{\sum_{v_x \in N_{rsubsq}^{l}} \exp\left(W_4 LeakyReLU\left(W_3{}^{T}\left[\left(h_{v_l} \bullet h_{v_x}\right) \| u_{lx}\right]\right)\right)} \tag{9}$$

Where $||$ denotes concatenation operation, $u_{lj} \in R^1$ is the number of co-occurrence between last item v_l and subsequent item v_j, $W_3 \in R^{(d+1)\times d}$, $W_4 \in R^{1\times d}$ are trainable parameters.Then, we aggregate r subsequent items to obtain session representation:

$$S_a^l = \sum_{h_{v_x} \in N_{rsubsq}^{l}} k_{lx} h_{v_x} \tag{10}$$

(2) Averaging based. The average subsequent information is calculated as:

$$S_{\text{avg}}^{l} = \text{Mean}\left(h_{v_x} \mid v_x \in N_{rsubsq}^{l}\right) \tag{11}$$

Ultimately, our global-level subsequent information is obtained by summing the two outputs of attention and average:

$$S_{\text{subsq}} = S_a^l + S_{\text{avg}}^l \tag{12}$$

3.6. **Fusion module.** We fuse item representation from normal graph and hypergraph to fully exploit sequential behavior and group intention capture capabilities. The formula is as follow:

$$h_{v_i}^f = tanh\left(W_5\left[h_{v_i}^n \| h_{v_i}^h\right]\right) \tag{13}$$

Where $h_{v_i}^f$ is fusion item embedding, $h_{v_i}^n$ is item embedding from normal graph, $h_{v_i}^h$ is item embedding from hypergraph, $W_5 \in R^{d \times 2d}$ is a trainable parameter.

The items in a session are not equally important. Intuitively, recently selected items can better reflect the user's next preference. Therefore, we employ a positional mechanism that combines the item embedding together with their positions into the session embedding. We use a learnable position embedding matrix $P = [p_1, p_2, \ldots, p_i, \ldots, p_l]$, Where $p_i \in R^d$ is a position vector for a particular position i. The item embedding with position encoded is obtained by concatenation node representations $h_{v_i}^f$ and position embedding p_{l-i+1}.

$$z_i = tanh\left(W_6\left[h_{v_i}^f || p_{l-i+1}\right]\right) \tag{14}$$

Where z_i is a item embedding with position information, $W_6 \in R^{d \times 2d}$ is a trainable parameter. Then the average of the current session is obtained by the averaging operation. The formula is as follow:

$$s^* = \frac{1}{l} \sum_{i=1}^{l} h_{v_i}^f \tag{15}$$

Next we integrate item embedding with position information and the average of the current session through the soft attention mechanism as weight β_i for the current item v_i.

$$\beta_i = W_7^T \sigma\left(W_8 z_i + W_9 s^* + b\right) \tag{16}$$

Where W_8, $W_9 \in R^{d \times d}$ and W_7, $b \in R^{d \times 1}$ are trainable parameters. Thus, the session representation $S_{nh} \in R^d$ from normal graph and hypergraph is given by:

$$S_{nh} = \sum_{i=1}^{l} \beta_i h_{v_i}^f \tag{17}$$

Finally, we fused session representations from normal graph, hypergraph and subsequent items to obtain the final session representation S.

$$S = S_{nh} + S_{\text{subsq}} \tag{18}$$

3.7. **Prediction module.** Based on the obtained session representation S, the final recommendation probability of each candidate item depends on the similarity between their initial embedding and the current session representation, and we first use the dot product and then apply the softmax function to obtain the output.

$$p_i = Softmax\left(S^T h_{v_i}\right) \tag{19}$$

Where p_i denotes the probability of item v_i to be selected as the next clicked item in the current session, h_{v_i} is the initial embedding vector of v_i. The loss function we define as:

$$L = -\sum_{i=1}^{m} y_i log\left(p_i\right) \tag{20}$$

Where y_i is the one-hot vector corresponding to the next interaction of the real user.

4. **Experiments.** By answering the following four key research questions, we conducted a large number of experiments to evaluate the accuracy of the proposed MM-GNN model.

