Tabriz Journal of Electrical Engineering (TJEE), vol. 55, no. 1, 2025 Serial no. 111
DOI: 10.22034/tjee.2024.60844.4821

A Position-Aware Self-Attention Neural Model for Personalized Session-Based Recommendation

Azam Ramazani, Ali-Mohammad Zareh-Bidoki*, Mohammad-Reza Pajoohan

Department of Computer Engineering, Yazd University, Yazd, Iran.
ramazani.azam@stu.yazd.ac.ir, alizareh@yazd.ac.ir, pajoohan@yazd.ac.ir
*Corresponding author

Received: 09/03/2024, Revised:12/08/2024, Accepted: 28/09/2024.

Abstract
Personalized session-based recommendation seeks to predict the user's next click or interaction based on the user's previous interactions and behaviour in the present and historical sessions. Recent research has focused on self-attention networks (SANs) to capture users' global interests and behaviours. Thanks to their modelling of the global dependencies among session interactions, SANs have exhibited superior performance in session-based recommendation compared to other deep network approaches. These networks do not account for items' position and order in the session sequence. Nonetheless, the sequential data of the sessions and the item order in the session sequence can reflect the user's sequential interests and improve the recommendation's performance. This article proposes a Position-Aware Self-Attention Neural (PASAN) model for personalized session-based recommendations. The model uses a reverse-position encoding mechanism that assigns position embedding to items based on their order in the session sequence. The PASAN model jointly learns global interests through the self-attention network and sequential interests via positional encoding. Moreover, it employs users' historical and current session interactions to model their long-term preferences. First, PASAN is trained using anonymous sessions. Subsequently, personalized recommendations are provided for individual users by combining the current session with their historical sessions in a weighted manner. Experiments conducted on two real-world datasets show that the proposed model outperforms the current methods. The proposed model has improved by about 20% in terms of precision and about 8% in terms of mean reciprocal rank on the Reddit dataset.

Keywords
Session-based recommendation, Personalized recommendation, Self-attention networks, Deep learning.

1. Introduction

Given the growing volume of information and services on the web, many e-commerce websites and online services have developed recommendation systems to predict users' interests and assist them in searching for and quickly locating the information and items they seek. Traditional recommendation systems use Matrix Factorization (MF) and Collaborative Filtering (CF) algorithms to model users' interests and preferences [1-3].

These methods predict users' interests based on the matrix of users' interactions on the set of items and the similarity between users. They necessitate a substantial volume of user interaction data, i.e., their long-term history. Nevertheless, they frequently underperform in generating accurate recommendations due to the cold start and a sparse user-item interactions matrix. Some studies have employed hybrid recommendation systems that combine collaborative filtering with additional data sources to address the cold start problem. For instance, Reference [4] uses the knowledge base extracted from interactions among the target user's neighbors, and Reference [5] incorporates user demographic data along with collaborative filtering. However, the current methods for modeling user interests are static, although user preferences are dynamic and subject to constant change.

Session-based recommendation (SBR) systems have recently gained substantial attention as measures to address these challenges and model user behaviour dynamics. A session sequence refers to a series of user interactions, such as clicks or purchases, that occur in a specific timeframe. A user's sessions can occur in a day or over several days, weeks, or months. Per the user's actions and clicks in the current and previous sessions, SBR systems attempt to make inferences about what the user will do next [6,7]. Several SBR approaches have been developed to model patterns and dependencies between items, including sequential and non-sequential ones. Earlier research on SBR has built on the similarity between items [8] and the Markov chain (MC) [9,10]. Similarity-based models disregard the order of items and sequential information between session interactions. MC models incorporate session sequence information but assume independent interactions and foretell the subsequent click only according to the user's last or a few preceding clicks. Consequently, these models have limited accuracy in making recommendations, particularly in complex sessions.

Deep learning techniques have gained attention from researchers for SBR and have notably enhanced the performance of recommendation systems in recent years. Several recurrent neural network (RNN) methods have been suggested for SBR. These methods focus on

learning the item order and sequential patterns within session sequences [11-14]. They only account for dependencies between adjacent and consecutive items and cannot capture complex interdependencies among items from a long distance. Several studies have utilized graph neural networks (GNNs) to improve SBR, including references [15-17]. These methods represent session sequences as session graphs, which are directed graph structures. They are more advanced than previous sequential methods because they can extract complex and rich dependencies and transitions between items. RNN- and GNN-based methods capture the local dependence among adjacent items while disregarding the global similarity between session items. Two sessions with identical sets of items but different consecutive patterns are considered distinct, while they may have the same target item. Hence, they are unable to provide accurate recommendations.

