IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 12 December 2022, accepted 3 January 2023, date of publication 9 January 2023, date of current version 25 January 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3235353

RESEARCH ARTICLE

A Graph Positional Attention Network for Session-Based Recommendation

LIYAN DONG[1,2], GUANGTONG ZHU[1], YUEQUN WANG[1], YONGLI LI[3], JIAYAO DUAN[4], AND MINGHUI SUN[1,2]

[1]College of Computer Science and Technology, Jilin University, Changchun 130012, China
[2]Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, Changchun 130012, China
[3]School of Information Science and Technology, Northeast Normal University, Changchun 130117, China
[4]College of Software Engineering, Jilin University, Changchun 130012, China

Corresponding author: Minghui Sun (smh@jlu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61872164, in part by the Jilin Provincial Science and Technology Development Program under Grant 20220201147GX, and in part by the Central University Basic Research Fund, Jilin University, under Grant 2022-JCXK-02.

ABSTRACT The main idea of a session-based recommendation system is to model the user's historical click sequence and then summarize user preferences and predict the items the user will interact with. The session recommendation model based on graph neural networks has attracted much attention in recent years because it can accurately obtain the local relationship between items. However, the traditional session recommendation model based on graph neural Networks lack the use of user's higher-order features or fail to address the impact of item position information on the current session, which are both critical to the recommendation system. In addition, some models proposed the position information while neglects the click frequency information. We propose a graph network recommendation model called GPAN based on position attention in response to the abovementioned problems. Specifically, we propose a novel high-low order session perceptron that uses the perceptron to model undirected and directed graphs separately to obtain high and low order item representations in a session. For position information, we designed a position layer to calculate independently. Finally, the user's short-term preference and long-term preference are aggregated to obtain the recommendation sequence. The results through a large number of experiments on three real datasets show that the performance of the proposed GPAN model is the best.

INDEX TERMS Graph neural network, session-based recommendations, attention network.

I. INTRODUCTION

The continuous advancement of science and technology has made problems such as information explosion and overload more severe. The recommendation system can solve the abovementioned issues effectively. However, the traditional recommendation method has some shortcomings, such as cold start, weak processing of sparse scoring matrix, and high model overhead, which are generally performed in the recommendation of new users. More auxiliary information needs to be added in the recommendation process to improve the accuracy of the recommendation for solving the aforementioned problems. For example, [1], [2], [3], [4], [5], [6] proposes adding session data to the recommendation system. The main idea is to predict the item to be clicked by the next user based on the historical click sequence of anonymous users and continuously update the embedded representation of the item according to the user's interactive behavior. Thus, the cold start problem is greatly alleviated. For example, recommendation based on Markov chain [7], [8], [9], [10], [11] assumes that the past and future information are independent under known conditions. However, this assumption will have side effects in many scenarios. The recurrent neural network [12], [13], which is added to the gated neural unit, greatly improves the recommendation system. However, it ignores the deeper items and the click behavior between items.

Recently, self-attention has achieved remarkable results in sequence modeling and has been widely used in deep

The associate editor coordinating the review of this manuscript and approving it for publication was Francisco J. Garcia-Penalvo.

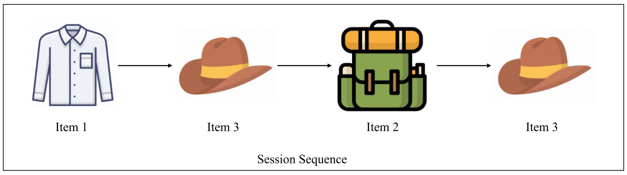


FIGURE 1. An instance of a session.

learning. Transformer [14] uses self-attention to form a codec to calculate the contextual relationship and becomes the best performing model in translation tasks. The application of transformer to language representation models [15] and computer vision models [16], [17], [18] have achieved great success. A main part of transformer is the self-attention network. This network is used to weight and aggregate the information of all items. It strengthens the influence of all nodes on the target, but it weakens the influence of local nodes on the target. Session-based graph convolution recommendation [19], [20], [21] once again pushes the recommendation model to a climax. It represents the session as a graph for the first time and injects it into a gated neural network, which has achieved remarkable results. However, it only aggregates successive and last items while ignoring the sequence information of the session clicks and the deeper item dependency information. Despite that existing Session-based graph convolution recommendation have made remarkable progress on SBRS, they mostly neglect position information and higher-order features. Take Figure 3 as an example,the user's click sequence is 1, 3, 2, 3. If the directed graph is used to predict, the next item that is most likely to interact is 2, but in fact, the next item that will interact may also be 1, because it starts from 1 to 3.

To address the above issues, we propose a Graph Positional Attention Network(GPAN) for session-based recommendation. Specifically, we first construct separate directed and undirected graphs based on user click sequences, and then use a new high-low order session perceptron to model the undirected and directed graphs separately to obtain high-order and low-order item representations in the session. Then, the result of the addition of high-order representation and low-order representation is aggregated with the position information. For position information, we propose a position layer to calculate.In detail,the position layer has two steps. First, calculate the occurrence frequency of each item in the session, and then multiply the sequence reverse representation and frequency to obtain the position representation. The local session representation is obtained by aggregating the position information after the addition of low-order and high-order, and then input this representation into the self-attention network to obtain the user's long-term preference representation. The combination of long-term preference and short-term preference forms the final preference representation. We conduct extensive experiments on three real-world datasets and the results show that the performance of the proposed GPAN model is the best.

The main contributions of this work are summarized as follows:

- In order to improve the accuracy of recommendation prediction, we propose a new graph based attention network model (GPAN). GPAN first integrates the high-level and low-level representations of the session, and then converges our proposed new position representation, which enables more accurate recommendations.
- We propose a novel perceptron for session high and low order items. The perceptron captures the high-order and low-order item transition relationships in a session separately, and uses these two relationships to accurately predict user preferences.
- A new session position information is proposed to highlight the relevance of sequence order to user preferences.
- We conduct extensive experiments on three benchmark datasets. Comprehensive analysis of our experimental results shows the effectiveness and superiority of GPAN compared with existing methods.