- RQ1: Does MM-GNN outperform the state-of-the-art SBR baseline in real-world datasets?
- RQ2: Are the modules we added valid?
- RQ3: How does MM-GNN perform in different aggregation operations?
- RQ4: Does stacking multiple NEN blocks have an effect on the results?

4.1. **Dataset and preprocessing.** In this paper, we validate the effectiveness of the proposed MM-GNN method on three real datasets, namely Diginetica, Tmall and Nowplaying. Diginetica dataset is from CIKM Cup 2016, consisting of typical transaction data. Tmall dataset comes from IJCAI-15 competition, which contains anonymous user's shopping logs on Tmall online shopping platform. Nowplaying dataset comes from [29], which describes the music listening behavior of users. Following the literature [5, 23], we preprocessed the three datasets. Specifically, sessions of length 1 and items that occur less than 5 times are filtered out in all three datasets. The latest data such as last week's data is set as the test set and more earlier data are used as the training set. Furthermore, similar to the literature [5, 14, 15], we enhanced the dataset using the technique of data augmentation.Specifically, for a session sequence $s = \{v_1, v_2, v_3, \ldots, v_l\}$, we augment s by generating a set of sequences and their corresponding labels $(\{v_1\}, v_2), \ldots, (\{v_1, \ldots, v_{i-1}\}, v_i), \ldots, (\{v_1, v_2, \ldots, v_{l-1}\}, v_l)$, where $\{v_1, v_2, \ldots, v_{i-1}\}$ is a generated sequence and the label v_i is its next visited item. We trained and tested on three publicly available datasets. The statistics of the pre-processed datasets are shown in Table 1.

TABLE 1. Statistical results of datasets

Dataset	Tmall	Nowplaying	Diginetica
all the clicks	818,479	1,367,963	982,961
train sessions	351,268	825,304	719,470
test sessions	25,898	89,824	60,858
all the items	40,728	60,417	43,097
Average length	6.69	7.42	5.12

4.2. **Evaluation metrics.** We used two ranking-based metrics: $P@\ K$ and $MRR@\ K$.

(1) $P@K$ (Precision) is a widely used metric that indicates the probability that a real user clicks on an item in the top K items in the recommendation list in the test set. K is assigned as $K = 20$ in this paper. The formula for calculating $P@\ K$ is as follows:

$$P@K = \frac{n_{\text{hit}}}{T} \tag{21}$$

Where T is the number of sessions in the test set and n_{hit} is the number of sessions that contain real user click items.

(2) $MRR@K$(Mean Reciprocal Rank) is the mean reciprocal rank of the user's real click item position in the recommendation list. If the top K items recommended do not include real clicked items, the value of MRR is 0 . The formula is as follows:

$$MRR@K = \frac{1}{T}\sum_{t=1}^{T}\frac{1}{Rank\left(v_i\right)} \tag{22}$$

Where $Rank\left(v_i\right)$ represents the ranking of the target item among the top K recommended items.

4.3. **Baseline algorithms.** In order to evaluate the performance of the MM-GNN method, we've compared it with the following representative works:

- FPMC [1]: It is a sequential model based on Markov chains.
- NARM [14]: It is the state-of-the-art RNN-based model that uses an attention mechanism to capture the user's primary purpose and combines it with sequential behavior to generate recommendations.
- STAMP [15]: It employs self-attention mechanism to enhance the performance of session-based recommendations.
- SR-GNN [5]: A gated graph convolutional layer is applied to obtain item embedding, and a soft attention mechanism is also used to compute session embeddings.
- GCE-GNN [6]: Two types of session graphs are constructed to capture different levels of local and global information.
- SHARE [10]: It uses a sliding window to build a hypergraph of a single session to capture group intentions.
- DHCN [20]: It constructs two types of hypergraphs to learn inter and intra session information, and uses self-supervised learning to enhance session-based recommendations.
- Disen-GNN [25]: It represents each item with independent factors and learns item embeddings from the factor level.

4.4. **Parameter setup.** Following previous work [30], we set the embedding dimension to be 100 and a batch size of 100. We use an Adam optimizer with an initial learning rate of 0.001, which decays to 0.1 times the original rate after every three rounds. All parameters are initialized using a Gaussian distribution with a mean of 0 and a standard deviation of 0.1. In all experiments, the above parameter settings are kept consistent across all models for a fair comparison.