Recently, self-attention networks (SANs) have gained widespread application in a number of fields, such as natural language processing, computer vision, and recommendation systems. These networks draw inspiration from the transformer network architecture [18]. The ability to parallelize calculations and processes makes these networks faster and more efficient than RNN and GNN networks. SANs have demonstrated exceptional SBR capabilities; they predict users' behaviour based on learning global or long-term dependencies between items, regardless of distance [19,20]. These networks do not account for the sequence of session interactions and the position of items, notwithstanding the significance of item sequence in SBR.

Although these SBR approaches have yielded effective results, they have some limitations. Firstly, these approaches exclusively concentrate on modeling either sequential or global dependency between items. In order to enhance the recommended performance, learning only one type of interaction dependency is insufficient. Instead, learning all interaction dependency types is necessary to predict user interests accurately and obtain optimal session representation. Secondly, many approaches only consider users' current session interactions when predicting their interests and preferences, disregarding the transition patterns and dependencies between items from users' historical sessions. User preferences are influenced by their past behaviours and interactions. Therefore, relying solely on current session interactions may not accurately represent user preferences.

This article proposes a position-aware self-attention recommendation model called PASAN, which provides personalized SBR. It jointly learns users' local sequential patterns and global interest in a session. The proposed model employs a reverse-position mechanism that assigns a positional embedding vector to items based on their order in the session sequence. This allows the modeling of sequential dependencies between session items. SANs are also used to model the global interdependence of items. The PASAN model comprises four layers: embedding, self-attention, prediction, and combination. Initially, the embedding layer incorporates the positional embedding vector of items with the embedding vector of items for each session. The resulting representation vector is then fed into the SAN layer. The SAN layer is trained using sessions from multiple anonymous users to generate latent vectors for the items. In SAN, the latent vector of the last item in a session reflects both the current and the global user interests in that session. Hence, this vector is used as the session representation vector. In order to provide the user with personalized recommendations, the current session sequence and some of the user's recent historical sessions are passed through the network separately. Then, the prediction layer calculates the item scores for each session. Finally, in the combination layer, the ranking scores obtained from user sessions are combined in a weighted manner to produce the ultimate recommended items.

Compared to existing methods, our proposed approach simultaneously extracts various types of dependencies among session interactions, including sequential and global dependencies, using a single network. Additionally, it utilizes both the user's current and historical sessions to model the user's short- and long-term preferences. It can also provide recommendations without encoding the user's historical sessions.

The following are the article's contributions:

- PASAN, a position-aware self-attention neural model, is proposed for personalized recommendations. It jointly captures the global dependencies between session interactions based on SANs, and the sequential dependencies between users' session sequence interactions by employing a reverse-position encoding mechanism and embedding the positions of items.
- The PASAN model incorporates users' long-term preferences and generates personalized recommendations by combining the weighted current session and historical sessions. Moreover, the proposed model can offer recommendations only according to the user's current session interactions, even if their historical session interactions are unavailable or if the user is anonymous.
- The model is assessed through extensive experiments on two real-world datasets Reddit and Xing, and compared to current methods based on precision (P@K) and mean reciprocal rank (MRR). Furthermore, the model's key components have been examined for their effect on recommendation performance.

The subsequent sections of the article are organized in the following order: Section 2 examines and reviews current research methods. Section 3 presents the problem statement and details of the proposed model. In Section 4, the experiment and evaluation results are presented. The research concludes with a final section that outlines directions for future works.

2. Related Work

SBR systems have been extensively researched. The studies may be categorized into traditional and deep learning methods. This section reviews research conducted in this field.

2.1. Traditional Methods

CF, specifically MF, is a traditional technique prevalently used in recommendation systems. MF is a technique that maps users and items to a low-dimensional latent space by decomposing the user-item interaction matrix. This allows estimating user preferences for items by multiplying user and item vectors [21,22]. Neighborhood methods have been employed in several studies to provide recommendations. Sarwar et al. [1] postulated a recommendation system that employs the K-Nearest Neighbor (KNN) algorithm to determine similarity and identify neighbors based on the user's most recent clicked item. Jannach and Ludewig [23] proposed a generalized model based on KNN algorithm for computing similarity with all session items. These methods use the entire history of user interactions to model the user's overall preferences, treating all interactions with equal importance and disregarding their sequential order.

