ISSN 2616-5775 Vol. 6, Issue 6: 37-48, DOI: 10.25236/AJCIS.2023.060606

# Category-Intent Fusion Graph Neural Network for Session-based Recommendation

**Fei Chu***

*School of Computer and Information Technology, Beijing Jiaotong University, Beijing, China*
**Corresponding author: anunverse@163.com*

***Abstract:*** *A user's interaction with an item is determined by a combination of intentions, such as a desire to purchase an electronic device while following a trend. However, these intentions are often unobservable, making it difficult to model user intentions and improve session recommendations. To tackle this problem, we propose a novel approach called the Category-Intent Graph Neural Network (CIGNN), which leverages the relationship between item categories and user intentions to provide accurate recommendations. We translate the category information into a compact representation, which represents the user's intent, and construct a category-intent fusion graph with item, category, and intent nodes. This graph connects multiple potential intents for each item in a session to capture user intentions and increase the expressiveness of item representations. The CIGNN model transfers information between intent, item, and category nodes, updating their representations alternately. Our experimental results on three benchmark datasets demonstrate the superiority of the CIGNN model over state-of-the-art (SOTA) methods in session-based recommendation (SBR).*

***Keywords:*** *Session-based Recommendation, Graph Neural Network, Deep Learning*

## 1. Introduction

In the modern society of information explosion, recommendation systems have become an essential tool for accessing information. These systems have demonstrated remarkable efficacy across a spectrum of domains, including e-commerce, social media, and numerous others. The efficacy of recommendation systems is largely dependent upon their ability to make informed recommendations based on comprehensive historical user behavior and personal information. However, in instances where this information is inaccessible, for example, due to anonymous logins or privacy regulations, the quality of recommendations may be negatively impacted. To address this issue, the field of recommendation systems is increasingly exploring the use of session-based recommendations, which rely on brief and anonymous user interaction sequences to predict the next item a user is likely to select.

Early approaches used Markov chains[1] to predict the user's next clicked item based on the previously clicked item, with limited prediction accuracy. Later, many deep learning-based session recommendation methods were proposed with the development of deep learning. The RNN class-based methods[2] treat each session as a series of items ordered by click time and solve for prediction by recurrent neural networks (RNNs). Benefiting from the advantages of recurrent neural networks in modeling sequential dependencies, RNN-based methods[3,4] have achieved significant results. However, these methods are insufficient to obtain an accurate representation of the user in a session and ignore the complex transformation relationships of the items.

In recent years, with the development of graph neural networks, more and more approaches have used GNNs to capture the complex information transfer between items in a session. SRGNN[5] first proposed constructing graphs over sessions and using GNNs to capture the transfer relationships between items in a session with remarkable results. While existing approaches have achieved good results[6,7], they assume that the user's primary intent in a session usually remains the same and ignore that the user has multiple intentions to decide together when interacting with the items. Even in relatively short sessions, users have many fine-grained interests intertwined with items. Furthermore, these more fine-grained interests are closely related to the user clicks on items category information. For example, considering the example session in Figure 1, the user clicks on a computer, an electronic product, reflecting the user's intent to update productivity tools, follow trends, etc. Later, the user clicks on clothes, which belong to the wear category, reflecting that the user may want to buy wear, dress up or pursue fashion. These analyses depict the following challenges: 1) Users' click intent is primarily related to the category of item. How to model

ISSN 2616-5775 Vol. 6, Issue 6: 37-48, DOI: 10.25236/AJCIS.2023.060606

the user's intent representation from the session; 2) Users click on items with multiple intents. How to identify the primary intent of users from them; 3) The items that users click on are constantly changing, how to model changes in user intent that intertwine with items.

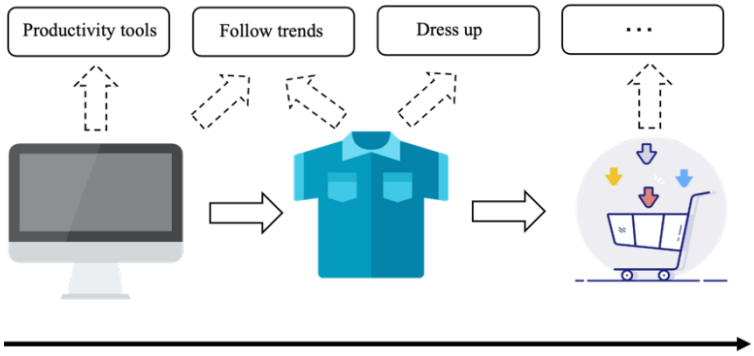


*Figure 1: Users' interaction can be driven by multiple underlying intents.*

We propose a simple model called Category-Intent fusion graph neural network (CIGNN) to tackle the challenges mentioned above. Initially, we merge each item's category information with its representation to enable the model to grasp the category information. To model the user's intent, we project the category representations of all items to obtain all possible intent representations that the user may have, because the user's intent and the item's category are closely intertwined. To account for changes in user intent during a session, we introduce multiple intent nodes to a graph consisting of complex interaction sessions. We then map each item in the session to the corresponding primary intents and interact with those intents. Specially, we aggregate multiple primary intents to obtain a mixed intent corresponding to each item. By the links between mixing intent nodes and item nodes, the model can focus on primary intents and ignore secondary intents. We further aggregate the mixed intents of each item in a session to derive a global intent representation. We update the item, category, and intent representations iteratively using CIGNN with graph neural networks and attention mechanisms. Finally, we aggregate the item representations to recommend the following item. Our experiments encompass three publicly available benchmark datasets, i.e., Yoochoose, Diginetica, and Nowplaying. Experimental results demonstrate that the proposed CIGNN model outperforms state-of-the-art baselines in terms of P@20 and MRR@20.

The main contributions of this work are summarized as follows.

(1) To the best of our knowledge, this is the first work to integrate the user's intents into the session graph as nodes display to investigate the association between items and user intents. As we consider the user's intent to be closely related to the category of the item, we abstract the representation of intent directly from the category of the item.

(2) We constructed a category-intent graph that not only explicitly models the historical transitions and item-category relations but also covers the connections of item-intent nodes. Moreover, the model generates a global intent node for each session to capture the overall intent, enhancing the model's understanding of the relationships between local and global intents.

(3) We proposed a novel update module, CIGNN, that utilizes graph neural networks and attention mechanisms to update the item, category, and intent nodes iteratively in the category-intent graph. With this module, the model can propagate information between items that are not adjacent but share a common intent.

(4) Extensive experiments show that CIGNN outperforms the state-of-the-art baselines and achieves significant improvements on three real-world datasets.