4.5. **Overall comparison(RQ1).** Table 2 shows the experimental results for the eight baselines and our proposed model on three real-world datasets, where the best results for each column are highlighted in bold. It can be observed that MM-GNN achieves the best performance on both metrics for all three datasets, which verifies the effectiveness of our proposed method.

TABLE 2. Performance comparisons between our model and the competitors over three datasets

DATASET	TMALL		NOWPLAYING		DIGINETICA	
METHODS	P@20	MRR@20	P@20	MRR@20	P@20	MRR@20
FPMC	16.06	7.32	7.36	2.82	22.14	6.66
NARM	23.30	10.70	18.59	6.93	48.32	16.00
STAMP	26.47	13.36	17.66	6.88	46.42	15.13
SR-GNN	28.04	13.60	19.38	7.78	52.49	18.39
GCE-GNN	32.42	15.42	22.37	8.06	54.06	19.04
SHARE	29.14	14.42	19.72	8.14	51.41	17.24
DHCN	31.42	15.05	23.08	8.18	53.66	18.51
DISEN-GNN	31.56	15.31	22.47	8.22	53.49	18.99
MM-GNN	**35.19**	**15.90**	**23.28**	**8.50**	**55.13**	**19.45**

In traditional methods, FPMC uses the first-order Markov chains and matrix decomposition, the results shows limited effectiveness on all three datasets, indicating that the first-order assumption relies mainly on the independence of consecutive items is unrealistic.

In deep learning-based works, neural network-based methods tend to have better performance than traditional methods. NARM and STAMP perform well on all three datasets. NARM not only uses RNN to model session sequences, but also uses attention mechanisms on each session sequence, indicating different importance of nodes in the session. STAMP extracts long-term interest based on the attention mechanism and takes the last click as short-term interest, which combines short-term and long-term interest to predict user preferences and obtains a more comprehensive representation for the current session. This results demonstrate the effectiveness of combining short-term and long-term preferences by assigning different attention weights to different items in the session.

Among all the baseline methods, GNN-based methods outperform traditional and RNN-based methods in most cases. It takes into account the transitions between items in the current session, models each session as a graph rather than a simple sequence, thus to capture richer information through GNN. SR-GNN demonstrates the effectiveness of applying GNN to extract local information from the session graph. It indicates that modeling local information for a session through GNN is more suitable than RNN. GCE-GNN constructs two graphs based on local and global item transitions, and the results show that this structure is superior to a single graph structure like SR-GNN. But it only models the transition between items, laking the ability to capture larger granularity information. DHCN also leverages both inter and intra session information in hypergraph. However, the results are lower compared to GCE-GNN, indicating that if inter session information is not discreetly exploiting, it may even be a form of noise. SHARE employs the hypergraph to model a session and uses a sliding window on the session to capture its group intention, but only using hypergraph structure is not effective because of the issue of sequential information loss. Disen-GNN learns the item representation at a finer level of granularity, i.e., factor level, and achieves better results than above models. Because it extracts deeper information and represents richer embedding. However, compared to performance improvement, the time and memory consumption brought by this factor level model are also multiplied.

Our approach greatly improves performance because our model fully combines group intention and sequential pattern of a session. Based on the learnt sequential pattern, the most likely item to be visited next is selected from a global level perspective, and subsequent information is injected into the current session representation using the fusion function. To capture the group intention, hypergraphs with sliding windows are constructed so that the concept of group intention. Besides, We use the whole session as a hyperedge to assist in creating hypergraph, which can provide more precise representations. As a result, our approach achieves better performance.

4.6. **Ablation experiments(RQ2).** To verify the contribution of the hypergraph module and the subsequent item injection module to the performance of MM-GNN, ablation experiments are performed in this section. We've done the experiments by removing one of the modules from MM-GNN to analyze its performance change.

- w/o s: Remove the subsequent item injection module from MM-GNN.
- w/o h: Remove the hypergraph learning module from MM-GNN.
- w/o w: Remove the hyperedge with session length from MM-GNN.