The contents of this article are as follows. The first section is introduction, the second section is related work, the third section is model introduction, the fourth section is experimental results and analysis, and the fifth section is future outlook.

II. RELATED WORK

A. TRADITIONAL RECOMMENDATION SYSTEM

The traditional recommendation can be divided into three categories: content-based recommendation, collaborative filtering-based recommendation, and hybrid model. The main method of the content-based recommendation model is to calculate the cosine similarity between items and make recommendations based on the similarity. The recommendation model based on collaborative filtering is currently the most widely studied in academia. The main method is to use groups of similar interests for recommending information that users are interested in and giving a certain degree of response to the information through group feedback. The collaborative filtering recommendation model focuses more on the historical interaction records of user items, that is, the user-commodity two-dimensional matrix, which is the main difference from content-based recommendation. Collaborative filtering is the earliest method, but it has not lost its competitiveness until now. An example is the POP model that recommends N hot items with the most interactions based on user historical behaviors. [22] proposes a new sorting technique for implicit feedback data by using Bayesian analysis to obtain the maximum posterior estimate for optimizing the ranking. The Item-KNN model proposed by [23] aims to calculate the item similarity matrix based on the user- commodity matrix and use the relationship between items to recommend. These algorithms are the enlightenment of recommendation algorithms and have a recommendation effect, but they do not perform well in sparse scoring matrix scenarios.

B. SESSION-BASED RECOMMENDATION SYSTEM

Content-based and collaborative filtering recommendations are two classic recommendation models. However, these traditional recommendation models cannot obtain information about users' short-term affairs, which leads to a great reduction in the prediction of users' short-term preferences. The session-based recommendation model effectively solves the abovementioned problems. The session is the user's recent affairs, such as the user's purchase of items and click order. The emergence of the session-based recommendation model has greatly promoted the development of the recommendation field, such as the original rule mining, which aims to use the association method to mine item relationships in the session. Subsequently, SKNN [24], [25] uses a session-based K neighbor recommendation model, which combines the session with the K neighbor algorithm, and recommends based on K sessions similar to the current session. SKNN considers the context information. In addition, a user-KNN approach built on users' session information is also proposed for next-basket recommendation [26]. MDPs [6] take the lead in applying Markov chains to sessions. Markov chains are used to model the interactions within and between sessions to predict the next user interaction. The method in [27] combines user-item matrix factorization technology with Markov chain, it first constructs a personalized transfer matrix based on Markov chain, then, it uses matrix decomposition model to solve the matrix Sparse problem. FPMC-LR [28] uses a factorization model based on FPMC to capture user personal preferences better. [11] proposes a combination of first and second order Markov models to provide more accurate recommendations. [10] proposes a personalized Markov model, which uses user item distance and inter-item distance to express the relationship. These algorithms are better than traditional recommendation models, but they ignore the conversion relationship between adjacent items in the session.

C. RECOMMENDATION SYSTEM BASED ON DEEP LEARNING

In recent years, a recommendation based on deep learning has become a main method. Google is the first to propose a video recommendation model based on YouTube deep learning [29]. The author divides the model into two parts: recall and sorting. Make accurate recommendations through these two parts. In the Kaggle competition, the wide and deep [30] model is proposed. The author divides the model into two parts: wide and deep. The wide part is a generalized linear model, which integrates all features. The deep part is a neural network model. The feature first enters the neural network to obtain the feature representation and then enters the wide part to obtain the predicted value. [31] proposes DeepFM, which is based on wide and deep, and adds a latent vector click method to obtain a second-order feature representation. Later, [32] proposes the deep cross model, which replaces the wide part in wide and deep with the cross part. Cross is mainly used for feature cross-coding and is no longer a generalized combination of all features before. [33] proposes to combine reinforcement learning with deep learning to predict user preferences. [34] proposes a neural network based on a factorization machine in 2016. This model uses a deep neural network to perform higher-order combinations of features and increases the learning ability of the model, but it only focuses on higher-order features. Low-level features are ignored, which results in limited model representation. These models integrate deep learning and recommendation effectively, which enables setting a model for later deep learning recommendation models.

D. RECOMMENDATION SYSTEM BASED ON SESSION AND DEEP LEARNING

In the upsurge of deep learning-based recommendation, a session-based deep learning recommendation model has been proposed. The GRU4Rec proposed by [12] first applies the recurrent neural network to the recommendation. The model has multilayer gating units to control the amount of information passed to the next layer in the neural network. [35] proposes embedding differential data to improve GRU4Rec. [36] proposes the NARM model, which integrates the attention mechanism into the GRU gating unit, assigns weights to each item to highlight the user's purpose, and aggregates the current session information global session information to obtain predictions. [37] proposes the STAMP model, which replaces the recurrent neural network with an attention network. Both works propose the combination of general interest and current interest, which emphasizes the importance of general interest. [19] proposes the SR-GNN model to represent the session as a graph, the session graph can strengthen the session information of adjacent items. The two conversational embeddings are combined to form the final conversational embedding. Although these models have greatly improved recommendation accuracy, they ignore the effect of the sequence of session items and the relevance of high-level items in the session on the recommendation. FGNN [20] adds multi-headed attention to learn that each item embeds and aggregates all items in the session. However, it does not highlight the conversion relationship between partial items. In our model GPAN, we use the new high-low order session perceptron to model the undirected and directed graphs separately to obtain high and low order item representations in the session, add position information to indicate the order of the items, and use the self-attention layer to learn the relationship between all items in the session. A linear combination of short-term and long-term preferences forms the final recommendation sequence.