The MC-based model is a traditional method in recommendation systems that models user preferences using the sequence of interactions. Shani et al. [10] modeled recommendation generation as a sequential optimization problem using a Markov decision process. The hybrid FPMC model [9] employs MC and MF to model two consecutive user actions and predict the next item. He and McAuley [24] proposed a method that integrates similarity models with MC to address sparse sequential recommendations and offer personalized recommendations to users. However, MC-based methods often rely on the last interaction or the last few interactions of the user and can only model short-term and local dependencies between interactions [25].

2.2. Deep Learning Methods

In the past few years, deep learning has significantly enhanced SBR's performance by extracting deep features. RNNs are commonly employed by recommendation systems to analyze the sequential patterns of users' behaviours. GRU4REC [11] is the first SBR model built on RNN, which utilizes gated recurrent unit (GRU) as an improved RNN model to predict the subsequent item in a user's current session. The NARM model [14] uses the attention mechanism on the RNN to obtain the user's primary goal in a session and produce recommendations. The STAMP model [26] employs a multilayer perceptron neural network and an attention network to capture the user's global interests from the long-term memory of a session and the user's current interests from the short-term memory of the user's most recent click. Wang et al. [27] suggested a collaborative SBR model that utilizes RNNs and an attention mechanism and enhances session representation by using session neighborhood information. Zhang and Wang [28] proposed a shared neural model for SBR that learns the user's sequential and global interests. It includes a GRU network with an attention mechanism to extract the sequential interests of the session and a multilayer neural network with residual connections to extract global interests. Kumar et al. [29] introduced a session-based recommendation system that employs a long short-term memory (LSTM) network with an attention mechanism to extract sequential context and focus on target items.

Several recommendation systems have used GNNs as they model complex transitions between user interactions or clicks using a graph structure. Wu et al. [15] introduced SR-GNN, the first SBR model employing GNNs. The authors use graph structures to represent session sequences and employ GNNs to learn the latent vectors of the items. The user's global preferences are extracted by aggregating the latent vectors of the session items using an attention network. The user's current preferences are extracted by the latent vector of the session's last item. The Global Context Enhanced GNNs (GCE-GNN) model, proposed by Wang et al. [30], learns session-level and global-level embeddings simultaneously using the current and global session graphs on all sessions. A target-aware GNN model was proposed by Yu et al. [31] for SBR. User interests are modeled by taking into account the target item and complex transitions between items in a session. Chen and Wong [32] handle the information loss problem in GNN-based models for SBR by combining edge-order preserving aggregation and shortcut graph attention layers.

E-GNN [33] introduces an enhanced GNN for session-based recommendation. It models the interaction sequence of all anonymous users as a weighted global graph and the interactions of the user's current session as a local session graph. By integrating these two graphs, E-GNN extracts the real interaction patterns of the target user. HGCAN [34] proposes a heterogeneous graph enhanced category-aware attention network for session-based recommendation. It utilizes the heterogeneous global graph to extract patterns between items and categories and then employs the category-aware attention network to estimate the target category and predict the recommendation. RNN- and GNN-based methods disregard the global dependency among session items and solely rely on sequence order.

In recent years, self-aware networks implemented in transformer network architecture have been used by many applications for natural language processing, machine translation, and recommendation systems. These networks are highly efficient and perform well because they model sequences without recurrence or convolution and accelerate network training by parallelizing attention mechanism calculations. Fang [19] presented SR-SAN, a SAN-based model for SBR. The author applied the self-attention mechanism immediately after the embedding layer and used the latent vector of the most recent sequence item to represent the session. Several models [16,35-37] have utilized the self-attention mechanism following GNN or GRU in order to learn the global dependence among session items. However, SANs ignore the sequential information of session interactions.

Several methods have utilized item position information to enhance performance. Wang et al. [38] proposed a position-aware gated graph attention network for SBR. This method assigns position embeddings to nodes in the session graph as per the items' order in the sequence. It subsequently uses a SAN to model global dependencies among items. Sang et al. [39] introduced a position-aware GNN model for SBR. This model represents the sequence of sessions as position-aware graphs to

incorporate item position information. An attention mechanism is subsequently employed to capture users' long-term interests. Lu et al. [40] introduced a global target preference attention network with position information for session-based recommendation. This method integrates item embeddings with item position embeddings in the session and subsequently extracts session target preferences based on a target preference attention layer.