Table 3 shows the impact of the different modules. It can be seen that the absence of the hypergraph-based learning module has the greatest impact on the performance of MM-GNN, which indicates that the combination of group intention and sequential pattern mining is effective. The subsequent item injection module helps to improve the performance, especially in the MRR@k metric, but in the P@k metric of the Nowplaying dataset, removing the subsequent item injection module leads to higher results, indicating that although the subsequent information injection this module utilizes extra information that does not exist in the current session, which leads to a higher ranking of the hit items, it also brings a certain degree of long-tail effect, but overall the subsequent item injection module is effective and we should focus more on introducing richer information. The results obtained by merely removing the hyperedge with session length are ranked between w/o h and MM-GNN, which indicates the effectiveness of this strategy.

Table 3. Ablation study results

DATASET	TMALL		NOWPLAYING		DIGINETICA	
Methods	P@20	MRR@20	P@20	MRR@20	P@20	MRR@20
w/o s	**35.22**	15.61	**23.74**	7.92	55.03	19.37
w/oh	33.81	15.26	23.36	8.37	54.64	19.20
w/o w	34.78	15.76	23.03	8.48	54.99	19.41
MM-GNN	35.19	**15.90**	23.28	**8.50**	**55.13**	**19.45**

4.7. **Impact of aggregation operations(RQ3).** When we use multiple session representations, it is significant to compare the sum aggregation operation taken by MM-GNN with other different aggregation operations (i.e., concatenation, gating).

For the concatenation operation, the final representation is the concatenation of the vector S_{nh}, S_{subsq}:

$$S = W_{10}\left([S_{nh} \| S_{subsq}]\right) \tag{23}$$

Where $W_{10} \in R^{d\times 2d}$ is a trainable parameter. For the gating operation, the formula is as follows:

$$\begin{aligned} m &= \sigma\left(W_{11}^{T} S_{nh} + W_{12}^{T} S_{subsq}\right) \\ S &= m S_{nh} + (1-m) S_{subsq} \end{aligned} \tag{24}$$

Where W_{11}, $W_{12} \in R^d$ are trainable parameters, and is used to balance the importance between the vectors.

Figure 3 shows the performance of different aggregation operations on the three datasets. It can be observed that the sum aggregation method we used outperforms the other two aggregation methods in almost all four metrics for the three data sets. The performance of the concatenation method is the worst on the three datasets. The gating method is in between, but it outperforms the sum aggregation method on the Tmall dataset, indicating that the sum method does not fully exploit the performance of the model on certain datasets and that the capabilities demonstrated by the gating mechanism deserve further investigation.

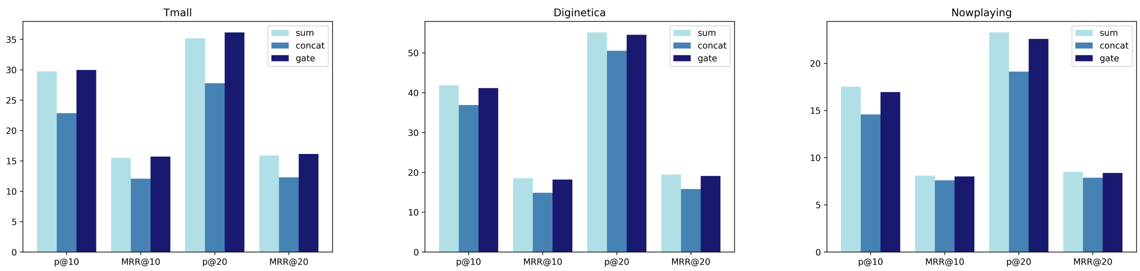

FIGURE 3. Impact of aggregation operations

4.8. **Impact of stacking multiple NEN blocks on results(RQ4).** Previously in our experiments, the hypergraph module used only 1 NEN block, which may limit the performance of the model. In this subsection we will investigate the effect of stacking multiple NEN blocks. We stacked 1, 3, 5, 7, 9, 11, and 13 NEN blocks on the Tmall dataset, and the experimental results are shown in Figure 4. From the figure, we can see that stacking multiple NEN blocks is effective, especially when stacking the second and third blocks, the performance increases dramatically. The performance of P@10 and P@20 metrics of the Tmall dataset is optimal when stacking the seventh NEN block, and the performance of MRR@10 and MRR@20 metrics is optimal when stacking the third NEN block. When there are more NEN blocks, the model performance decreases because more NEN blocks lead to over-smoothing.

