



# Modeling Price-Aware Session-Based Recommendation Based on Graph Neural Network

Jian Feng\*, Yuwen Wang and Shaojian Chen

College of Computer Science & Technology, Xi'an University of Science and Technology, Xi'an, 710600, China

\*Corresponding Author: Jian Feng. Email: fengjian@xust.edu.cn

Received: 27 December 2022; Accepted: 17 April 2023; Published: 09 June 2023

**Abstract:** Session-based Recommendation (SBR) aims to accurately recommend a list of items to users based on anonymous historical session sequences. Existing methods for SBR suffer from several limitations: SBR based on Graph Neural Network often has information loss when constructing session graphs; Inadequate consideration is given to influencing factors, such as item price, and users' dynamic interest evolution is not taken into account. A new session recommendation model called Price-aware Session-based Recommendation (PASBR) is proposed to address these limitations. PASBR constructs session graphs by information lossless approaches to fully encode the original session information, then introduces item price as a new factor and models users' price tolerance for various items to influence users' preferences. In addition, PASBR proposes a new method to encode user intent at the item category level and tries to capture the dynamic interest of users over time. Finally, PASBR fuses the multi-perspective features to generate the global representation of users and make a prediction. Specifically, the intent, the short-term and long-term interests, and the dynamic interests of a user are combined. Experiments on two real-world datasets show that PASBR can outperform representative baselines for SBR.

**Keywords:** Session-based recommendation; graph neural network; price-aware; intention; dynamic interest

## 1 Introduction

Recommendation systems successfully alleviate information overload by recommending helpful content to users. Traditional recommendation methods (e.g., collaborative filtering) typically rely on user profiles' availability and long-term historical interactions. However, when the information is unavailable (e.g., non-logged-in users) or has limited availability (e.g., short-term historical interactions), it is challenging to capture users' preferences accurately. To meet the challenges, many works in Session-based Recommendation (SBR) have been done recently, which predict the next item in temporal order based on sessions composing anonymous behaviors [1].



This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

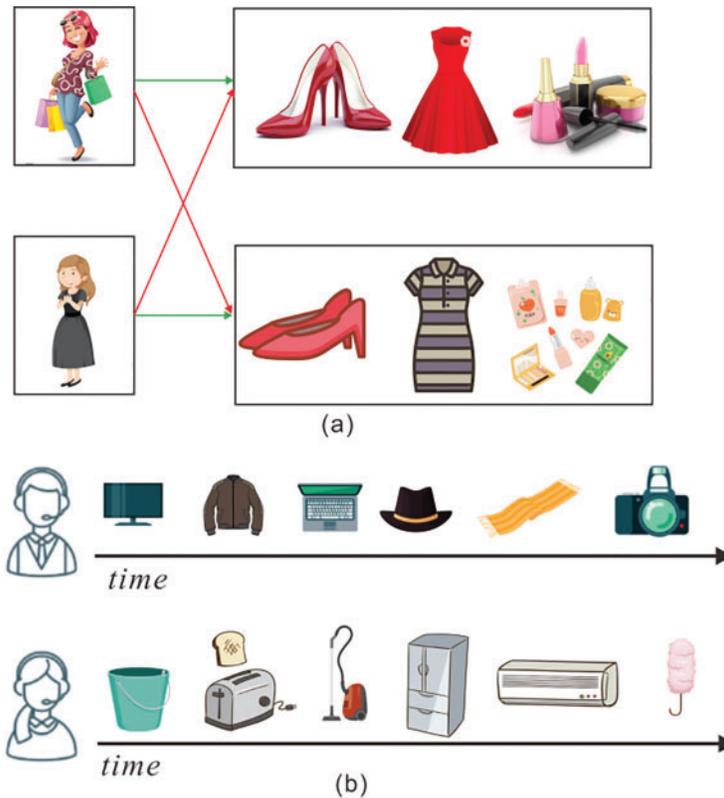
Methods for SBR can be divided into traditional and Deep Neural Network (DNN) methods, and the DNN-based methods are the mainstream of current research. Due to the sequential nature of sessions, Recurrent Neural Network (RNN)-based methods are first used for SBR [2,3] because they are suitable for processing sequential data in nature. However, RNN-based methods assume dependencies between adjacent items in sessions, and their representation of complex transfer patterns between items is inadequate. Consequently, Convolutional Neural Network (CNN)-based methods have been introduced for SBR [4–6]. CNN-based methods relax the order constraint of session data and can effectively capture the co-dependencies between non-adjacent items. However, like RNN-based methods, CNN-based methods are incapable of dealing with the association between multiple sessions neither, which can be solved to a certain extent by Graph Neural Network (GNN)-based methods [7–9], and GNN-based methods have become increasingly popular both in academic research and practical applications.

However, there are still some limitations in current research on SBR. Firstly, some important influencing factors are not considered enough. For example, price is often the decisive factor when a user buys an item. Fig. 1a illustrates the problem. In Fig. 1a, a fashion girl would pay good money on daily wear and cosmetics, while the plain girl always chooses to buy a cheaper alternative. The essence of this type of problem is that the users' price tolerance for an item is related to the category of the item. However, few SBR methods consider the influence of price factors [10]. So, how to model price tolerance for users? It is the first challenge. Secondly, users' interests often change significantly over time, while the affecting factors on interest evolution can't be directly observed from the session sequences. These make recommendations difficult. For example, in Fig. 1b, users' purchase interests are complex in both two session sequences and change significantly over time. So how to model the dynamic evolution of user interests? It is the second challenge. Finally, existing studies usually model users' intentions at the level of individual items. However, the information extracted from individual items lacks commonality, which may limit the expression of the users' intentions. Therefore, the third challenge is capturing user intent more accurately beyond individual items.

To tackle these challenges, an SBR method called Price-aware Session-based Recommendation (PASBR) is proposed, which models the users' intentions and interests based on GNN. To use GNN, PASBR constructs two kinds of lossless session graphs according to [9] to eliminate information loss caused by session graph construction. Then a price-aware layer is designed to model users' tolerance to price by incorporating the price and category of items. User intent is extracted from the category of items, and user interests in PASBR further consist of dynamic interest, short-term interest, and long-term interest. Among them, dynamic interest is modeled based on the representation of items, combined with a new time-evolving weight calculation function. Short-term interests are expressed in the category of the last interacted item, and long-term interests are incorporated to reduce the influence of false clicks on users' current intent. Finally, the learned intent and various interests are combined to make recommendations.