III. METHOD

In this section, we introduce the Graph Positional Attention Network(GPAN) model. First, we describe the construction of the graph. Then, the composition of the model is described in detail.

A. SYMBOL DEFINITION

Let $V = \{v_1, v_2, v_3, \ldots, v_{|n|}\}$ represent the set of all items involved in the session, and n is the total number of items. Each session is defined by timestamp as $S = [V_{s,1}, V_{s,2}, V_{s,3}, \ldots, V_{s,m}]$, among them, $V_{s,i} \in V$, m is the session length. Our purpose is to predict the next item the user will click, $V_{s,m+1}$. Thus, we generate a probability sequence $\hat{y} = \{\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_{|V|}\}$, where each item in the sequence represents the probability that the corresponding item may click. Finally, we select the top N items in $\hat{y}$ to generate a candidate sequence to recommend to the user.

B. CONSTRUCTION OF GRAPH

Graph can more intuitively show the project interaction in the session and can highlight the transition between adjacent projects. Thus, we build the session into a graph. Therefore, each session can be represented as two graphs. As shown in Figure 3, we set each embedding space to the maximum length of the session and fill in the sessions that do not reach the full length with zeros. For directed graphs, we divide the interaction between nodes into four categories. The first category is self-connection. First, each node is self-connected. The purpose is to strengthen the influence of its information in the subsequent network. Self-connection is represented by 1. The second type is the out-degree connection, which means out-of-degree for the current node and is represented by 2. The third type is the in-degree connection, which means in-of-degree for the current node and is represented by 3. The fourth type is interconnection, which means that the current node interacts with other nodes and is represented by 4. If no interaction occurs, then it is represented by 0. For undirected graphs, we set the interactive project connection relationship to 1. We set the session representation space and node vector V, d is the dimension, and the session representation is composed of node vectors.

C. SESSION HIGH-LOW ORDER PERECPTRON

As shown in the gray rectangle in the left of Figure 2, The session graph contains information about the transformation of items in the current session, and in order to capture this relationship, in this model we propose a new high-low order session representation perceptron. We first input the directed and undirected graphs into the attention network separately to obtain the session high-order and low-order item transformation relations, and then fuse the two relations to obtain a session representation with both low-order and high-order item information.

$$f_{i,j} = \text{LeakyReLU}\left(A^T{}_{r_{ij}}\left(W_{v_i} \| W_{v_j}\right)\right) \tag{1}$$

$$f^*_{i,j} = \text{LeakyReLU}\left(A^{T^*}{}_{r^*_{ij}}\left(W'_{v_i} \| W'_{v_j}\right)\right) \tag{2}$$

where $A_{r_{ij}} \in \mathbb{R}^{2d}$, r_{ij} is the relationship between node i and node j. We use four different weight matrices to train different interaction relationships between nodes, namely A_{self}, A_{in} A_{out}, A_{inout}. They correspond to self-connection, in-degree connection, out-degree connection, and two-way connection. $A^*_{r^*_{ij}} \in \mathbb{R}^{2d}$, r^*_{ij} is the relationship between undirected graph nodes and A^* is the weight matrix of undirected graph. W is a linear transformation of shared parameters applied to all nodes of the directed graphs, $W \in \mathbb{R}^{d\times d}$. W' is the parameter matrix of undirected graph. Then, we enter the LeakyReLU activation function. LeakyReLU can assign all negative values to a non-zero slope. Thus, it can be backpropagated even for negative input values. Finally, the weights are normalized.

$$\beta_{ij} = \frac{\exp\left(f_{i,j}\right)}{\sum_{v_j \in N_{s_{v_i}}} \exp\left(\text{LeakyReLU}\left(e^T{}_{r_{ij}}\left(W_{v_i} \| W_{v_j}\right)\right)\right)} \tag{3}$$

$$\beta^*_{ij} = \frac{\exp\left(f^*_{i,j}\right)}{\sum_{v_j \in N_{s_{v_i}}} \exp\left(\text{LeakyReLU}\left(e^T{}^*_{r_{ij}}\left(W'_{v_i} \| W'_{v_j}\right)\right)\right)} \tag{4}$$

where, $N_{s_{v_i}}$ is the first-order neighbor of node v_j. Our purpose is to obtain the conversion information of adjacent items in the session. Thus, the first-order neighbor information is sufficient. After we obtain the weights of the neighbors of the nodes, we perform weighted multiplication. The purpose is to aggregate the information of each neighbor node. The calculation formula is as follows.

$$h^*_{v_i} = \sum_{v_j \in N_{s_{v_i}}} \beta_{\text{ij}} W_{v_j} + \sum_{v_j \in N_{s_{v_i}}} \beta^*_{\text{ij}} \, W'_{v_j} \tag{5}$$

Through the attention network, we get representations that combine the lower and higher order information of a session.

D. POSITION LAYER

As shown in the gray part in the middle of Figure 2, SR-GNN and many previous models have proven the importance of the item clicked last in the session. The importance of the item in the session changes with the time stamp, which is also in line with common sense. The previous method treats the last interactive item of the learned session as short-term interest. However, we adopt the method of splicing the item sequence information of the learned session to not only aggregate the information of the timestamp but also highlight the short-term interest. Therefore, we propose a new position representation that incorporates not only the position information of the items in the session, but also the frequency of the items in the session. In this model, our position information location information is the reverse learning project location embedding multiplied by the frequency of occurrence of the project. We set the position matrix $R_l = [P_1, P_2, P_3, \ldots, P_m]$,where m is the length of the current session, and l is the number of sessions. The session representation $S = [h_{s,1}, h_{s,2}, h_{s,3}, \ldots, h_{s,m}]$, and the fusion position embedding representation of the i-th item is

$$p_t = \tanh\left(W_1\left[t_i * P_{m-i+1} \| h^*_{v_i}\right] + b_1\right) \tag{6}$$

where $W_1 \in \mathbb{R}^{d\times 2d}$, $b_1 \in \mathbb{R}^d$, W_1 is the position weight matrix, and b_1 is the bias term. t_i is the frequency with