## 2. Related Work

### *2.1. Session-based Recommendation*

Session-based recommendation (SBR) aims to reflect dynamic user preferences to provide more relevant and accurate recommendations[8]. Early SBR mainly used Markov chains to capture the sequence signals in interactions. It can learn the transition probabilities between items from the user's interaction

ISSN 2616-5775 Vol. 6, Issue 6: 37-48, DOI: 10.25236/AJCIS.2023.060606

history[9] and outperforms traditional item-based recommendations on large-scale e-commerce datasets. PME[10] proposed a Personalized Markov Embedding strategy to embed songs and users into a Euclidean space to reflect their relationship strength, and demonstrated its effectiveness on real-world datasets. As RNNs are effective in modeling sequential data, the use of RNN-based models to model the user's interaction history in session-based recommendation (SBR) has gradually increased[11]. GRU4Rec[2] view each session as a sequence of items arranged by click time and use GRU to make predictions and achieve significant results.

In recent years, several studies have explored the utilization of GNNs in modeling complex transitions within or between sessions with promising results[12,13]. SRGNN[5] were the pioneers in modeling user sessions as a graph and using GNNs to capture the complex relationships between items, achieving remarkable performance. FGNN[14] introduced a weighted attention graph layer to facilitate the learning of item and session embeddings for next item recommendation. GC-SAN[15] dynamically constructed a graph structure for session sequences and employed the self-attention network and GNN to capture global and local dependencies, respectively. Furthermore, LESSR[7] proposed a lossless encoding scheme to tackle the issue of lossy session encoding and devised a shortcut graph attention layer to accommodate long-range dependencies. Despite the promising results obtained from these studies, modeling users' multiple intents and item associations remains challenging due to the short session lengths and limited information.

### *2.2. Category Information and User Intent for Recommendation*

The utilization of category information, as a crucial auxiliary information for items, has been extensively investigated in various other domains of recommendation[16-18]. CoCoRec[19] uses item category to enhance dependency modeling based on the user's past actions and to retrieve users with similar preferences. The method makes use of self-attention and attention to capture in-category transition patterns, achieved a good result in category-aware recommendation. Specially, IAGNN[20] introduced an intention adaptive graph neural network, which utilizes the relationship between items and their categories to improve recommendation accuracy in scenarios where a user specifies a target category. However, our model operates differently by not requiring the categorization information of the target item to generate recommendations, and captures the underlying user intentions manifested through the items.

Recently, many methods have studied user intent to improve recommendation performance[21-24]. Some work[25] incorporate multiple future interactions as guidance and an intent variable from both the user's historical and future behavior sequences. This intent variable helps to capture the interdependence between the individual's past and future behavior patterns in the sequential recommendation. In the SBR task, some methods also focus on user intent research. ICM-SR[26] proposed a session encoder to model both the sequential signal and the recent interest in the session, and captured the user's intent from the current session for detecting correct neighbor sessions as auxiliary information. MCPRN[27] think sessions may often contain multiple item subsets with distinct purposes. They proposed a mixture-channel model that represents multi-purpose sessions by accommodating these distinct item subsets, with mixture-channel purpose routing networks to identify the purpose of each item and a purpose-specific recurrent network to model dependencies within each channel. However, multiple intents influence the determination of users' interactions with an item. Unlike these work, we set and filter out multiple intent nodes for each item and model the relationship between items and intents, enhancing the expressiveness of item representation.

## 3. Methods

### *3.1. Problem Statement*

Session recommendation aims to predict the next clicked item based on the ongoing session. We formalize this task as follows. Let $V = \{v_1, v_2, \dots, v_{|V|}\}$ denote the items in all sessions, where $|V|$ denotes the number of different items. Given a session $S = [v_1^s, v_2^s, \dots, v_m^s]$ denote the list of items sorted by timestamp, where $v_i^s \in V$ represents the $i$-th clicked item within the session $S$. The goal of session recommendation is to predict the next clicked item $v_{m+1}^s$ for a given session $S$.

ISSN 2616-5775 Vol. 6, Issue 6: 37-48, DOI: 10.25236/AJCIS.2023.060606

### *3.2. Overall Architecture*

The architecture of the model CIGNN is initially depicted in Figure 2. Firstly, the embedding layer is initialized for all items and categories to obtain the corresponding ID embeddings. Following this, the category embeddings of all items are projected into a distinct space as the initialized representation of the intent nodes. Subsequently, the Category-Intent Fusion Graph is established by integrating item category information and user intent representation to model the intricate relationship between user intents and items. An update module, comprising a graph neural network and an attention mechanism, is utilized to facilitate information transmission and update each node's representation in the Category-Intent Fusion Graph. Lastly, an embedding fusion layer and a prediction layer are employed to make predictions regarding the next item that will be clicked for a given session $S$.

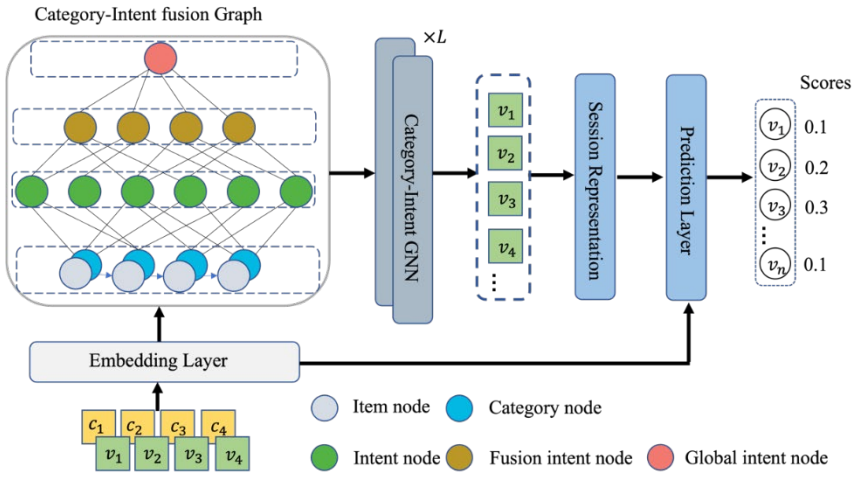


*Figure 2: The workflow of CIGNN.*

### *3.3. Category-Intent Fusion Graph Construction*

Previous recommendation models have rarely integrated category information of items and user intention information explicitly into the recommendation process. To address this gap, we propose a novel approach that incorporates user intention nodes and item category nodes into the model using graph neural networks to model the relationship between different nodes. Specifically, we construct a directed heterogeneous graph $G_s = (V_s, E_s)$, where $V_s = (\{v_1, v_2, \dots, v_n\}, \{c_1, c_2, \dots, c_n\}, \{t_1, t_2, \dots, t_k\}, \{f_1, f_2, \dots, f_n\}, p)$ encompasses $n$ item nodes $v$, category nodes $c$ and mixed intent nodes $f$, $k$ intent nodes $t$, and a global intent node $p$ that symbolizes the overall session intention. $E_s$ are composed of edges representing item transitions, category-item connections, mixed intent-intent connections, mixed intent-item connections, item-intent transitions, category-intent transitions, and global intent-mixed intend transitions. We describe each of these seven edges relationship in detail below.