Our main contributions are as follows:

- A novel method PASBR is proposed to model price tolerance, user intent, and interests from new perspectives.
- To explore new factors that affect user preferences, a price tolerance factor is designed to model the price tolerance of users for various items.
- We conducted extensive empirical studies on two commonly used datasets and showed that PASBR outperforms presentational baselines.



**Figure 1:** Price preference and dynamic interest. (a) Shows the different price tolerances from different users for the same categories of goods, where the green line indicates high tolerance and the red line indicates low tolerance. (b) Shows users' interests changing over time

In the rest of this paper, Section 2 introduces related works, and Section 3 gives a detailed description of PASBR. Then the extensive experiments are analyzed in Section 4, and conclusions and future work are given in Section 5.

## 2 Related Works

Many SBR methods are proposed, including traditional and DNN methods.

### 2.1 Traditional Methods

Traditional SBR methods fall into pattern/rule mining-based methods and Markov chain-based methods. The former guides subsequent recommendations by mining association rules from interactions within a session [11–13]. However, these methods ignore the order of items within a session and make predictions mainly according to the last click. The latter takes the sequence properties of the session into account. For example, Factorizing Personalized Markov Chains for Next-Basket Recommendation (FPMC) captures both sequential patterns and the users' long-term preferences by combining matrix decomposition and first-order Markov chains [14]; Session-based K-Nearest-Neighbors approach (SKNN) proposed a hidden Markov probabilistic model by introducing other information, such as contextual features in [15]. The disadvantage of Markov chain-based methods is

their strong assumption of conditional dependence on the transfer of two adjacent terms, which limits prediction accuracy.

These traditional approaches are shallow models and can't capture complex features and dependencies.

## 2.2 DNN Methods

With the development of deep learning, DNN methods have been widely applied to SBR. Due to the sequential nature of session data, it is natural for RNN to be used first. For example, GRU4Rec adopts Gated Recurrent Unit (GRU) to model the session sequence [2]. Such methods usually capture the transfer patterns of the session but ignore users' potential purposes and preference characteristics. To solve these problems, Neural Attentive Session-based Recommendation (NARM) fused attention mechanisms into a stacked GRU to capture representative transfer information of items while modeling users' primary purposes [3]. Since then, attention mechanisms have been increasingly used to capture users' long-term and short-term preferences [16–18].

However, RNN-based methods are incapable of modeling complex dependencies between non-adjacent items. Therefore, CNN-based methods have received a lot of attention. They relax the assumption of strict order among items and can effectively capture co-dependencies between non-adjacent items. For example, Ref. [6] proposed a convolutional sequence modeling method that constructs the session sequence as an interaction matrix and performs horizontal and vertical convolutions on the matrix to extract the sequence presentation. Other typical works include [4,5].

Although the above methods effectively handle a single session, they are unsuitable for modeling multi-session sequences. Therefore, GNN-based methods are increasingly used in SBR. They combine multiple session sequences to build session graphs and then use GNN to model the complex relations between items [19,20]. Session-based Recommendation with Graph Neural Networks (SR-GNN) is the first work to apply GNN to SBR by using Gated Graph Neural Network (GGNN) [7], and Hybrid-order Gated Graph Neural Network for Session-based Recommendation (SRHGNN) proposes a hybrid order GGNN to solve the over smoothing problem in SBR [21]. However, it does not consider other types of information, such as the browse time of items [22] and the co-occurrence frequency of items [23]. So, many subsequent works have been done by combining auxiliary information. To extract features better, attention mechanisms are widely used. Graph Contextualized Self-Attention Network for Session-based Recommendation (GC-SAN) combines self-attention to capture the global dependency of session sequences [24], and Target Attentive Graph Neural Networks for Session-based Recommendation (TAGNN) goes further extended SR-GNN by proposing target-aware attention [8]. However, some drawbacks remain, such as the information loss caused by session graph construction. To address this, Lossless Edge-order preserving aggregation and Shortcut graph attention for Session-based Recommendation (LESSR) designs lossless Edge-Order Preserving Aggregation (EOPA) mechanism and Shortcut Graph Attention (SGAT) mechanism to propagate information efficiently [9]. However, it ignores the user's dynamic interests.

In conclusion, GNN-based methods have great potential for making further progress in SBR but still have some questions to be solved, such as constructing a lossless session graph, taking item

categories and prices into account, considering user's dynamic interest, and integrating users' interests and intentions, and so on. Therefore, this paper focuses on exploring new possible factors affecting user preferences and decisions based on existing research and how to research and design effective feature learning modules to improve the comprehensiveness and accuracy of user feature modeling.

### 3 Method

In this section, the problem definition is presented first, and the detail of PASBR is followed.

#### 3.1 Problem Definition

Let  $I = \{i_1, i_2, \dots, i_m\}$  denote the set of  $m$  items involved in all sessions, where  $|I| = m$  is the total number of items. Each item  $i_j \in I$   $j \in [1, 2, \dots, m]$  has the corresponding category  $c_j^{cg} \in C^{cg}$  and price  $e_j^{pr} \in E^{pr}$ , where  $C^{cg} = \{c_1^{cg}, c_2^{cg}, \dots, c_{|C^{cg}|}^{cg}\}$  is a set of item categories, and  $E^{pr} = \{e_1^{pr}, e_2^{pr}, \dots, e_{|E^{pr}|}^{pr}\}$  is a set of the item price, while  $|C^{cg}| \in \mathbb{R}^1$  and  $|E^{pr}| \in \mathbb{R}^1$  denote the total number of category and price, respectively. SBR aims to predict the next item based on the session sequence  $S = [s_1, s_2, \dots, s_n]$ , where  $n$  is the length of the session sequence. The goal is to calculate the probabilities  $\hat{Y} = \{\hat{y}_1, \hat{y}_2, \dots, \hat{y}_m\}$  for items  $I$ , where  $\hat{y}_j$  indicates the recommendation score of the candidate item  $i_j$ , and the recommendation result is a set of items with top  $k$  recommendation score.

#### 3.2 Architecture of PASBR

PASBR consists of three modules: input module, feature extraction module, and prediction module, shown in Fig. 2.

The inputs of PASBR include a session and the category and price of items in the session. The feature extraction module comprises six layers: the Session Graph Representation Layer (SGRL), the Price-aware Layer (PAL), the User Intention Representation Layer (UURL), the Short-term Interest Representation Layer (STIRL), the Long-term Interest Representation Layer (LTIRL), and the Dynamic Interest Representation Layer (DIRL). These layers learn users' long-term, short-term, and dynamic interests and intentions. The prediction module concatenates the features learned in the feature extraction module to perform recommendation predictions. The details of each module are described below.