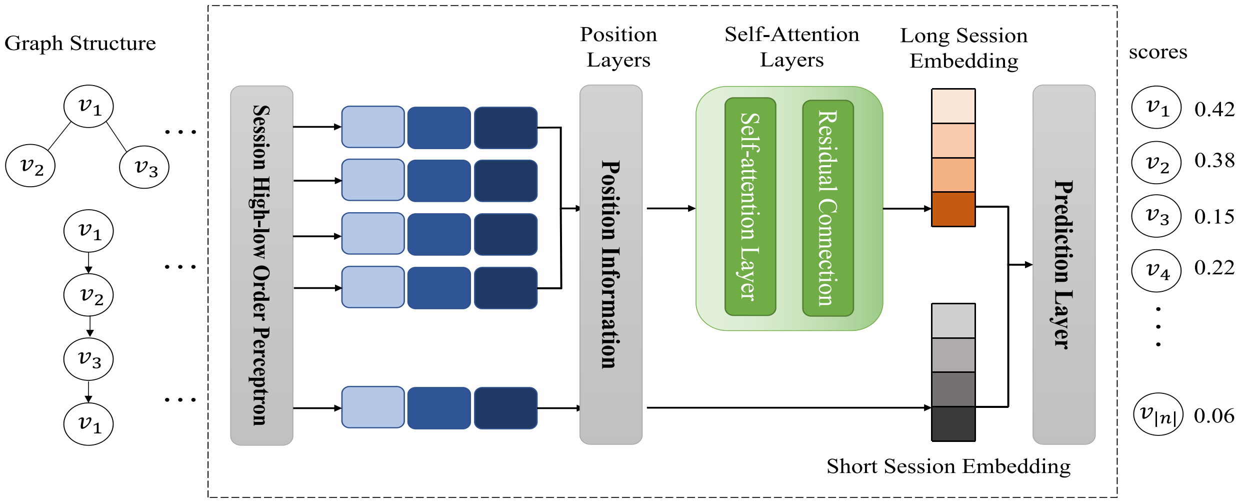
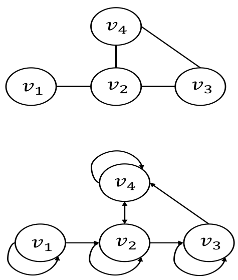


FIGURE 2. We first create a directed graph and an undirected graph based on the session. Then, we pass the high-low order session perceptron to obtain the embedding of each node in the current session, merge the position embedding of the node to obtain the short-term preference, and pass it to the self-attention layer to obtain the long-term preference. The two preferences are combined to finally obtain the probability of the next click on the item.



0	1	0	0
1	0	1	1
0	1	0	1
0	1	1	0

1	2	0	0
3	1	2	4
0	3	1	2
0	4	3	1

FIGURE 3. An instance of a session and the corresponding connection matrix.

which the i-th item appears in the current session. The weight matrix is integrated in the embedded matrix of the position information to ensure that the position information can obtain an appropriate weight. Given that the position information is only auxiliary information, our real focus is on the session representation.

$$H_o^* = \frac{1}{m}\sum_{t=1}^{m} h_m \tag{7}$$

$$z_t = W_4^T \sigma\left(W_2 H_o^* + W_3 p_t + c\right) \tag{8}$$

$$S^* = z_t \, h_{v_i}^* \tag{9}$$

where $W_4^T \in \mathbb{R}^{d\times 2d}$, $W_2 \in \mathbb{R}^{d\times d}$, $W_3 \in \mathbb{R}^{d\times d}$, $c \in \mathbb{R}^d$. W_2, W_3, W_4 are weight matrices, and c is a bias term, which are all trainable parameters. H_o^* is the average value of all items in the current session. We multiply the matrix after adjusting the weight of the position information with the corresponding session representation. The obtained S^* is the local preference embedding, and then S^* is passed to the self-attention layer.

E. SELF-ATTENTION LAYER

Self-attention has been widely used in various fields, and NLP obtains the greatest success [15]. Self-attention is a variant of the attention mechanism, and it reduces the dependence on external information and can better capture the internal correlation of features. In this model, we use self-attention to aggregate all item embeddings in the session to obtain long-term preferences. In the previous step, we have obtained the session representation combined with the position embedding and then aggregated it into the final session representation through a self-attention layer, as shown in the green rectangle in the middle of Figure 2. The specific approach is as follows:

$$F = \mathrm{softmax}\left(\frac{\left(S^* W^Q\right)\left(S^* W^K\right)^T}{\sqrt{d}}\right)\left(S^* W^V\right) \tag{10}$$

where $W^Q, W^K, W^V \in \mathbb{R}^{2d\times d}$, and they are all linear transformation matrices in self-attention and are all derived from S^*. Self-attention is divided into two processes. The first process is to calculate the weight coefficient based on W^Q and W^K, and the second process is to sum W^V based on the weight coefficient.

We propose to use a new residual connection to ensure stability of the attention network. During training, the ReLU activation is performed first. Then, the linear connection is performed. We use a two-layer residual link, which can retain more information from the previous layer and reduce training loss. We use a new residual connection to stabilize the self-attention layer and reduce loss.

$$C' = \mathrm{ReLU}\left(\mathrm{ReLU}(F) W_5 + b_2\right) W_6 + b_3 \tag{11}$$

$$C^* = \mathrm{Dropout}\left(C'\right) \tag{12}$$

where $W_5, W_6 \in \mathbb{R}^{d\times d}$, b_2 and b_3 are bias terms. To prevent over-fitting, each linear transformation is added with a

Dropout layer. We make a linear combination of the long-term conversational embedding and the short-term conversational embedding to obtain the final conversational embedding.

$$C = W_8 C^* + (E - W_8) S^* \tag{13}$$

where, $W_8 \in \mathbb{R}^{d\times d}$, E is the unit matrix of shape $d \times d$. In the end, we obtain the final conversational embedding that contains short- and long-term embeddings. In summary, we summarize the location information and attention network as

$$C = \text{PAN}\left(S^*\right) \tag{14}$$

$$C^K = \text{PAN}\left(C^{k-1}\right) \tag{15}$$

where, multi-layer self-attention C^K can obtain higher-level project complex relationships and ensure stability of the network. We set a layer of self-attention to C. In the experiment, we set the number of self-attention layers K to 4. The later part of this work will explore the influence of K on the experimental results.