**Item Transitions.** For each session $S = \{v_1, v_2, \dots, v_n\}$, we can construct a weight matrix $A^S = [A_O^S; A_I^S]$ to model the relationship of the items in the session according to SRGNN[5], where $[;]$ is the concatenation operation, and $A_O^S \in R^{N \times N}$ and $A_I^S \in R^{N \times N}$ are the normalized weight matrices consisting of each item's outgoing and incoming edges, respectively.

**Category-Item Connections.** In a session $S$, a bidirectional edge can be established between each item node $v_i$, and its respective category node $c_{v_i}$. By implementing such connections between item and category nodes, the category information can be integrated into the item representation.

**Item-Intent Transitions and Category-Intent Transitions.** The mapping of categories of all items to $K$ intention nodes, where $K$ represents the total number of intentions for all users, is employed to incorporate users' intentions into the model. To further enhance the accuracy of modeling the intricate intent of users upon clicking an item, an attention mechanism is utilized to select the h highest weighted intents out of the $K$ intents for each item, where $h$ is a hyperparameter. These $h$ intentions are the primary intentions that the user has when clicking on the item. In addition, we incorporates unidirectional edges that connect each item to its potential h intent nodes. These edges facilitate the update of intent nodes by utilizing the information gathered from the item nodes and their corresponding category nodes.

**Mixed Intent-Intent Connections and Mixed Intent-Item Connections.** During a session, users

ISSN 2616-5775 Vol. 6, Issue 6: 37-48, DOI: 10.25236/AJCIS.2023.060606

may have several intents for each item node they click. So we aggregate the $h$ intent nodes to create a mixed intent node $f$ for each item. This mixed intent node allows the model to focus on the primary intent and disregard any secondary intent from the user. The transition between mixed intents can be propagated back to all relevant intent nodes, ultimately spreading to all relevant items. Furthermore, the model can extract higher-level information via the edges between mixed intent and items and between mixed intent and intents. The propagation of information between items with the same intentions, even if they are not adjacent in the session, can be facilitated through the intent nodes, thereby enhancing the expressiveness of the item representation.

**Global Intent-Intent Transitions.** For a given session $S$, we aggregate the mixed intent corresponding to each item in the session to obtain a global intent node representation $p$ as the global intention of session $S$. By adding edges from the global intent node $p$ to the item nodes, we can model the relationship between each item and the global intention. It minimizes the information propagation distance between non-adjacent items, and enhances the model's ability to grasp the delicate interplay between localized and overarching intents.

### *3.4. Category-Intent fusion Graph Neural Networks*

Next, we present how category-intent fusion graph neural networks propagate information between different nodes.

**Embeddings.** For given a session $S$, we embed every item $v \in S$ into a dense embedding space $\boldsymbol{E_V} \in \mathbb{R}^{\boldsymbol{n\times d}}$, using $\boldsymbol{v_i}$ to denote the embedding representation of item $v_i \in S$, where $d$ is the dimensionality of the item embedding. Similarly, we embed each category $c$ into a dense embedding space $\boldsymbol{E_C} \in \mathbb{R}^{\boldsymbol{n\times d}}$, using $\boldsymbol{c_i}$ to denote the embedding representation of category $c_i$. We map the embedding layer $E_C$ of categories onto $K$ vectors to obtain the embedding $\boldsymbol{E_T}$ for representing $K$ user intents. In addition, we add category embeddings to each item as the initialization embeddings of items to enhance the expressiveness of the item representation.

$$\boldsymbol{v_i = W_1([v_i; c_i])} \tag{1}$$

where $[;]$ is the concatenation operation, $\boldsymbol{W_1} \in \mathbb{R}^{\boldsymbol{d\times 2d}}$ is the projection matrix utilized to maintain the dimension $v_i$.

**Update.** Firstly, for item-item transitions, we use a gated graph neural network (GGNN) [5] to update the representation of each item node. For node $v_i^s$ in graph $G_s$, the update function is as follows.

$$\boldsymbol{a_i^{s,l} = M_{i:}^s\left[v_1^{s,l-1}, \ldots, v_n^{s,l-1}\right]^\top W_2 + b} \tag{2}$$

$$\boldsymbol{z_i^{s,l} = \sigma\left(W_z a_i^{s,l} + U_z v_i^{s,l-1}\right)} \tag{3}$$

$$\boldsymbol{r_i^{s,l} = \sigma\left(W_r a_i^{s,l} + U_r v_i^{s,l-1}\right)} \tag{4}$$

$$\boldsymbol{\widetilde{v_i^{s,l}} = \tanh\left(W_o a_i^{s,l} + U_o\left(r_i^{s,l} \odot v_i^{s,l-1}\right)\right)} \tag{5}$$

$$\boldsymbol{v_i^{s,l} = \left(1 - z_i^{s,l}\right) \odot v_i^{s,l-1} + z_i^{s,l} \odot \widetilde{v_i^{s,l}}} \tag{6}$$

where $\boldsymbol{W_2, W_z, W_r, W_o} \in \mathbb{R}^{\boldsymbol{d\times 2d}}$, and $\boldsymbol{U_z, U_r, U_o} \in \mathbb{R}^{\boldsymbol{d\times d}}$ controls the weights, and $\boldsymbol{b} \in \mathbb{R}^{\boldsymbol{d}}$ is a bias vector. $\boldsymbol{z_i^s, r_i^s}$ are the reset and the update gates, which decide what information to be preserved and discarded, respectively. $\boldsymbol{\sigma}(\cdot)$ is the sigmoid function, and $\odot$ is the element-wise multiplication operator, $\boldsymbol{l}$ is the $l$-th layer in CIGNN.