#### 3.3 Input Module

The input module aims to learn the initial embedding of  $S$  and its corresponding category and price. The initial embedding  $X = [x_1, x_2, \dots, x_n]$ ,  $P^{pr} = [p_1^{pr}, p_2^{pr}, \dots, p_n^{pr}]$  and  $Q^{cg} = [q_1^{cg}, q_2^{cg}, \dots, q_n^{cg}]$  is obtained by a 2-layer Feedforward Neural Network (FFN), respectively, where  $q_i^{cg}, p_i^{pr} \in \mathbb{R}^d$ , and  $d$  is the dimension of embedding.

#### 3.4 Feature Extraction Module

The module consists of 6 layers to extract users' interests and intentions. SGRL learns the representation of nodes for input sessions. PAL learns users' price tolerance for various items through price and category information. UURL learns users' intentions at the item category level. STIRL and LTIRL are designed to learn the short-term interest and long-term interests of users separately, while DIRL is designed to capture users' dynamic interests over time.

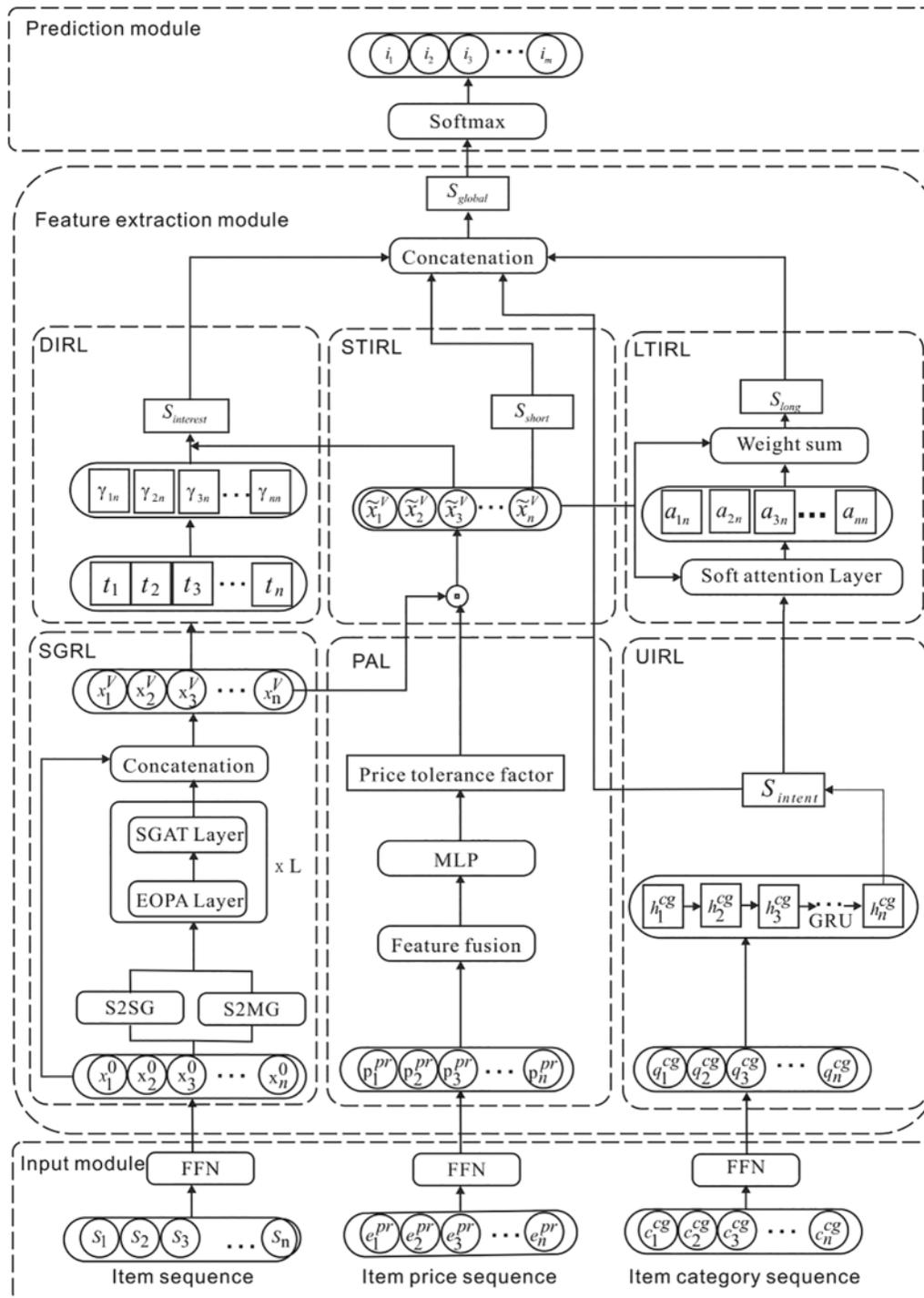
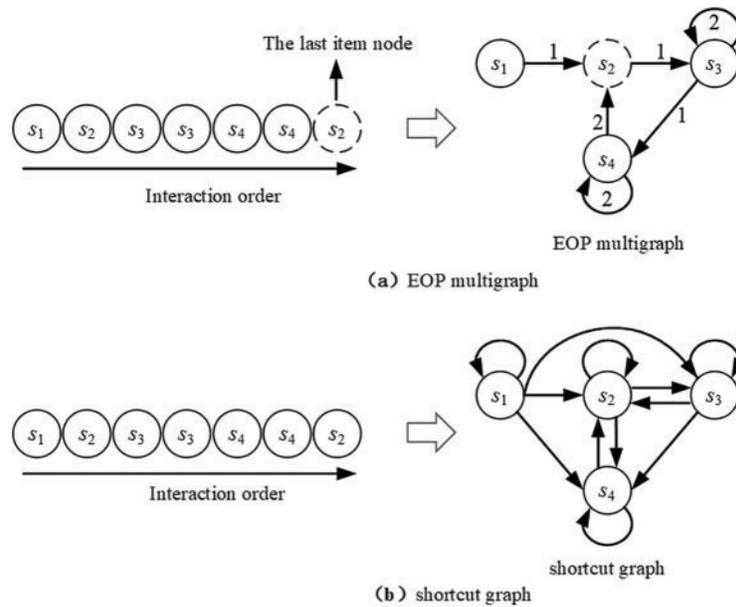


Figure 2: The architecture of PASBR

### 3.4.1 SGRL

The layer constructs session graphs according to the position of items in sessions and then learns the node representations from these session graphs.