The focus in the aforementioned studies is predominately on the user's current session interactions. They do not account for the user's historical activities and only model their short-term interests. In the past few years, a number of models have utilized historical user sessions as well as the current session sequence to generate recommendations. For instance, references [6] and [41] model user interests according to the current session and the user's historical sessions using a hierarchical architecture comprised of multiple RNNs. The HierTCN model [42] employs a hierarchical architecture comprising a GRU network and a temporal convolutional network (TCN) in order to learn the user's long-term and short-term interests. These models encode historical user sessions as representations and then use this information to predict user interest. This encoding mechanism can, however, restrict the model's capabilities.

A-PGNN [43] is a successful model recently presented. It models each user's sessions, including the current and historical sessions, as a user behaviour graph. By feeding it to a personalized GNN with an attention mechanism, it then extracts personalized structural information in the user's behaviour graph and models the user's long-term preferences. Using a heterogeneous global GNN, HG-GNN [44] extracts the transitions of items across all sessions and learns the user's long-term preferences. It later generates personalized recommendations by combining the user's global preference and the items of the current session via a personalized session encoder. UGG-GCN [45] employs a GNN based on grouping similar users for personalized session-based recommendation. It constructs multiple heterogeneous user-item subgraphs for different user groups with similar item sequences. It models the user's general preferences by integrating subgraphs and further provides personalized recommendations by combining them with the current session. DIARec [46] introduces a hierarchical attention-based recommendation model that first learns dynamic user intents using an intention-aware module focused on modeling item attributes in each session. Subsequently, it combines information from all sessions to create high-level user preferences based on a collaborative multi-head SAN. These models rely on the creation of user embedding, which can pose a limitation for anonymous users. Additionally, these models extract user behavior patterns only based on one type of sequential or global dependency between session interactions.

Our proposed model relies on only one network to extract all types of sequential and global dependencies between session items. To strengthen the recommendations, the proposed model also incorporates the user's historical session sequence in addition to the current session and models the user's short-term and long-term interests. An important advantage of our model is that, unlike previous methods, it does not encode the user's historical sessions and also does not require the creation of user embedding. As a result, it can provide recommendations to the anonymous user for the first time using only their current session's interactions. In this manner, the model is initially built on the behavioural patterns of sessions created by anonymous users. The model then generates personalized recommendations for the user by passing the current session and the user's historical sessions separately through the network model and combining the results.

3. The Proposed Method

In this section, the proposed PASAN model for personalized SBR is introduced. The PASAN model is constructed using the model we presented in [47]. The model we previously suggested only extracts the user's global interests based on SANs and provides personalized recommendations for the user. The proposed PASAN model simultaneously models all types of sequential and global dependencies between interactions by embedding the position of items in the session using reverse-position encoding and combining it with SANs. In general, the PASAN model intends to simultaneously extract the user's sequential and global interests and provide the user with personalized recommendations based on a combination of the current and historical session sequences. As Fig. 1 illustrates, the PASAN model includes four layers in its architecture: embedding, self-attention, prediction, and combination. The SBR problem will be discussed in Section 3.1, which is followed by a description of the various components of the proposed model.

3.1. Problem Statement

The SBR predicts the user's next item or click according to their preceding interactions in a session. In SBR, $V = \{v_1, v_2, ..., v_m\}$ represents the set of all unique items in all sessions, and *m* is the number of all items in *V*. A session sequence is represented by the list $S = [v_1, v_2, ..., v_n]$ with $v_i \in V$. This set contains user clicks in a session, ordered by timestamps, and *n* denotes the length of the session *S*. The SBR aims to predict the subsequent item, i.e., v_{n+1}, in the session *S*. In order to generate recommendations in a SBR model, a recommendation score $\hat{z}$ and corresponding probability $\hat{y}$ for each item in the set *V* are computed for each session *S*. The items with top-*K* values in $\hat{y}$ are suggested to the user as candidate items.