Then, we select d intentions from all $K$ intents representations for each item to model the information transfer between items and intentions. Specifically, we use the following soft attention mechanism.

$$\boldsymbol{v_i^{l'} = W_3\left(ReLU\left(v_i^l + c_i^l\right)\right)} \tag{7}$$

$$\boldsymbol{\alpha = Softmax\left(\frac{(W_T T)(W_v v_i^{l'})^T}{\sqrt{d}}\right)} \tag{8}$$

where $\boldsymbol{T} \in \mathbb{R}^{\boldsymbol{K\times d}}$ represents the representations of all intent nodes, while $\boldsymbol{W_2, W_T, W_v} \in \mathbb{R}^{\boldsymbol{d\times d}}$ denote the learnable parameter matrices.

ISSN 2616-5775 Vol. 6, Issue 6: 37-48, DOI: 10.25236/AJCIS.2023.060606

Then, we select the intent node corresponding to the first d large α as the primary intent that the user has when clicking on the item. Therefore, we can re-normalize these d intents based on their weights and aggregate them to obtain the corresponding mixed intent $f$ for each item. We consider the information from mixed intent $f$ to item node $v$ by a gated network as follow.

$$\boldsymbol{g_1} = \boldsymbol{\sigma}\left(\boldsymbol{W_4}\left[\boldsymbol{f_i^{s,l}}; \boldsymbol{v_i^{s,l}}\right]\right) \tag{9}$$

$$\boldsymbol{v_i^{s,l}} = \boldsymbol{g_1 v_i^{s,l}} + (\boldsymbol{1} - \boldsymbol{g_1})\boldsymbol{f_i^{s,l}} \tag{10}$$

where $\boldsymbol{W_4} \in \mathbb{R}^{\boldsymbol{2d \times d}}$ are learnable parameters and $[;]$ is the concatenation operation.

Consider the messaging related to category nodes. We use the gated neural network in Equation (9) and Equation (10) to aggregate the item node representation before the update and the corresponding category node representation, as the intermediate state of the item representation, denoted as $v'$. After that, we compute the similarity weights α between the intermediate state $v'$ of the item and all $K$ intent nodes, and normalize the calculated weight α from the item side to obtain weighted $\boldsymbol{q}$, which represent the degree of contribution of different items to each intent node. By aggregating the item node representations and category node representations corresponding to the items, weighted by the normalized weights $q$, we obtain the information representation $\boldsymbol{t_{c,j}^{s,l}}$ from category nodes and the information representation $\boldsymbol{t_{v,j}^{s,l}}$ from item nodes. Similarly, we use the gating network in Equation (9) and Equation (10) to fuse the category information representation $\boldsymbol{t_{c,j}^{s,l}}$ and the item node information representation $\boldsymbol{t_{v,j}^{s,l}}$, thereby obtaining the updated intent node representation $\boldsymbol{t_j^{s,l}}$. The update function is shown below.

$$\boldsymbol{t_{v,j}^{s,l}} = \boldsymbol{q_j v^{s,l}},\ \boldsymbol{t_{c,j}^{s,l}} = \boldsymbol{q_j c^{s,l}} \tag{11}$$

$$\boldsymbol{t_j^{s,l}} = \boldsymbol{ReLU}\left(\boldsymbol{W_5}\left[\boldsymbol{t_{v,j}^{s,l}}; \boldsymbol{t_{c,j}^{s,l}}\right]\right) \tag{12}$$

$$\boldsymbol{g_2} = \boldsymbol{\sigma}\left(\boldsymbol{W_6}\left[\boldsymbol{t_j^{s,l}}; \boldsymbol{t_j^{s,l-1}}\right]\right) \tag{13}$$

$$\boldsymbol{t_{s,j}^{l}} = \boldsymbol{g_2 t_{s,j}^{l-1}} + (\boldsymbol{1} - \boldsymbol{g_2})\boldsymbol{t_j^{s,l}} \tag{14}$$

where $\boldsymbol{W_5}, \boldsymbol{W_6} \in \mathbb{R}^{\boldsymbol{2d \times d}}$ are learnable parameters, $\boldsymbol{t_j^{s,l}}$ is the candidate embedding for the intent node. $\boldsymbol{q_j}$ is the normalized weight of the item corresponding to the $j$-th intent in the session, and aggregate $\boldsymbol{t_j^{s,l}}$ *and* $\boldsymbol{t_j^{s,l-1}}$ to update and obtain the intent representation $\boldsymbol{t_{s,j}^{l}}$.

In addition, to learn the representation of the intent nodes more accurately, we apply an average pooling on mixed intent $f_i^{s,l}$ for each item in this session to obtain a global intent representation $\boldsymbol{p^{s,l}}$ for the entire session as follow.

$$\boldsymbol{p^{s,l}} = \frac{\boldsymbol{1}}{\boldsymbol{m}} \sum_{\boldsymbol{i=1}}^{\boldsymbol{m}} \boldsymbol{f_i^{s,l}} \tag{15}$$

The gating network is employed to transmit the information of $p^s$ to all $K$ intent nodes. The model acquires more expressive item representations through multiple iterations of updating item nodes, category nodes, and intent nodes. However, multi-layer GNNs may lead to overfitting problem[28] in graph neural networks. To alleviate this problem, we use the highway network[6] to aggregate the output of the last layer of the module with the initial input as the final item representation in the following.

$$\boldsymbol{g_3} = \boldsymbol{\sigma}\left(\boldsymbol{W_s}\left[\boldsymbol{v_i^{s,0}}; \boldsymbol{v_i^{s,l}}\right]\right) \tag{16}$$

$$\boldsymbol{v_i^s} = \boldsymbol{g v_i^{s,0}} + (\boldsymbol{1} - \boldsymbol{g_3})\boldsymbol{v_i^{s,l}} \tag{17}$$

where $\boldsymbol{W_s} \in \mathbb{R}^{\boldsymbol{2d \times d}}$ are learnable parameters, and $\boldsymbol{v_i^s}$ is the final representation of the item $v_i^s$.

### *3.5. Session Representation and Prediction Layer*

In order to incorporate the sequential information into CIGNN, we add learned position embeddings $\boldsymbol{z} \in \mathbb{R}^{\boldsymbol{n \times d}}$ to the item representations, i.e., $\boldsymbol{v} = \boldsymbol{v} + \boldsymbol{z}$. We then take the representation of the last item $\boldsymbol{v_n}$ as the local embedding $\boldsymbol{s_l}$ of the session $S$. Then, we aggregate all items embeddings of the session as the global preference embedding $\boldsymbol{s_g}$. Adopting the soft-attention mechanism to learn their priority, we hybrid the local and the global embeddings $\boldsymbol{s_l}$ and $\boldsymbol{s_g}$ as below.