When constructing session graphs, most of the current construction methods have an information loss problem; restoring all the original session sequences from the constructed session graphs is impossible. To avoid this problem, constructing two types of lossless information session graphs, referring to [9]: Session to Multigraph (S2MG) and Session to Shortcut Graph (S2SG), as shown in Fig. 3. Notice that besides the same session graph construction mechanisms, PASBR adopted different feature learning method from LESSR.



**Figure 3:** Edge-Order Preserving (EOP) multigraph and shortcut graph

The node representations of session graphs are learned by the EOPA layer and the SGAT layer [9] separately.

Totally  $L$  EOPA and SGAT layers are stacked and the output of the last layer is represented as  $X^V \in \mathbb{R}^{(2L+1)d}$ , which concatenates the output of all previous layers.

### 3.4.2 PAL

The layer aims to learn a new influencing factor-price. Price is the key feature of items, and different users can tolerate different price ranges for different categories of items. The intuition is to combine price and category information of items to model users' price tolerance at the category level.

For this purpose, the price tolerance factor is obtained by fusing the category embedding  $q_i^{cg}$  and the price embedding  $p_i^{pf}$  of the item  $s_i$  first and then encoding fusion result by Multilayer Perceptron (MLP). The possible fusion methods include Concatenation (Concat), Addition (Add),

Multiplication (Mul), and Gated Fusion (GF), and the best method will be chosen through experiments. The computational process is illustrated here by Add method only.

$$\rho_i = \sigma \left( W_{pr} (q_i^{cg} + p_i^{pr}) \right) \quad (1)$$

$$\tilde{x}_i^V = \rho_i \odot x_i^V \quad (2)$$

where  $W_{pr} \in \mathbb{R}^{(2L+1)d \times d}$  denotes the learnable parameter,  $\rho_i \in \mathbb{R}^{(2L+1)d}$  denotes the price tolerance factor of the item  $s_i$ ,  $\sigma$  is sigmoid function,  $\odot$  denotes the corresponding element multiplication,  $\tilde{x}_i^V \in \mathbb{R}^{(2L+1)d}$  is the updated embedding representation of  $s_i$ , and  $\tilde{X}^V$  is updated embedding representations of all items.

### 3.4.3 UIRL

The layer aims to model the users' intention at the category level.

Existing studies typically model users' intention at the item level. However, the category of items can better express users' purpose.

To this end, considering the orderliness of the item category sequence, the sequence transfer pattern between the categories in the session sequence is captured by GRU based on the initial representation  $q_i^{cg}$  of the item category. The calculation process is as follows:

$$h_i^{cg} = \text{GRU} (q_i^{cg}, h_{i-1}^{cg}) \quad (3)$$

where  $h_i^{cg} \in \mathbb{R}^{(2L+1)d}$  is the hidden state of the item category  $q_i^{cg}$ . The initial state  $h_0^{cg}$  is set to a zero vector.

The hidden state of the last interacted item's category is regarded as the user's intention representation, namely  $s_{intend} = h_n^{cg}$ .

### 3.4.4 STIRL

The embedding representation of the last interacted item in the current session is regarded as the user's short-term interest representation, namely  $s_{short} = \tilde{x}_n^V$ .

### 3.4.5 LTIRL

When modeling users' long-term interest, existing studies often adopt the attention mechanism on the last item to learn the importance score of the item to all other items and then capture users' long-term interest by the weighted sum of the embedding representation of all items. However, on the one hand, if the last item is a noise item, such as a false click, the noise will be introduced to the computation, which would affect the recommendation performance. On the other hand, the traditional attention mechanism has high computational complexity. To overcome these problems, first, the influence of the noisy item is reduced by considering the users' intention  $s_{intend}$  when calculating the importance score of items. Then the computation is simplified by replacing traditional attention with soft attention [25]. The process of modeling users' long-term interests  $s_{long}$  is as follows:

$$\eta_{in} = q_{agg}^T \sigma \left( W_1 \tilde{x}_i^V + W_2 \tilde{x}_n^V + W_3 s_{intend} + b_{agg} \right) \quad (4)$$

$$a = \text{softmax}(\eta) = \frac{\eta_{jn}}{\sum_{i \in [1, \dots, n]} \eta_{in}} \quad (5)$$

$$s_{long} = \sum_{i=1}^n a_{in} \tilde{x}_i^V \quad (6)$$

where  $q_{agg}, b_{agg} \in \mathbb{R}^d$ ,  $W_1, W_2, W_3 \in \mathbb{R}^{d \times (2L+1)d}$  are learnable parameters, and  $\eta_{in} \in \mathbb{R}^1$  is the importance score of each item  $s_i$  to the last interacted item  $s_n$ .  $\eta_{in}$  is normalized through Eq. (4).

### 3.4.6 DIRL

This part is designed to capture users' dynamic interests.

Existing studies often capture users' static interest or long-term and short-term interest but ignore dynamic interest. However, users' interest is diverse and dynamic, and the influence of each item on users' interest will gradually fade over time. So, we want to capture users' dynamic interest by assigning a weighted score to each item in the session. To represent the weighted score of items, considering the reverse position encoding [26], which achieved good results in Transformer. However, reverse positional encoding is only designed to encode position information and it is regardless of the importance and weight of each position. But in SBR, in instinctive, historical items should be assigned smaller weights than recent items because the latter can better express users' current interests. To this end, the weighted position score, namely time decay weights, is proposed to represent weight decay over time.

The time decay weights of an item can be calculated as follows. Firstly, the position information  $T = [t_1, t_2, \dots, t_n]$  of each item in the session is extracted, where  $t_i \in \mathbb{N}^*$ ,  $\mathbb{N}^*$  represents a set of positive integers; secondly, the relative position weight  $\phi_{in} \in \mathbb{R}^1$  of each item  $s_i$  according to the last interacted item  $s_n$  is calculated by Eq. (7), and at last, the time decay weight of all items is normalized by Eq. (8).

$$\phi_{in} = \exp(|t_i - t_n| + \log_2(t_n + 1)) \quad (7)$$

$$\gamma = \text{softmax}(\phi) \quad (8)$$

where  $\gamma_{in} \in \mathbb{R}^1$  is the time decay weight of the item  $s_i$  according to item  $s_n$ , and  $\gamma \in \mathbb{R}^n$  denotes the normalized time decay weight vector.