F. PREDICTION LAYER

We use the final session embedding to predict the next item the user will click, as shown in the gray part on the right side of Figure 2.

$$\hat{y}_i = \text{soft}\max\left(C^T v_i\right) \tag{16}$$

Here, $\hat{y}_i$ is the predicted probability of the next item clicked by the user. We use cross-entropy as the loss function, and the formula is as follows:

$$L = -\sum_{i=1}^{n} y_i \log\left(\hat{y}_i\right) + (1 - y_i)\log\left(1 - \hat{y}_i\right) \tag{17}$$

where, y is the representation of the real term, L is the value of the loss function. We obtain the smallest loss through Adam's optimization.

IV. EXPERIMENTS

In this part, we first introduce the experimental dataset and the evaluation index. Finally we briefly introduce the baseline models used for comparison in this paper.

A. DATASETS

We conduct experiments on three real-world datasets, namely Diginetica, Nowplaying and Tmall.we preprocess each dataset as follows:

- The session of length 1 is deleted.
- Items with less than five occurrences are deleted.
- We set the sessions of the following week as the test set of Yoochoose and the sessions of the following few weeks as the test set of Diginetica.
- We cut the session and expressed it in a sequence-tag format.

For session $\left[V_{s,1}, V_{s,2}, \ldots, V_{s,m}\right]$, the sequence we generate is $\left[\left[V_{s,1}\right], \left[V_{s,1},\ V_{s,2}\right] \ldots, \left[V_{s,1},\ V_{s,2}, \ldots, V_{s,m-1}\right]\right]$, and the corresponding labels are $\left[\left[V_{s,2}\right], \left[V_{s,3}\right], \ldots, \left[V_{s,m}\right]\right]$. After the abovementioned practices, the basic data we finally obtained are shown in Table 1.

TABLE 1. Statistics of datasets.

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

B. RESEARCH QUESTIONS AND EVALUATION INDEX

We list the following research questions to guide our experiments:

- **RQ1.** How do our proposed GPAN model perform in session-based recommendation task compared with the state-of-the-art baselines?
- **RQ2.** Does higher-order information about the session help improve model performance?
- **RQ3.** Does position information of items in a session help improve model performance?
- **RQ4.** How do the key hyper-parameter K affect model performance?

To assess the effectiveness of the model, we use two broad indicators: **P@N** and **MMR@N**. **P@N** is the accuracy, which means that the correct ratio is recommended among the first N recommended items.

$$\text{P} = \frac{\sum_{u\in U} |R(u) \cap T(u)|}{\sum_{u\in U} |R(u)|} \tag{18}$$

Here, $R(u)$ is a list of recommendations made to the user based on the user's behavior on the training set, and $T(u)$ is a list of the user's behavior on the test set.

MMR@N is the average reciprocal level related to the order of the first N recommended items.

$$\text{MRR} = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{\text{rank}_i} \tag{19}$$

Here, Q is the number of correct recommendations, and represents the correct ranking of the i-th recommendation. In each dataset, we set N to 10 and 20, that is, the results of recommending TOP-10 and TOP-20 to users are compared.

C. BASELINE MODELS AND PARAMETER SETTINGS

We compare with the following eight previous models.

- POP: It recommends the N most popular items to users; it has the effect of recommendation but has low accuracy.
- Item-KNN [23]: It recommends similar items that users have clicked before, and it starts to be related to the user's historical information. It performs better in scenarios where certain items are closely connected.
- FPMC [27]: It combines Markov chain and matrix factorization technology to obtain user preferences. The

author establishes a decomposition model to solve the problem of data sparseness. In the case of sparse user ratings, the model performs well.

- GRU4Rec [12]: It uses a recurrent neural network with a gated neural unit GRU to obtain user preferences. The author finds that GRU is sufficient to encode item information, and a single layer is fine.
- NARM [36]: It uses a recurrent neural network with an attention mechanism to obtain user preferences. The author proposes that each item representation in the session should not be treated equally. Weight should be trained for each item representation to aggregate into a session representation.
- STAMP [37]: On the basis of the cyclic neural network of the attention mechanism, the last session clicking on my item is regarded as a short- term preference, and the long- and short-term preferences are combined to express user preferences. The importance of short-term sessions is proposed for the first time and proven in the paper.
- SR-GNN [19]: It constructs the session into a graph and uses a gated neural network to obtain user preferences. The last clicked item is regarded as the user's short-term preference, and the two are combined to predict the user's next clicked item. The graph can better obtain the conversion relationship between items, and this feature improves the accuracy of recommendations.
- FGNN [20]: It uses a weighted attention network to aggregate project neighbors, highlights the local conversion relationship, and obtains a preference expression closer to the user.

We set the hyperparameters of each model to be the same, the batch is 100, the latent vector dimension is 100, the validation set is a one-tenth random subset of the training set, and the parameters are all Gaussian distributions with a mean of 0 and a standard deviation of 0 initialization. For FGNN, we follow the setting of the original paper and set its weighted attention network to three layers. For our model, the number of attention network layers is 1, the self-attention head is 4, the other initial values are 0, the initial learning rate is 0.001, the decay rate is 0.1, and the decay is once every three batches by using the Adam optimizer.