ISSN 2616-5775 Vol. 6, Issue 6: 37-48, DOI: 10.25236/AJCIS.2023.060606

$$g_i = W_6^T \sigma(W_7 v_n + W_8 v_i + b) \tag{18}$$

$$s_g = \sum_{i=1}^{n} g_i v_i \tag{19}$$

$$s_h = W_9 [s_l; s_g] \tag{20}$$

where $W_6, W_7, W_8, W_9 \in \mathbb{R}^{d \times d}$ are learnable parameters. We then obtain the final recommendation probability of the item as below.

$$\hat{y} = Softmax(s_h^{\top} v) \tag{21}$$

We use the cross-entropy of the prediction results $\hat{y} = \{\hat{y}_1, \cdots, \hat{y}_{|V|}\}$ and the ground truth labels $y$ as the main loss defined in the following.

$$L_{rec}(\hat{y}) = -\sum_{i=1}^{|V|} y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i) \tag{22}$$

## 4. Experiments

### *4.1. Experimental Settings*

We conduct our experiments on three benchmark datasets, Diginetica, Yoochoose and Nowplaying. We set the sessions of the latest data (such as, the data of the last week) as the test data, the remaining data for training and validation. Following the previous work[5], we filter out sessions of length 1 and items that appear less than 5 times over all datasets. Due to the large size of Yoochoose, we utilize the recent 1/64 and 1/4 fractions of the training sequences, denoted as Yoochoose1/64 and Yoochoose1/4, respectively. Furthermore, for session $S = [(v_1, c_1), (v_2, c_2), \ldots, (v_m, c_m)]$, we generate a series of sequences and labels $([(v_1, c_1)], v_2)$, $([(v_1, c_1), (v_2, c_2)], v_3)$, ... , $([(v_1, c_1), (v_2, c_2), \ldots, (v_{m-1}, c_{m-1})], v_m)$. The statistics of datasets are summarized in Table 1.

In addition, the Yoochoose dataset is comprised of only 12 item categories, however, certain items are designated with distinctive attributes such as special offers or item brands. In our experiments, these distinct characteristics are considered a separate category, hence the Yoochoose dataset encompasses a total of 337 categories.

*Table 1: Dataset statistics.*

| Dataset | Diginetica | Yoochoose1/64 | Yoochoose1/4 | Nowplaying |
|---|---|---|---|---|
| #Training Sessions | 719,470 | 369,859 | 5,917,746 | 825,304 |
| #Test Sessions | 60,858 | 55,898 | 55,898 | 89,824 |
| #Items | 43,097 | 16,766 | 29,618 | 60,417 |
| #Category Num. | 996 | 337 | 337 | 11,462 |
| #Average Lengths | 5.12 | 6.16 | 5.71 | 7.42 |

In evaluating recommendation performance, we utilize two well-established ranking-based metrics, P@K and MRR@K. The P@K metric assesses the presence of the target item within the top-K recommended items, while the MRR@K metric considers the target item's position in the recommended items list. A higher value of the metrics indicates a greater accuracy in ranking. Additionally, we compare the efficacy of our model with the following session recommendation models to validate its superiority.

**Item-KNN**[29] recommends items similar to the previously clicked item in the session, where the cosine similarity between the vector of sessions is used.

**FPMC**[9] factorizes the transition matrix of a Markov Chain into two smaller matrices for a recommendation.

**GRU4Rec**[2] uses Gated Recurrent Unit (GRU) to model sequential behavior for session recommendation.

**NARM**[4] employs RNNs with attention mechanisms to capture the user's primary purpose.

**STAMP**[30] uses an attention mechanism to weigh the importance of each item in a user's sequence, allowing it to make recommendations based on the historical behavior and the current context of a user.

**SRGNN**[5] firstly model session sequences as graph-structured data and uses GNN to capture complex transitions of items.

**FGNN**[14] considers the sequence order and the latent order, and formulate the next item

ISSN 2616-5775 Vol. 6, Issue 6: 37-48, DOI: 10.25236/AJCIS.2023.060606

recommendation as a graph classification problem.

**SGNN-HN**[6] adds a star node for each session and applies a star graph neural network to model the transition relationship between items.

**NISER+**[31] using normalized item and session-graph representations for recommendation.

**STAR**[32] appends multiple interest nodes to construct a multi-interest graph for a session and distill multi-interest representations with the injection of multi-form temporal information.

The hyperparameters are selected on the validation set, which is randomly selected from the training set with a proportion of $10\%$. For a general setting, the embedding size is 256, the batch size is 100, and each session is truncated within a maximum length of 10. We adopt the Adam optimizer with an initial learning rate $1e^{-3}$ and a decay factor of 0.1 for three epochs. Moreover, the $L_2$ regularization is $10^{-5}$.

### *4.2. Results and Analysis*

#### *4.2.1. Overall Results*

The experimental results of the overall performance are presented in Table 2. The best performing method is shown in bold, and the second-best performing method is shown with an underline. '*' indicates the statistical significance for $p < 0.01$ compared to the best baseline method with paired t-test.. Our model (CIGNN) consistently achieves good performance (statistical significance) for both evaluation metrics on all datasets, validating the superiority of our model. From these results, we can draw the following conclusions.

*Table 2: Performances of all comparison methods.*

| Method | Yoochoose1/64 | | Diginetica | | Nowplaying | | Yoochoose1/4 | |
|---|---|---|---|---|---|---|---|---|
| | P@20 | M@20 | P@20 | M@20 | P@20 | M@20 | P@20 | M@20 |
| Item-KNN | 51.60 | 21.81 | 35.75 | 11.57 | 15.94 | 4.91 | 52.31 | 21.70 |
| FPMC | 25.99 | 13.38 | 32.37 | 13.82 | 13.10 | 7.12 | - | - |
| GRU4REC | 60.64 | 22.89 | 29.45 | 8.33 | 7.92 | 4.48 | 59.53 | 22.60 |
| NARM | 68.32 | 28.63 | 49.70 | 16.17 | 18.59 | 6.93 | 69.73 | 29.23 |
| STAMP | 68.74 | 29.67 | 45.64 | 14.32 | 17.66 | 6.88 | 70.44 | 30.00 |
| SR-GNN | 70.57 | 30.94 | 50.73 | 17.59 | 18.87 | 7.47 | 71.36 | 31.89 |
| FGNN | 71.12 | 31.68 | 51.36 | 18.47 | 18.78 | 7.15 | 71.97 | 32.54 |
| SGNNHN | <u>72.06</u> | <u>32.61</u> | <u>55.67</u> | <u>19.45</u> | <u>23.29</u> | <u>8.61</u> | <u>72.85</u> | 32.55 |
| NISER+ | 71.27 | 31.61 | 53.39 | 18.72 | 17.76 | 7.85 | 71.80 | 31.80 |
| STAR | 71.31 | 31.30 | 53.98 | 18.66 | 21.98 | 7.88 | 72.46 | <u>32.70</u> |
| CIGNN | **72.13*** | **32.98*** | **55.94*** | **19.65*** | **23.82*** | **9.10*** | **72.98*** | **33.02*** |

(1) Some methods that consider temporal information, such as GRU4REC, NARM, STAMP, SR-GNN, improve performance compared to traditional methods like FPMC. This highlights the significance of sequential patterns in these systems and underscores the efficacy of deep learning models in achieving superior results.