Finally, users' dynamic interest  $s_{interest} \in \mathbb{R}^d$  is obtained by the weighted sum of the latest embedding representation  $\tilde{X}^V$  of the items and the decay weight  $\gamma$  of the items according to Eq. (9).

$$s_{interest} = \sum_{i=1}^n \gamma_{in} \tilde{x}_i^V \quad (9)$$

## 3.5 Prediction Module

The module aims to fuse the multi-perspective features obtained by the feature extraction module to generate the global representation of users in the current session and make the prediction. Specifically, the intent, the short-term and long-term interest as well as the dynamic interest are concatenated, and then the users' global representation  $s_{global} \in \mathbb{R}^d$  is generated by a linear transformation.

$$s_{global} = W_{global} ([s_{short} || s_{intent} || s_{long} || s_{interest}]) \quad (10)$$

where  $W_{global} \in \mathbb{R}^{d \times (4L+4)d}$  is the learnable parameter matrix.

Then the recommendation score of each candidate item is calculated by the inner product between each candidate item  $i_j$  and the global representation  $s_{global}$ , and then the softmax function is used to generate the output vector  $\hat{y}_j$ .

$$\hat{y}_j = \text{softmax}(s_{global}^T x_j) \quad (11)$$

where  $\hat{y}_j$  denotes the recommendation score of the candidate item  $i_j$ .

In the training process, the cross-entropy loss is used, and the dropout strategy is adopted to keep the training process stable.

$$L(y, \hat{y}) = - \sum_{j=1}^m y_j \log(\hat{y}_j) \quad (12)$$

where  $y_j$  denotes the ground-truth probability distribution of the next item, which is a one-hot vector.

## 4 Experiments and Analysis

In this section, PASBR is evaluated on two datasets, and the experiments are designed to answer the following questions.

RQ1: Does PASBR outperform representative baselines?

RQ2: How do different components of PASBR affect performance?

RQ3: How do the key hyper-parameters affect the performance?

RQ4: How is the computational cost of PASBR?

### 4.1 Experimental Settings

The experimental settings include datasets, evaluation metrics, experimental settings, and baselines.

#### 4.1.1 Datasets

Yoochoose and Diginetica are used in the experiments. Yoochoose is a 6-month user click-stream on an e-commerce website published in the RecSys Challenge 2015. Diginetica is used as a challenging dataset for the ACM Conference on Information and Knowledge Management (CIKM) Cup 2016, and only the transaction data among Diginetica is used in this study.

Considering the different data scales of Diginetica and Yoochoose and following the same experimental settings as in [9], the test set for Diginetica and Yoochoose is selected from the last week and the last day respectively. The training sets of Yoochoose are selected from the last 1/10 and 1/4 of the Yoochoose because the original dataset is too large, denoted as Yoochoose1/10 and Yoochoose1/4, respectively. Furthermore, similar to [9], the sequence is spliced in chronological order to increase the training samples. For example, a complete session sequence  $S = [s_1, s_2, \dots, s_n]$  will be split into multiple sub-sequences and corresponding labels denoted as  $([s_1], s_2)$ ,  $([s_1, s_2], s_3)$ ,  $\dots$ ,  $([s_1, s_2, \dots, s_{n-1}], s_n)$ , and so on.

The statistics of the datasets are shown in [Table 1](#).

**Table 1:** Datasets

Dataset	Item number	Category number	Price number	Sequence number
Yoochoose 1/10	14048	12	51	112,088
Yoochoose 1/4	14048	12	51	280,221
Diginetica	42596	993	15	188,636

#### 4.1.2 Evaluation Metrics

Two widely used evaluation metrics for SBR, HR@20 and MRR@20, are adopted to evaluate the effectiveness of PASBR [9].

#### 4.1.3 Experimental Settings

In the experiments, the models are implemented by Tensorflow. The experiments run on a Linux server with Intel E5-2678 v3 @ 2.50 GHz CPU and GeForce RTX 2080 Ti 62G GPUs.

The Optimizer is Adam. The dropout rates are selected from {0.1, 0.2, 0.3, 0.4, 0.5}, and the embedding size is selected from {16, 32, 64, 128, 256}.

#### 4.1.4 Baselines

Considering four groups of competitive baselines for performance comparison as below.

**Group I** includes two non-sequential methods based on the traditional shallow model, Popularity Predictor (POP) and SKNN [15].

In POP, the recommendation is based on the items' frequency, aiming to recommend the most popular items to users.

SKNN recommends items that are like the previous items in the session. The cosine similarity between two items defines the similarity.

**Group II** includes three RNN-based methods, GRU4Rec [2], NARM [3], and the Short-Term Attention/Memory Priority Model for Session-based Recommendation (STAMP) [18].

GRU4Rec makes use of GRU to mine sequential patterns within a session.

NARM introduces the attention mechanism into RNN to capture the users' primary purpose and sequence behavior.

STAMP models users' current and long-term interests by incorporating an attention mechanism.

**Group III** is a CNN-based method, NextItNet [4].

NextItNet employs dilated convolution to expand the receptive field to model long-term dependencies.

**Group IV** includes four GNN-based methods, SR-GNN [7], TAGNN [8], LESSR [9], GC-SAN [24], and SRHGNN [21].

SR-GNN employs GNN to capture complex sequential dependencies within a session.

TAGNN employs GNN to capture item transfer patterns during a session to learn users' dynamic interests in target items.

LESSR tackles the information loss problem of GNN-based models by introducing SGAT and EOPA layers.

GC-SAN combines GNN with a self-attention mechanism to capture the local and global dependencies between items.

SRHGNN proposes a hybrid order GGNN to address the over-smoothing problem in SBR.

## 4.2 Results and Analysis

### 4.2.1 Comparison with Baselines

In response to RQ1, PASBR is compared with the baselines mentioned in Section 4.1.4. [Table 2](#) summarizes the experimental results, and the highest performance is highlighted in bold; the suboptimal performance is shown as wavy lines and Improv. (%) indicates the improvement percentage of PASBR over the suboptimal performance.