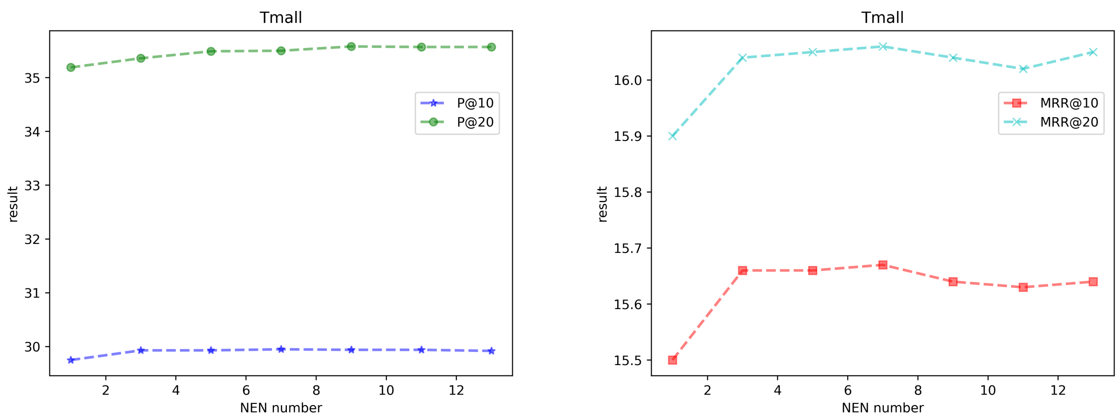

FIGURE 4. Impact of stacking multiple NEN blocks

5. **Conclusions.** Recommendation models that combine graph neural networks and sessions have become a hot topic. In this context, we propose a graph neural network session recommendation model (MM-GNN) that incorporates multi meta information. First, we fuse normal graph and hypergraph to achieve dual information capture of group intention and sequential behavior. Then, we propose the subsequent item injection module to inject possible next information of the last item of the current session at the global level. Extensive experimental and empirical studies have demonstrated the validity of our framework and shown it to be superior to other recent baselines.

Acknowledgment. This work is supported by National Key Research and Development Program of China (No.2022ZD0119501); National Natural Science Foundation of China (No.52374221); Natural Science Foundation of Shandong Province (Nos.ZR2021MG038, ZR2022MF288, ZR2023MF097); Taishan Scholar Foundation of Shandong Province (No.ts20190936); Special Study on Cultural Tourism of Shandong Social Science Planning (No.21CLYJ32); SDUST Intelligent Science and Security Governance Innovation Team; Education Ministry Humanities and Social Science Research Planning Fund Project of China ("Personalized Learning Path Recommendation Driven by Multi-Source Educational Data and Its Quantitative Evaluation" with grant number 23YJAZH192).

REFERENCES

[1] 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*. Association for Computing Machinery, 2010, pp. 811–820.

[2] P. Wang, J. Guo, Y. Lan, J. Xu, S. Wan, and X. Cheng, "Learning hierarchical representation model for nextbasket recommendation," in *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*. Association for Computing Machinery, 2015, pp. 403–412.

[3] G. Shani, D. Heckerman, R. I. Brafman, and C. Boutilier, "An mdp-based recommender system," *Journal of Machine Learning Research*, vol. 6, no. 9, pp. 1265–1295, 2005.

[4] W. Wang, H. Yin, S. Sadiq, L. Chen, M. Xie, and X. Zhou, "Spore: A sequential personalized spatial item recommender system," in *2016 IEEE 32nd International Conference on Data Engineering (ICDE)*. IEEE, 2016, pp. 954–965.

[5] 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. 1, 2019, pp. 346–353.

[6] 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*. Association for Computing Machinery, 2020, pp. 169–178.

[7] C. Chen, J. Guo, and B. Song, "Dual attention transfer in session-based recommendation with multi-dimensional integration," in *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. Association for Computing Machinery, 2021, pp. 869–878.

[8] R. Qiu, H. Yin, Z. Huang, and T. Chen, "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*. Association for Computing Machinery, 2020, pp. 669–678.