(2) GNN-based models, such as SR-GNN, LESSR, , SGNNHN, and FGNN, demonstrate superior performance compared to RNN-based methods, thereby demonstrating the effectiveness of GNN in SBR. Additionally, SGNNHN produces exceptional results due to its utilization of star nodes, which facilitate the capture of inter-item relationships and enhance information exchange between distant items.

(3) The proposed model, CIGNN, demonstrates superior performance compared to all baseline models across all datasets. The improvement in the performance of the CIGNN model can be attributed to three key factors. Firstly, the model abstracts the user's intention by incorporating category information of the items, as the category information of the item clicked by the user can effectively reflect the user's intention. We abstract the intentions of all users and explicitly add them to the graph neural network as nodes to establish connections with item nodes. Secondly, an attention mechanism is employed to map each item in a session to its corresponding intent, and to construct a mixed intent representation for each item, indicating that a mixture of multiple intents influences a user's click on an item. The user may have the same intention when clicking on different items and may also click on multiple items for the same intention. The model can effectively transmit information between items with the same intention by creating mixed intent nodes and a global intent node for the session. Finally, the item and category nodes are iteratively updated using a gated graph neural network and an attention mechanism. Overfitting in the graph neural network is mitigated through a highway network.

ISSN 2616-5775 Vol. 6, Issue 6: 37-48, DOI: 10.25236/AJCIS.2023.060606

#### *4.2.2. Ablation study*

*Table 3: Ablation Experiments.*

| Method | Yoochoose1/64 | | Diginetica | | Nowplaying | |
|---|---|---|---|---|---|---|
| | P@20 | M@20 | P@20 | M@20 | P@20 | M@20 |
| w/o Category Info. | 72.03 | 32.86 | 23.70 | 8.81 | 55.89 | 19.55 |
| w/o Intent Node | 71.97 | 32.62 | 23.61 | 8.76 | 55.58 | 19.36 |
| w/o Global Intent | 72.07 | 32.76 | 23.62 | 8.82 | 55.72 | 19.43 |
| Random Init | 72.10 | 32.81 | 23.76 | 9.03 | 55.86 | 19.53 |
| CIGNN | **72.13** | **32.98** | **23.82** | **9.10** | **55.94** | **19.65** |

In this section, we conduct a series of ablation experiments to validate the proposed model design. The model incorporates category information for each item and establishes global intent nodes for both the intent node and the session based on the category information. To evaluate the impact of these added nodes on model performance, we designed several variants: 1) Initialization of the intent nodes using random initialization (Random Init), 2) Removal of category information (w/o Category Info.), 3) Removal of the introduced intent nodes (w/o Intent Node), and 4) Removal of the introduced session's global intent node (w/o Global Intent). The performance of these variants is compared to the original CIGNN model and reported in Table 3.

According to the data in Table 3, it can be observed that removing the global intent node from the model leads to a decrease in performance on both metrics across all three datasets. Furthermore, the model's performance declines even further after removing the intent nodes, indicating that the intent nodes and global intent node play a crucial role in determining the model's effectiveness. The results in the table also reveal a connection between the category of clicked items and user intent, as randomly initializing the intent nodes in the model and omitting category information both result in reduced performance on both metrics. Additionally, the performance decline on the Nowplaying dataset is not as pronounced when category information is removed. This may be due to the fact that the number of categories in the Nowplaying dataset is much larger than in the other datasets, making it difficult for the category representations to be fully trained.

#### *4.2.3. Impacts of Hyper-parameters*

In the model, a hyperparameter $K$ is introduced to control the number of user intents. To investigate its impact, we report the model's performance on the Yoochoose1/64, Diginetica, and Nowplaying datasets for a representative set of $K$ values $\{0,15,20,25,30,50,100\}$, and show the result in Figure 3.

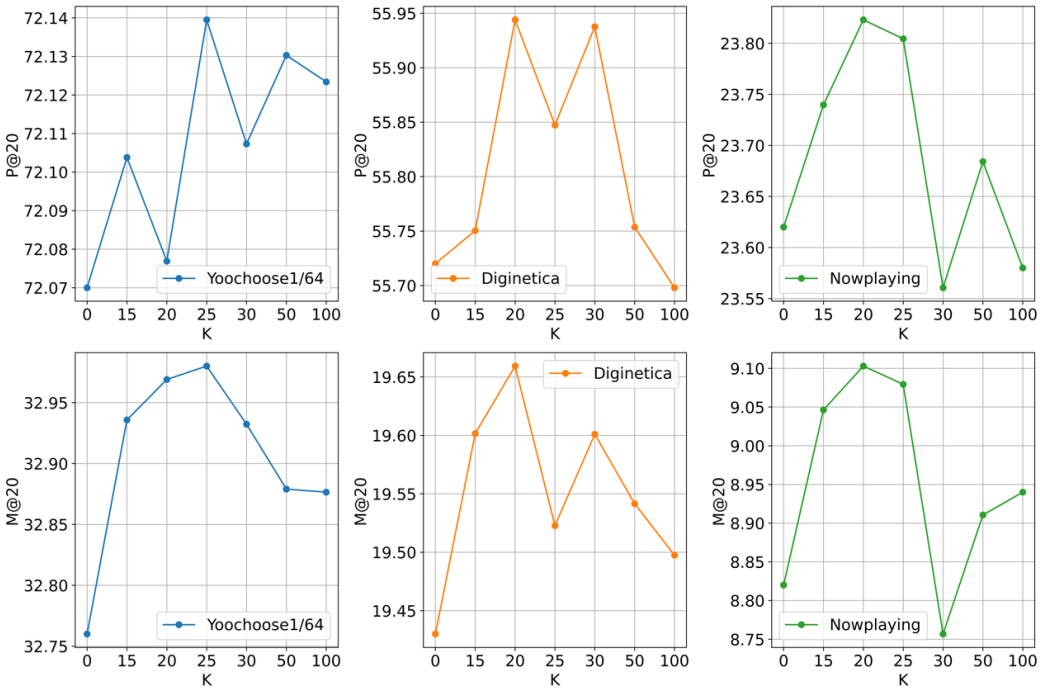


*Figure 3: Impact of the number of K.*

According to the results shown in Figure 3, both metrics generally exhibit an initial increase followed