**Table 2:** Performance comparison of PASBR and baselines

Method	Yoochoose 1/10		Yoochoose 1/4		Diginetica	
	HR@20	MRR@20	HR@20	MRR@20	HR@20	MRR@20
POP	4.57	1.24	3.87	1.09	0.93	0.21
SKNN	65.85	31.39	68.54	32.30	49.35	16.48
GRU4Rec	61.74	23.62	63.41	24.41	22.73	6.03
STAMP	63.15	28.42	68.21	30.65	44.76	14.08
NARM	66.04	32.86	71.41	35.37	50.43	17.55
NextItNet	65.57	31.91	68.50	33.28	41.69	14.10
SR-GNN	<u>68.48</u>	<u>34.05</u>	72.41	35.95	50.17	17.17
GC-SAN	67.37	33.52	72.37	35.67	51.02	17.87
TAGNN	67.31	33.99	72.33	<u>36.22</u>	<u>52.42</u>	18.34
LESSR	67.98	33.84	<u>72.50</u>	35.70	51.65	17.93
SRHGNN	67.28	31.70	72.34	33.05	51.92	<u>18.48</u>
PASBR	<b>68.97</b>	<b>34.37</b>	<b>73.12</b>	<b>36.47</b>	<b>53.22</b>	<b>18.63</b>
Improv. (%)	0.71	0.94	0.85	0.69	1.52	0.76

[Table 2](#) shows that PASBR outperforms all baselines on different datasets. In addition, DNN-based methods usually outperform traditional methods, and SKNN achieves excellent results and demonstrates the importance of collaborative information in SBR.

In RNN-based baselines, STAMP outperforms GRU4Rec, which demonstrates the effectiveness of combining RNN with attention. NARM usually performs better than STAMP and GRU4Rec, which indicate the effectiveness of emphasizing the users' primary purpose.

One possible reason for the performance of the CNN-based method NextItNet lower than the RNN-based method NARM and GNN-based methods is that the CNN-based method is not good at capturing long-term dependencies.

The GNN-based methods outperform CNN-based and RNN-based methods, perhaps because GNN is effective in mining more complex dependencies and deeper features between items in a session. Because TAGNN pays more attention to users' dynamic interests, its overall performance is slightly better than other GNN-based methods. And PASBR is designed to capture user features comprehensively so it achieves the best performance on different datasets.

Because PASBR is designed to capture user features comprehensively, it achieves the best performance on different datasets.

#### 4.2.2 Ablation Studies

In response to RQ2, the ablation experiments are performed over the key components of PASBR. Three variants of PASBR are designed to verify the effectiveness of PAL, UIRL, and DURL, respectively, namely PASBR-price, PASBR-intent, and PASBR-interest. A variant is designed to move the corresponding layer from PASBR. The experimental results are shown in Table 3.

**Table 3:** Performance comparison of PASBR and its variants

Method	Yoochoose 1/10		Yoochoose 1/4		Diginetica	
	HR@20	MRR@20	HR@20	MRR@20	HR@20	MRR@20
PASBR-price	68.70	34.11	72.73	36.04	52.91	18.28
PASBR-intent	68.77	34.29	72.80	36.07	53.08	18.35
PASBR-interest	68.78	34.25	72.63	35.79	52.68	18.19
PASBR	<b>68.97</b>	<b>34.37</b>	<b>73.12</b>	<b>36.47</b>	<b>53.22</b>	<b>18.63</b>

Table 3 shows the results in which PASBR outperforms its three variants.

The substantial difference between PASBR and PASBR-price suggests that considering users' price tolerance on various items can improve recommendation accuracy. The improvement of PASBR over PASBR-intent validates that the UIRL effectively extracts users' intentions. Furthermore, a comparison of PASBR-interest to PASBR demonstrates the importance of capturing the users' dynamic interest. In conclusion, the above ablation experiments validate the effectiveness of the three core components of PASBR.

#### 4.2.3 Hyper-Parameter Analysis

In response to RQ3, four experiments are performed involving key hyper-parameters, including embedding size  $d$ , dropout rate  $r$ , number of combination layer  $L$ , and fusion function  $f$ .

The first experiment involves  $d$ . Fig. 4 shows that increasing the embedding size does not always result in better performance. When  $d = 32$  on the Yoochoose1/10 and Diginetica, and  $d = 64$  on Yoochoose1/4, PASBR reaches the best performance. The embedding size  $d$  needs to be set appropriately for different datasets.

The second experiment involves  $r$ . The experiment results of different dropout rates  $r$  ranging in  $\{0.1, 0.2, 0.3, 0.4, 0.5\}$  are shown in Fig. 5. The two datasets have different optimal dropout rates, and one possible reason is that the two datasets have different sparsity.

The third experiment involves  $L$ . From Fig. 6, the optimal number of combination layers is  $L = 2$ . The performance decreases when it exceeds this optimal value. One possible reason is that the model overfits when the number of combination layers increases.

The fourth experiment involves  $f$ . The experiment here aims to analyze the effect of the different fusion functions when fusing item category information and price information in PAL. In the experiment,  $f$  is chosen from Add, Concat, GF, and Mul. The results are shown in Fig. 7. From Fig. 7, the optimal fusion function is Add. Notice that Concat performs poorly, and one possible reason is that Concat can't effectively fuse the underlying associative relationship between two features and can't handle the differences between them.