V. EXPERIMENTAL ANALYSIS

A. RESULT ANALYSIS

To answer RQ1, we use four indicators to compare each model on three real datasets. As shown in Table 2, we present the highest model in each index in bold, analyze the experimental results, and obtain the following conclusions.

In the traditional model, POP performs poorly because the items that focus only on the most frequent occurrences in a session are too limited. Item-KNN, as a classic algorithm, has the best performance. The reason is that Item-KNN regards the relationship between items as a recommended key step, and the session items in Diginetica, Nowplaying and Tmall datasets are closely connected. Therefore, Item-KNN can achieve better recommendation effect. Recommendation models based on deep learning proposed in recent years (GRU4REC, NARM, STAMP, SR-GNN, FGNN and our proposed GPAN) have greatly improved compared with traditional models, which also proves that deep learning in the field of recommendation indeed plays a key role. The deep learning model is divided into the model based on the graph neural network structure and the model based on the recurrent neural network structure. Models based on the recurrent neural network structure include NARM and STAMP. GRU4REC applies the recursive unit (GRU) to the network. NARM designs a two-way multilayer recursive unit (GRU) and attention mechanism to integrate the user's conversational behavior and the user's main purpose. On the basis of NARM, STAMP regards the last clicked item in the conversation as a short-term preference and integrates it into the conversational expression. This method has been excellently promoted, which also proves the importance of short-term preference. NARM and STAMP are superior to GRU4REC, which demonstrates that obtaining the user's order behavior is insufficient, and the attention mechanism must be added to assign weights to each item for aggregation. The effect of the model based on the graph neural network structure is better than that of the model based on the recurrent neural network structure because the graph can better reflect the relationship between the items in the session. SR-GNN constructs the sequence into a graph and inputs it into the gated neural network to obtain the long- term preference representation. The last item clicked in the session is the short-term preference representation, and the two are combined to form the final preference. FGNN replaces the gated neural network with an attention network based on SR-GNN, the result is a partial improvement, which proves that the attention network can obtain the session representation better than the gated neural network added to the GRU in some scenarios. However, the above model only focuses on current session lower-order items and ignores the impact of higher-order items. At the same time, the frequency of items appearing in the session is also ignored. To solve the above problem, we propose a session-based position attention graph neural network recommendation (GPAN) model. Compared with SR-GNN and FGNN, our advantages are as follows:

- Our proposed high-low order session perceptron can obtain the conversion relation of high order items in a session and the conversion relation of low order items respectively, and using these two relations we can fully explore the user preferences.
- After the session representation are constructed, we add the position information of each item in the session. Through the position information, the recently interacted items are given more weight, which highlights the user's recent preferences. FGNN does not add position information. Thus, it does not perform effectively on some datasets such as Nowplaying.

TABLE 2. Performance of all methods on three datasets.

Dataset	Diginetica				Nowplaying				Tmall			
Methods	P@10	P@20	MRR@10	MRR@20	P@10	P@20	MRR@10	MRR@20	P@10	P@20	MRR@10	MRR@20
POP	0.76	1.18	0.26	0.28	1.86	2.28	0.83	0.86	1.67	2.00	0.88	0.90
Item-KNN	25.07	35.77	10.77	11.57	10.96	15.94	4.55	4.91	6.65	9.15	3.11	3.31
FPMC	15.43	26.53	6.20	6.95	5.28	7.36	2.68	2.82	13.10	16.06	7.12	7.32
GRU4REC	17.93	29.45	7.33	8.33	6.74	7.92	4.40	4.48	9.47	10.93	5.78	5.89
NARM	35.44	49.70	15.13	16.17	13.60	18.59	6.62	6.93	19.17	23.30	10.42	10.70
STAMP	33.98	45.64	14.26	14.32	13.22	17.66	6.57	6.88	22.63	26.47	13.12	13.36
SR-GNN	36.86	50.73	15.52	17.59	14.17	18.87	7.15	7.47	23.41	27.57	13.45	13.72
FGNN	37.72	50.58	15.95	16.84	13.89	18.78	6.80	7.15	20.67	25.24	10.07	10.39
GPAN	**40.82**	**53.96**	**17.94**	**18.84**	**17.14**	**22.64**	**7.23**	**7.66**	**23.78**	**28.27**	**13.57**	**13.86**

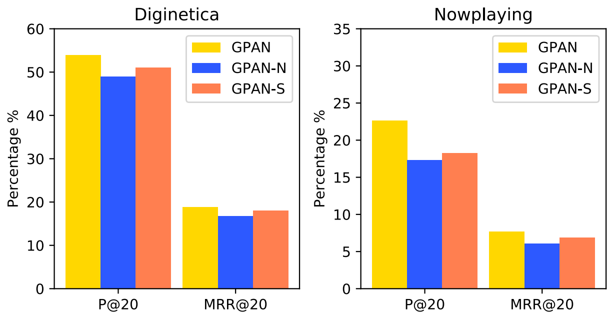


FIGURE 4. Performance of different components.

In the three datasets, the four indicators of GPAN are higher than other baseline models.

B. ABLATION EXPERIMENT

To answer RQ2 and RQ3, we conduct further research. The first experiment is the influence of position information and higher-order information on experiment results. The second experiment is the influence of the selection of self-attention K on the results. We compare GPAN with its different variants, The variant models include:

- GPAN-N: GPAN without the position layer.
- GPAN-S: GPAN without the higher-order information and only applies directed graph to model the current session.

The performance of different variants is presented in Figure 4. The x-axis of the graph is the evaluation index (P@20,MRR@20) and the y-axis is the percentage (percentage %). We can draw the following conclusions:

- In the two datasets, the position information is beneficial to the recommendation. When the position information is not added, both indicators will decrease. This finding shows the importance of position information in the session representation.
- GPAN-N has no inter-session item position information and cannot highlight the short-term preferences of users. Thus, the recommendation effect is not ideal compared with that of the other two models, which once again proves the importance of short-term preferences.
- The performance of GPAN-S on the two datasets is not as good as that of GPAN, which proves that the conversion relationship of high-level items in the session is potentially affecting the recommendation effect.