[9] P. Zhang, J. Guo, C. Li, Y. Xie, J. B. Kim, Y. Zhang, X. Xie, H. Wang, and S. Kim, "Efficiently leveraging multi-level user intent for session-based recommendation via atten-mixer network," in *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*. Association for Computing Machinery, 2023, pp. 168–176.

[10] 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)*. SIAM, 2021, pp. 82–90.

[11] R. Salakhutdinov and A. Mnih, "Bayesian probabilistic matrix factorization using markov chain monte carlo," in *Proceedings of the International Conference on Machine Learning*, vol. 25. Association for Computing Machinery, 2008, pp. 880–887.

[12] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, 2009.

[13] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk, "Session-based recommendations with recurrent neural networks," *arXiv preprint arXiv:1511.06939*, 2015.

[14] J. Li, P. Ren, Z. Chen, Z. Ren, T. Lian, and J. Ma, "Neural attentive session-based recommendation," in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. Association for Computing Machinery, 2017, pp. 1419–1428.

[15] 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*. Association for Computing Machinery, 2018, pp. 1831–1839.

[16] Q. Tan, J. Zhang, J. Yao, N. Liu, J. Zhou, H. Yang, and X. Hu, "Sparse-interest network for sequential recommendation," in *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*. Association for Computing Machinery, 2021, pp. 598–606.

[17] Q. Guo, X. Qiu, P. Liu, Y. Shao, X. Xue, and Z. Zhang, "Star-transformer," *arXiv preprint arXiv:1902.09113*, 2019.

[18] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in Neural Information Processing Systems*, vol. 30, pp. 6000–6010, 2017.

[19] W.-C. Kang and J. McAuley, "Self-attentive sequential recommendation," in *2018 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2018, pp. 197–206.

[20] 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. Assoc Advancement Artificial Intelligence, 2021, pp. 4503–4511.

[21] Y. Pang, L. Wu, Q. Shen, Y. Zhang, Z. Wei, F. Xu, E. Chang, B. Long, and J. Pei, "Heterogeneous global graph neural networks for personalized session-based recommendation," in *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*. Association for Computing Machinery, 2022, pp. 775–783.

[22] S. Lai, E. Meng, F. Zhang, C. Li, B. Wang, and A. Sun, "An attribute-driven mirror graph network for session-based recommendation," in *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. Association for Computing Machinery, 2022, pp. 1674–1683.

[23] R. Yeganegi and S. Haratizadeh, "Star: A session-based time-aware recommender system," *arXiv preprint arXiv:2211.06394*, 2022.

[24] Y. Hou, B. Hu, Z. Zhang, and W. X. Zhao, "Core: simple and effective session-based recommendation within consistent representation space," in *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. Association for Computing Machinery, 2022, pp. 1796–1801.

[25] A. Li, Z. Cheng, F. Liu, Z. Gao, W. Guan, and Y. Peng, "Disentangled graph neural networks for session-based recommendation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 8, pp. 7870–7882, 2023.

[26] T.-Y. Wu, H. Li, S. Kumari, and C.-M. Chen, "A spectral convolutional neural network model based on adaptive fick's law for hyperspectral image classification," *Computers, Materials & Continua*, vol. 79, no. 1, pp. 19–46, 2024.

[27] Y. Ma, Y. Peng, and T.-Y. Wu, "Transfer learning model for false positive reduction in lymph node detection via sparse coding and deep learning," *Journal of Intelligent & Fuzzy Systems*, vol. 43, no. 2, pp. 2121–2133, 2022.

[28] S.-M. Zhang, X. Su, X.-H. Jiang, M.-L. Chen, and T.-Y. Wu, "A traffic prediction method of bicycle-sharing based on long and short term memory network." *Journal of Network Intelligence*, vol. 4, no. 2, pp. 17–29, 2019.

[29] E. Zangerle, M. Pichl, W. Gassler, and G. Specht, "# nowplaying music dataset: Extracting listening behavior from twitter," in *Proceedings of the First International Workshop on Internet-scale Multimedia Management*. Association for Computing Machinery, 2014, pp. 21–26.

[30] 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*. Association for Computing Machinery, 2021, pp. 2180–2190.