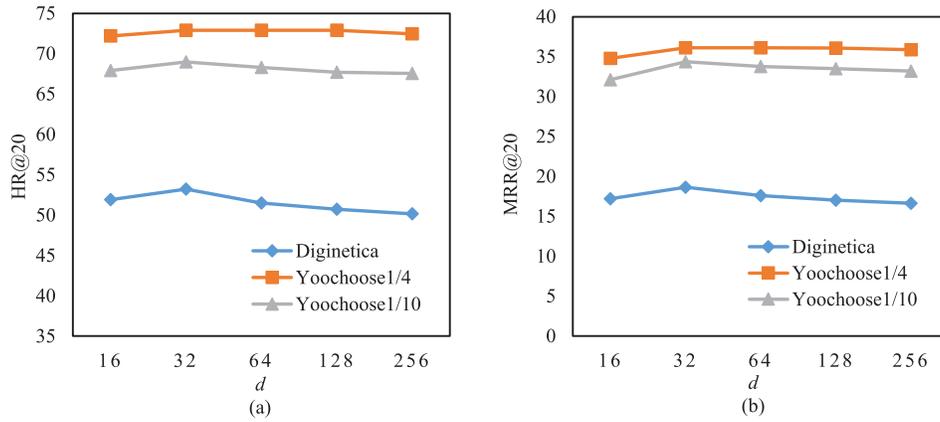


Figure 4: Performance under different  $d$

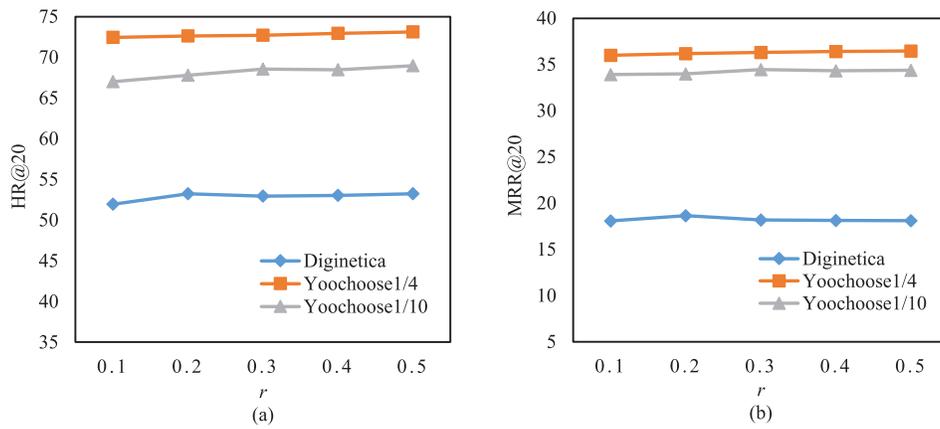


Figure 5: Performance under different  $r$

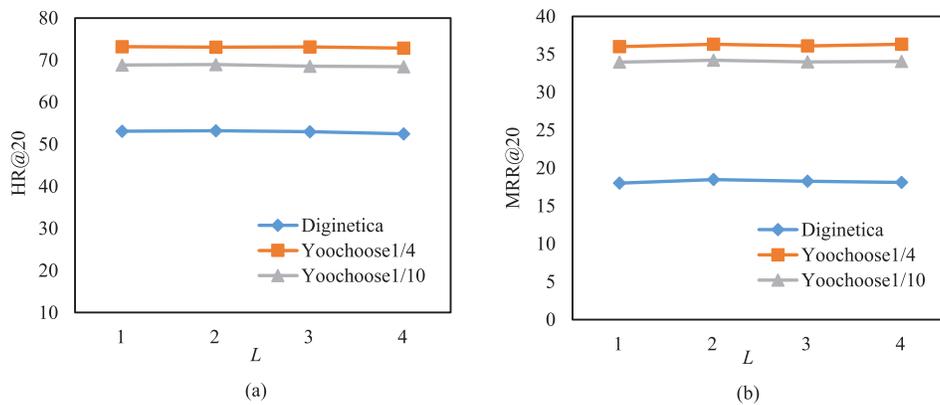


Figure 6: Performance under different  $L$

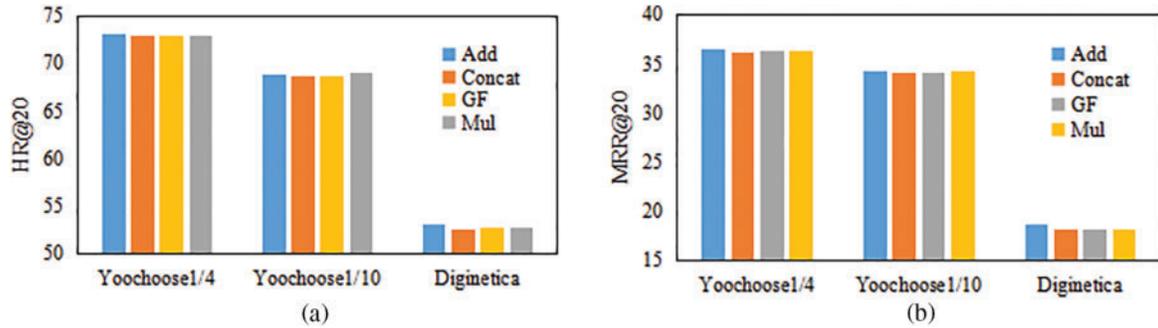


Figure 7: Performance under different  $f$

#### 4.2.4 Comparison of Efficiency

In response to RQ4, the training efficiency of PASBR is evaluated. Taking Diginetica as an example, the running time of all methods in an epoch is recorded, and the results are shown in Fig. 8.

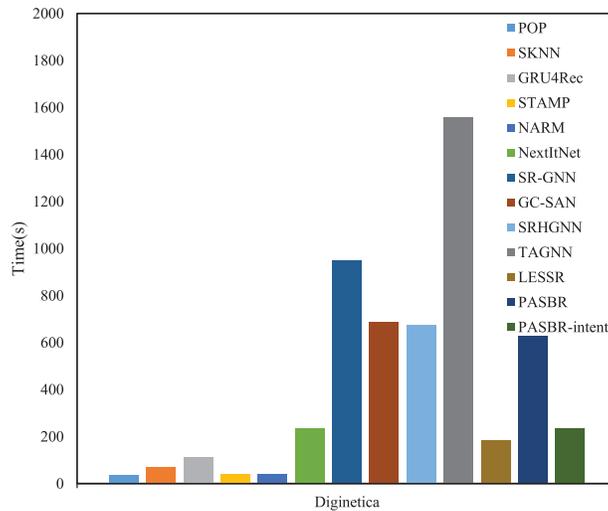


Figure 8: The running time of each model

In general, the shallow models are more efficient, while GC-SAN, SR-GNN, LESSR, TAGNN, and PASBR take relatively more time because they require time to construct session graphs. PASBR takes more time than its variant because PASBR has additional modules. However, overall, the running time of PASBR is acceptable, considering the improvement of model performance.

The experimental results show that PASBR is a feasible and effective method.

## 5 Conclusions

In this paper, a novel price-aware session-based recommendation model PASBR is proposed, which combines the intent, the short-term interests, the long-term interests, and the dynamic interests of the user to improve recommended performance. The innovative design includes: firstly, item price and category information are combined to explore users' price tolerance for various items, which effectively models user intention from the level of the item's category; secondly, a new method is

developed to capture the dynamic nature of users' interests evolving, fully considering users' preference features, and incorporating users' current intentions in modeling long-term user interests to reduce the problems caused by false clicks. Extensive experimental analysis verifies that PASBR outperforms existing representative recommendation methods.

In the future, contrastive learning will be applied to SBR, leading to further advances in recommendation performance. Furthermore, the incorporation of additional auxiliary information, such as browsing time and item co-occurrence frequency, holds great potential for enhancing the effectiveness of PASBR.