TABLE 3. The impacts of K.

Dataset	Diginetica		Nowplaying	
Methods	P@20	MRR@20	P@20	MRR@20
1-K	53.85	18.79	21.23	7.26
2-K	53.89	18.76	22.12	7.51
3-K	53.94	18.82	22.53	**7.69**
4-K	**53.96**	**18.84**	**22.64**	7.66
5-K	53.94	18.80	22.15	7.48

C. THE INFLUENCE OF K ON THE EXPERIMENT

To answer RQ4, we explore the influence of K on the experiment. We choose the number of self-attention levels (K) in [1], [2], [3], [4], and [5], and the result is shown in Table 3. In the Diginetica dataset, the number of layers of the self-attention network has little effect on the P@20 indicator, and the MRR@20 indicator reaches its maximum after the fourth round. In the Nowplaying dataset, P@20 and MRR@20 achieve better results in the fourth and third layers, but the effect of the fifth layer is slightly reduced. This result may be due to that too many layers lead to greater losses. In this model, the number of self-attention layers is 4, that is, $K = 4$.

VI. CONCLUSION AND FUTURE WORK

The recommendation model combining deep learning and session has become a boom. In this context, we propose a session-based position attention network recommendation model (GPAN). First, we construct the session into directed and undirected graphs. We then use the attention network to aggregate the information of adjacent items to more accurately capture the session relationship between adjacent items. Through the attention network, we get high-order and low-order information about a session. Then, the session representation is fused with the position information to form the user's short-term preferences, and the position information can effectively highlight the importance of recent projects. Finally, the user's short-term preferences are input into our improved residual self-attention network to obtain the user's long-term preferences. The combination of long- and short-term preferences forms the user's final preference. In this way, the last preferences we obtain include not only short-term preferences with position information but also long- term preferences of users. Experiments on three datasets show that the model outperforms eight benchmark methods.

Our future work will focus on further improving the recommendation ability of GPAN to obtain user preference expressions in more complex scenarios for predicting users.

REFERENCES

[1] B. Mobasher, H. Dai, T. Luo, and M. Nakagawa, ''Effective personalization based on association rule discovery from web usage data,'' in *Proc. 3rd Int. Workshop Web Inf. Data Manage. (WIDM)*, 2001, pp. 9–15.

[2] R. Forsati, M. R. Meybodi, and A. G. Neiat, ''Web page personalization based on weighted association rules,'' in *Proc. Int. Conf. Electron. Comput. Technol.*, Feb. 2009, pp. 130–135.

[3] B. Shao, M. Ogihara, D. Wang, and T. Li, ''Music recommendation based on acoustic features and user access patterns,'' *IEEE Trans. Audio, Speech, Language Process.*, vol. 17, no. 8, pp. 1602–1611, Nov. 2009.

[4] M. N. Moreno, F. J. García, M. J. Polo, and V. F. López, ''Using association analysis of web data in recommender systems,'' in *Proc. Int. Conf. Electron. Commerce Web Technol.* Cham, Switzerland: Springer, 2004, pp. 11–20.

[5] U. Niranjan, R. Subramanyam, and V. Khanaa, ''Developing a web recommendation system based on closed sequential patterns,'' in *Proc. Int. Conf. Adv. Inf. Commun. Technol.* Cham, Switzerland: Springer, 2010, pp. 171–179.

[6] M. Eirinaki, M. Vazirgiannis, and D. Kapogiannis, ''Web path recommendations based on page ranking and Markov models,'' in *Proc. 7th Annu. ACM Int. Workshop Web Inf. Data Manage.*, Nov. 2005, pp. 2–9.

[7] G. Shani, D. Heckerman, R. I. Brafman, and C. Boutilier, ''An MDP-based recommender system,'' *J. Mach. Learn. Res.*, vol. 6, no. 9, pp. 1–31, 2005.

[8] S. Chen, J. L. Moore, D. Turnbull, and T. Joachims, ''Playlist prediction via metric embedding,'' in *Proc. 18th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2012, pp. 714–722.

[9] D.-T. Le, Y. Fang, and H. W. Lauw, ''Modeling sequential preferences with dynamic user and context factors,'' in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*. Cham, Switzerland: Springer, 2016, pp. 145–161.

[10] X. Wu, Q. Liu, E. Chen, L. He, J. Lv, C. Cao, and G. Hu, ''Personalized next-song recommendation in online Karaokes,'' in *Proc. 7th ACM Conf. Recommender Syst.*, Oct. 2013, pp. 137–140.

[11] Z. Zhang and O. Nasraoui, ''Efficient hybrid web recommendations based on Markov clickstream models and implicit search,'' in *Proc. IEEE/WIC/ACM Int. Conf. Web Intell. (WI)*, Nov. 2007, pp. 621–627.

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

[13] B. Hidasi and A. Karatzoglou, ''Recurrent neural networks with top-k gains for session-based recommendations,'' in *Proc. 27th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2018, pp. 843–852.

[14] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, ''Attention is all you need,'' in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–11.

[15] I. Sutskever, O. Vinyals, and Q. V. Le, ''Sequence to sequence learning with neural networks,'' in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 27, 2014, pp. 1–9.

[16] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, ''End-to-end object detection with transformers,'' in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 213–229.

[17] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, ''An image is worth 16×16 words: Transformers for image recognition at scale,'' 2020, *arXiv:2010.11929*.

[18] N. Parmar, A. Vaswani, J. Uszkoreit, L. Kaiser, N. Shazeer, A. Ku, and D. Tran, ''Image transformer,'' in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 4055–4064.