ISSN 2616-5775 Vol. 6, Issue 6: 37-48, DOI: 10.25236/AJCIS.2023.060606

by a decrease as the value of $K$ increases. When the number of intent nodes is small, the model may not adequately capture users' diverse interests, resulting in lower recommendation accuracy. As the number of intent nodes increases, the model is better able to capture users' diversified interests, thus improving recommendation performance. However, when there are too many intent nodes, the model may face overfitting issues. Excessive intent nodes may cause the model to be overly sensitive to noise and special cases in the training data, neglecting the general patterns of user interests. This leads to a decline in model performance on test data. Furthermore, although there are significant differences in the number of categories across the three datasets, the optimal total number of user intents still falls within a relatively similar range. This suggests that the total number of user intents is stable and does not exhibit a proportional increase with the number of items.

We follow the similar approach to investigate the experimental impact of the hyperparameter $h$, which is set in the model to control the number of intents per item in the session. Figure 4 reports the performance of the model on the three datasets with a representative set of $h$ values. Since the best hyperparameter $K$ for the Yoochoose1/64 dataset is 25, the choice of $h$ values is $\{0,1,3,5,9,15,20,25\}$, while for the other datasets, the best hyperparameter $K$ is 20, so the choice of $h$ values for the Diginetica and Nowplaying datasets is $\{0,1,3,5,9,15,20\}$.

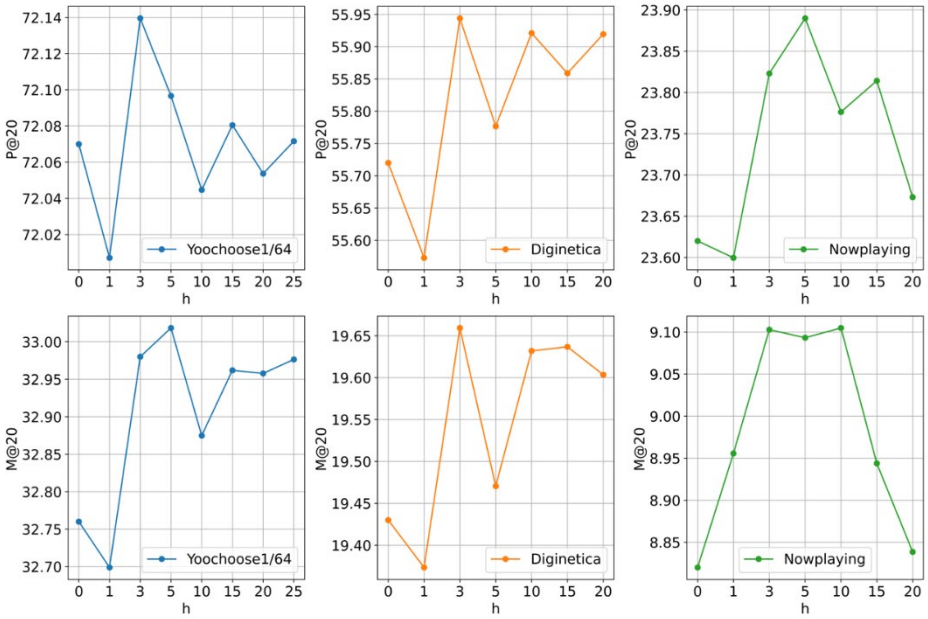


*Figure 4: Impact of the number of h.*

From the results in the Figure 4, it can be seen that setting the $h$ value to 1, assuming that users have only one intent when clicking on each item, may impair the model's performance. As the $h$ value in the model increases, both metrics typically show a trend of increasing first and then decreasing. This may be because when the model allocates fewer intents for each item, it cannot sufficiently capture the diverse intents of users, resulting in poorer model performance. As the $h$ value increases, the model can allocate more primary intents for each item, which helps the model better capture users' diverse interests and thereby improve recommendation performance. When $h$ increases further, the model is influenced by secondary intents and cannot accurately model users' primary interests, leading to a decline in model performance. An appropriate setting of $h$ value allows the model to focus more on users' primary interests, making the modeling of user intents more accurate.

### *4.2.4. Efficiency*

In this section, we evaluate the training efficiency of CIGNN. To ensure fairness in comparison, we set the batch and hidden sizes to 100 for all methods, including SRGNN, SGNN-HN and STAR. All experiments are performed on a single Nvidia RTX A4000 GPU and the same computing environment. All methods train for 10 epochs, and we report the average training time per epoch in Table 4. From Table 4, we can observe that CIGNN performs worse than other methods on the Nowplaying dataset, but our model's time is similar to other methods on the other two datasets. Our method uses the category information of items and builds more nodes, and its time consumption is similar to SRGNN and SGNN-HN. Considering the performance improvement, the difference in CIGNN training time is acceptable.

ISSN 2616-5775 Vol. 6, Issue 6: 37-48, DOI: 10.25236/AJCIS.2023.060606

*Table 4: Performances of average training time(s) per epoch.*

| Method | Yoochoose1/64 | Diginetica | Nowplaying |
|---|---|---|---|
| SRGNN | 564 | 606 | 516 |
| NISER+ | 573 | 717 | 727 |
| SGNNHN | 230 | 559 | 625 |
| CIGNN | 285 | 776 | 920 |

## 5. Conclusion

In this study, we explored the relationship between user intent and items in graph-based session recommendation methods. Existing graph-based recommendation methods seldom model the intertwined relationship between user intent and items, ignoring users' multiple intents when clicking on items. To address this gap, we focused on the close relationship between the category of the item clicked and the user's intention. We built a session graph with intent nodes and category nodes, and designed a simple intent-category fused graph neural network (CIGNN) to solve the problem. We adopt a simple initialization method, reducing the dimensionality of all item categories to obtain the representation of user intent nodes and adding edges between these intent nodes and item nodes. We use the attention mechanism to select the most critical multiple intentions for each item in the session, generating a unique mixed intent. It allows items with the same intent to propagate information through the edges between intent nodes and items, even if they are not adjacent. Furthermore, we fuse the information of mixed intentions to create a global intention node of the session, improving the model's understanding of the relationship between local and global intentions. The experimental results demonstrate the superiority of our proposed model.