**Funding Statement:** The authors received no specific funding for this study.

**Conflicts of Interest:** The authors declare that they have no conflicts of interest to report regarding the present study.

## References

- [1] C. Gao, X. Wang, X. N. He and Y. Li, "Graph neural networks for recommender system," in *Proc. WSDM*, Online, pp. 1623–1625, 2022.
- [2] B. Hidasi, A. Karatzoglou, L. Baltrunas and D. Tikk, "Session-based recommendations with recurrent neural networks," in *Proc. ICLR*, San Juan, PR, USA, pp. 1–10, 2016.
- [3] J. Li, P. J. Ren, Z. M. Chen, Z. C. Ren, T. Lian *et al.*, "Neural attentive session-based recommendation," in *Proc. CIKM*, Singapore, Singapore, pp. 1419–1428, 2017.
- [4] F. J. Yuan, A. Karatzoglou, I. Arapakis, J. M. Jose and X. N. He, "A simple convolutional generative network for next item recommendation," in *Proc. WSDM*, Melbourne, VIC, Australia, pp. 582–590, 2019.
- [5] J. X. You, Y. C. Wang, A. Pal, P. Eksombatchai, C. Rosenburg *et al.*, "Hierarchical temporal convolutional networks for dynamic recommender systems," in *Proc. WWW*, San Francisco, CA, USA, pp. 2236–2246, 2019.
- [6] J. X. Tang and K. Wang, "Personalized top-N sequential recommendation via convolutional sequence embedding," in *Proc. WSDM*, Marina Del Rey, CA, USA, pp. 565–573, 2018.
- [7] S. Wu, Y. Y. Tang, Y. Q. Zhu, L. Wang, X. Xie *et al.*, "Session-based recommendation with graph neural networks," in *Proc. AAAI*, Honolulu, HI, USA, pp. 346–353, 2019.
- [8] F. Yu, Y. Q. Zhu, Q. Liu, S. Wu, L. Wang *et al.*, "TAGNN: Target attentive graph neural networks for session-based recommendation," in *Proc. SIGIR*, Online, pp. 1921–1924, 2020.
- [9] T. W. Chen and R. C. W. Wong, "Handling information loss of graph neural networks for session-based recommendation," in *Proc. SIGKDD*, Online, pp. 1172–1180, 2020.
- [10] X. K. Zhang, B. Xu, L. Yang, C. L. Li, F. L. Ma *et al.*, "Price does matter! Modeling price and interest preferences in session-based recommendation," in *Proc. SIGIR*, Madrid, ESMD, Spain, pp. 1684–1693, 2022.
- [11] F. Abel, I. I. Bittencourt, N. Henze, D. Krause and J. Vassileva, "A rule-based recommender system for online discussion forums," in *Proc. AH*, Hannover, LS, Germany, pp. 12–21, 2008.
- [12] W. Y. Lin, S. A. Alvarez and C. Ruiz, "Efficient adaptive-support association rule mining for recommender systems," *Data Mining and Knowledge Discovery*, vol. 6, no. 1, pp. 83–105, 2002. <https://doi.org/10.1023/A:1013284820704>
- [13] B. Sarwar, G. Karypis, J. Konstan and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proc. WWW*, Hong Kong, China, pp. 285–295, 2001.
- [14] S. Rendle, C. Freudenthaler and L. Schmidt-Thieme, "Factorizing personalized Markov chains for next-basket recommendation," in *Proc. WWW*, Raleigh, NC, USA, pp. 811–820, 2010.
- [15] D. T. Le, Y. Fang and H. W. Lauw, "Modeling sequential preferences with dynamic user and context factors," in *Proc. ECMLPKDD*, Riva del Garda, Italy, pp. 145–161, 2016.

- [16] T. Z. Zang, Y. M. Zhu, J. Zhu, Y. N. Xu and H. B. Liu, "MPAN: Multi-parallel attention network for session-based recommendation," *Neurocomputing*, vol. 471, pp. 230–241, 2022.
- [17] S. Zhang, Y. Tay, L. N. Yao, A. X. Sun and J. An, "Next item recommendation with self-attentive metric learning," in *Proc. AAAI*, Honolulu, HI, USA, pp. 1–9, 2019.
- [18] Q. Liu, Y. F. Zeng, R. Mokhosiand and H. B. Zhang, "STAMP: Short-term attention priority model for session-based recommendation," in *Proc. KDD*, London, United Kingdom, pp. 1831–1839, 2018.
- [19] J. Y. Guo, P. Y. Zhang, C. Z. Li, X. Xie, Y. Zhang *et al.*, "Evolutionary preference learning via graph nested GRU ODE for session-based recommendation," in *Proc. CIKM*, Atlanta, GA, USA, pp. 624–634, 2022.
- [20] M. Zhang, S. Wu, M. Gao, X. Jiang, K. Xu *et al.*, "Personalized graph neural networks with attention mechanism for session-aware recommendation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 8, pp. 3946–3957, 2022. <https://doi.org/10.1109/TKDE.2020.3031329>
- [21] Y. H. Chen, L. Huang, C. D. Wang and J. H. Lai, "Hybrid-order gated graph neural network for session-based recommendation," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 3, pp. 1458–1467, 2022. <https://doi.org/10.1109/TII.2021.3091435>
- [22] X. Sun, X. J. Liu, B. Li and K. Liang, "Session sequence recommendation based on graph neural network and temporal attention," *Computer Engineering and Design*, vol. 41, no. 10, pp. 2913–2920, 2020.
- [23] R. H. Qiu, J. J. Li, Z. Huang and H. B. Yin, "Rethinking the item order in session-based recommendation with graph neural networks," in *Proc. CIKM*, Beijing, China, pp. 579–588, 2019.
- [24] C. F. Xu, P. P. Zhao, Y. C. Liu, V. S. Sheng, J. J. Xu *et al.*, "Graph contextualized self-attention network for session-based recommendation," in *Proc. IJCAI*, Macao, China, pp. 3940–3946, 2019.
- [25] S. L. Xu, F. Zhang, X. S. Wei and J. Wang, "Dual attention networks for few-shot fine-grained recognition," in *Proc. AAAI*, Online, pp. 2911–2919, 2022.
- [26] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones *et al.*, "Attention is all you need," in *Proc. NIPS*, Long Beach, CA, USA, pp. 6000–6010, 2017.