[19] S. Wu, Y. Tang, Y. Zhu, L. Wang, X. Xie, and T. Tan, ''Session-based recommendation with graph neural networks,'' in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 346–353.

[20] R. Qiu, J. Li, Z. Huang, and H. YIn, ''Rethinking the item order in session-based recommendation with graph neural networks,'' in *Proc. 28th ACM Int. Conf. Inf. Knowl. Manage.*, Nov. 2019, pp. 579–588.

[21] R. Qiu, Z. Huang, J. Li, and H. Yin, ''Exploiting cross-session information for session-based recommendation with graph neural networks,'' *ACM Trans. Inf. Syst.*, vol. 38, no. 3, pp. 1–23, Jul. 2020.

[22] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, ''BPR: Bayesian personalized ranking from implicit feedback,'' 2012, *arXiv:1205.2618*.

[23] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl, ''Item-based collaborative filtering recommendation algorithms,'' in *Proc. 10th Int. Conf. World Wide Web (WWW)*, 2001, pp. 285–295.

[24] M. Ludewig and D. Jannach, ''Evaluation of session-based recommendation algorithms,'' *User Model. User-Adapt. Interact.*, vol. 28, pp. 331–390, Dec. 2018.

[25] D. Jannach and M. Ludewig, ''When recurrent neural networks meet the neighborhood for session-based recommendation,'' in *Proc. 11th ACM Conf. Recommender Syst.*, Aug. 2017, pp. 306–310.

[26] H. Hu, X. He, J. Gao, and Z.-L. Zhang, ''Modeling personalized item frequency information for next-basket recommendation,'' in *Proc. 43rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2020, pp. 1071–1080.

[27] S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme, ''Factorizing personalized Markov chains for next-basket recommendation,'' in *Proc. 19th Int. Conf. World Wide Web (WWW)*, 2010, pp. 811–820.

[28] C. Cheng, H. Yang, M. R. Lyu, and I. King, ''Where you like to go next: Successive point-of-interest recommendation,'' in *Proc. 23rd Int. Joint Conf. Artif. Intell.*, 2013, pp. 1–7.

[29] P. Covington, J. Adams, and E. Sargin, ''Deep neural networks for Youtube recommendations,'' in *Proc. 10th ACM Conf. Recommender Syst.*, Sep. 2016, pp. 191–198.

[30] H.-T. Cheng, L. Koc, J. Harmsen, T. Shaked, T. Chandra, H. Aradhye, G. Anderson, G. Corrado, W. Chai, and M. Ispir, ''Wide & deep learning for recommender systems,'' in *Proc. 1st Workshop Deep Learn. Recommender Syst.*, 2016, pp. 7–10.

[31] H. Guo, R. Tang, Y. Ye, Z. Li, and X. He, ''DeepFM: A factorization-machine based neural network for CTR prediction,'' 2017, *arXiv:1703.04247*.

[32] R. Wang, B. Fu, G. Fu, and M. Wang, ''Deep & cross network for ad click predictions,'' in *Proc. ADKDD*, Aug. 2017, pp. 1–7.

[33] X. Zhao, L. Xia, L. Zhang, Z. Ding, D. Yin, and J. Tang, ''Deep reinforcement learning for page-wise recommendations,'' in *Proc. 12th ACM Conf. Recommender Syst.*, Sep. 2018, pp. 95–103.

[34] D. Liang, J. Altosaar, L. Charlin, and D. M. Blei, ''Factorization meets the item embedding: Regularizing matrix factorization with item co-occurrence,'' in *Proc. 10th ACM Conf. Recommender Syst.*, Sep. 2016, pp. 59–66.

[35] Y. K. Tan, X. Xu, and Y. Liu, ''Improved recurrent neural networks for session-based recommendations,'' in *Proc. 1st Workshop Deep Learn. Recommender Syst.*, Sep. 2016, pp. 17–22.

[36] J. Li, P. Ren, Z. Chen, Z. Ren, T. Lian, and J. Ma, ''Neural attentive session-based recommendation,'' in *Proc. ACM Conf. Inf. Knowl. Manage.*, Nov. 2017, pp. 1419–1428.

[37] Q. Liu, Y. Zeng, R. Mokhosi, and H. Zhang, ''STAMP: Short-term attention/memory priority model for session-based recommendation,'' in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2018, pp. 1831–1839.

LIYAN DONG received the Ph.D. degree from Jilin University, in 2007. He is currently a Professor with the College of Computer Science and Technology, Jilin University. His research interests include database theory, data mining, distributed database and application, embedded database, and information retrieval.

GUANGTONG ZHU received the B.S. degree in information and computing science from Shandong Jiaotong University, Shandong, China, in 2020. He is currently pursuing the master's degree in computer science and technology. His research interests include recommendations systems and information retrieval.

JIAYAO DUAN received the bachelor's degree in engineering from the Jiangxi University of Technology, Jiangxi, China, in 2020. She is currently pursuing the master's degree in software engineering. Her research interests include recommendations systems and information retrieval.

YUEQUN WANG received the Ph.D. degree from the School of Computer Science and Technology, Jilin University, in 2022. His research interests include social networking, deep learning, and artificial intelligence. He wants to combine the field of natural language processing with the field of social networks to solve problems related to social networks.

YONGLI LI was born in Harbin, China, in 1965. She received the Ph.D. degree from Jilin University, in 2010. She is currently an Associate Professor with the School of Information Science and Technology, Northeast Normal University. Her research interests include information security and data processing.

MINGHUI SUN received the Ph.D. degree in computer science from the Kochi University of Technology, Japan, in 2011. He is currently an Assistant Professor with the College of Computer Science and Technology, Jilin University, China. His research interests include using HCI methods to solve challenging real-world computing problems in many areas, including tactile interfaces, pen-based interfaces, and tangible interfaces.