## References

*[1] Zimdars A, Chickering D M, Meek C. Using temporal data for making recommendations[J]. arXiv preprint arXiv:1301.2320, 2013.*
*[2] Hidasi B, Karatzoglou A, Baltrunas L, et al. Session-based recommendations with recurrent neural networks[J]. arXiv preprint arXiv:1511.06939, 2015.*
*[3] Jannach D, Ludewig M. When recurrent neural networks meet the neighborhood for session-based recommendation[C]// Proceedings of the eleventh ACM conference on recommender systems. 2017: 306-310.*
*[4] Li J, Ren P, Chen Z, et al. Neural attentive session-based recommendation[C]// Proceedings of the 2017 ACM on Conference on Information and Knowledge Management. 2017: 1419-1428.*
*[5] Wu S, Tang Y, Zhu Y, et al. Session-based recommendation with graph neural networks[C]// Proceedings of the AAAI conference on artificial intelligence. 2019: 346-353.*
*[6] Pan Z, Cai F, Chen W, et al. Star graph neural networks for session-based recommendation[C]// Proceedings of the 29th ACM international conference on information & knowledge management. 2020: 1195-1204.*
*[7] Chen T, Wong R C W. Handling information loss of graph neural networks for session-based recommendation[C]// Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2020: 1172-1180.*
*[8] Wang S, Cao L, Wang Y, et al. A survey on session-based recommender systems[J]. ACM Computing Surveys, 2021, 54(7): 1-38.*
*[9] Rendle S, Freudenthaler C, Schmidt-Thieme L. Factorizing personalized markov chains for next-basket recommendation[C]// Proceedings of the 19th international conference on World wide web. 2010: 811-820.*
*[10] Wu X, Liu Q, Chen E, et al. Personalized next-song recommendation in online karaokes[C]// Proceedings of the 7th ACM Conference on Recommender Systems. 2013: 137-140.*
*[11] Hidasi B, Karatzoglou A. Recurrent neural networks with top-k gains for session-based recommendations[C]// Proceedings of the 27th ACM international conference on information and knowledge management. 2018: 843-852.*
*[12] Chen J, Li K, Li K, et al. Dynamic planning of bicycle stations in dockless public bicycle-sharing system using gated graph neural network[J]. ACM Transactions on Intelligent Systems and Technology, 2021, 12(2): 1-22.*
*[13] Zheng C, Fan X, Wang C, et al. Gman: A graph multi-attention network for traffic prediction[C]// Proceedings of the AAAI conference on artificial intelligence. 2020: 1234-1241.*

ISSN 2616-5775 Vol. 6, Issue 6: 37-48, DOI: 10.25236/AJCIS.2023.060606

*[14] Qiu R, Li J, Huang Z, et al. Rethinking the item order in session-based recommendation with graph neural networks[C]// Proceedings of the 28th ACM international conference on information and knowledge management. 2019: 579-588.*
*[15] Xu C, Zhao P, Liu Y, et al. Graph Contextualized Self-Attention Network for Session-based Recommendation [C]// Proceedings of the 28th International Joint Conference on Artificial Intelligence. 2019: 3940-3946.*
*[16] Guo L, Tang L, Chen T, et al. DA-GCN: A Domain-aware Attentive Graph Convolution Network for Shared-account Cross-domain Sequential Recommendation[C]// Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence. 2021: 2483–2489.*
*[17] Albadvi A, Shahbazi M. A hybrid recommendation technique based on product category attributes [J]. Expert Systems with Applications, 2009, 36(9): 11480-11488.*
*[18] He J, Li X, Liao L. Category-aware next point-of-interest recommendation via listwise bayesian personalized ranking[C]// Proceedings of the 28th International Joint Conference on Artificial Intelligence. 2017: 1837-1843.*
*[19] Cai R, Wu J, San A, et al. Category-aware collaborative sequential recommendation[C]// Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval. 2021: 388-397.*
*[20] Cui C, Shen Q, Zhu S, et al. Intention Adaptive Graph Neural Network for Category-Aware Session-Based Recommendation[C]// Proceedings of the 27th International Conference on Database Systems for Advanced Applications. 2022: 150-165.*
*[21] Li C, Liu Z, Wu M, et al. Multi-interest network with dynamic routing for recommendation at Tmall[C]// Proceedings of the 28th ACM international conference on information and knowledge management. 2019: 2615-2623.*
*[22] Cen Y, Zhang J, Zou X, et al. Controllable multi-interest framework for recommendation[C]// Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2020: 2942-2951.*
*[23] Li H, Wang X, Zhang Z, et al. Intention-aware sequential recommendation with structured intent transition[J]. IEEE Transactions on Knowledge and Data Engineering, 2021, 34(11): 5403-5414.*
*[24] Liu Z, Li X, Fan Z, et al. Basket recommendation with multi-intent translation graph neural network[C]// Proceedings of the 8th IEEE International Conference on Big Data. 2020: 728-737.*
*[25] Ma J, Zhou C, Yang H, et al. Disentangled self-supervision in sequential recommenders[C]// Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2020: 483-491.*
*[26] Pan Z, Cai F, Ling Y, et al. An intent-guided collaborative machine for session-based recommendation[C]// Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval. 2020: 1833-1836.*
*[27] Wang S, Hu L, Wang Y, et al. Modeling multi-purpose sessions for next-item recommendations via mixture-channel purpose routing networks[C]// Proceedings of the 28th International Joint Conference on Artificial Intelligence. 2019: 3771–3777.*
*[28] He X, Deng K, Wang X, et al. Lightgcn: Simplifying and powering graph convolution network for recommendation[C]// Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval. 2020: 639-648.*
*[29] Sarwar B, Karypis G, Konstan J, et al. Item-based collaborative filtering recommendation algorithms [C]// Proceedings of the 10th international conference on World Wide Web. 2001: 285-295.*
*[30] Liu Q, Zeng Y, Mokhosi R, et al. STAMP: short-term attention/memory priority model for session-based recommendation[C]// Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining. 2018: 1831-1839.*
*[31] Gupta P, Garg D, Malhotra P, et al. NISER: Normalized item and session representations to handle popularity bias[J]. arXiv preprint arXiv:1909.04276, 2019.*
*[32] Shen Q, Zhu S, Pang Y, et al. Temporal aware multi-interest graph neural network for session-based recommendation[C]// Proceedings of The 14th Asian Conference on Machine Learning. 2023.*