3.2. Embedding Layer

In the embedding layer, first, the items of an input session sequence $S = [v_1, v_2, ..., v_n]$ are mapped to d-dimensional integrated embedding vectors $X = [x_1, x_2, ..., x_n]$ and $x_i \in R^d$. As SAN does not consider the position and order of items, we use the d-dimensional learnable positional embedding vectors $P = [p_1, p_2, ..., p_n]$ and $p_i \in R^d$ in order to learn the

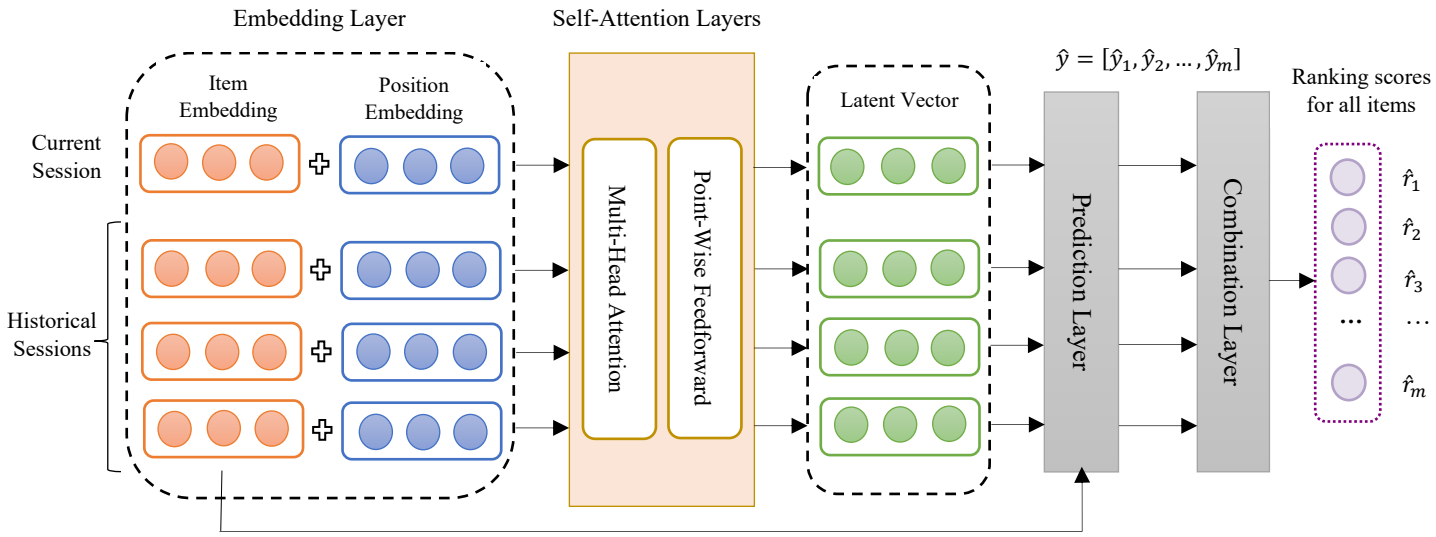


Fig. 1. Architecture of the proposed PASAN model.

sequence of the session and the sequential interests. For each session sequence, the position embedding vectors of items, *P*, are added with the embedding vectors of items *X*. The resulting embedding vectors $E = [e_1, e_2, ..., e_n]$ and $e_i \in R^d$ are obtained as follows:

$$E = \begin{pmatrix} x_1 + p_1 \\ x_2 + p_2 \\ \dots \\ x_n + p_n \end{pmatrix} \quad (1)$$

In order to avoid overfitting, a dropout is applied to the combined vector of item embedding and position embedding (*E*) before sending it to the SAN layer.

3.2.1 Reverse-Position Embedding

In each session sequence, duplicate items in different positions are merged into a single item. For item position information, each item's last position in the session is considered as position embedding. Indeed, interactions close to the current moment are more influential and significant in predicting the user's next item. For instance, the session sequence $S = [v_3, v_2, v_5, v_7, v_5, v_2]$ reduces to $S = [v_3, v_7, v_5, v_2]$ by removing duplicate items and considering the most recent position of each item.

The forward position mechanism encodes items in the same sequence as they appear, resulting in the encoding of current (final) items in the sessions with different position vectors due to varying session sequence lengths. The reverse-position mechanism encodes item position in reverse sequence. This mechanism can intensify the incidence of the same position encoding for the tail items in different session sequences compared to the forward position mechanism. Therefore, in the inverse-position mechanism, the embedding vector of positions that are more important for predicting the target item is learned with greater efficacy [38]. The proposed model utilizes reverse-position encoding.

3.3. SAN Layers

Following the embedding layer, SAN layers are employed to capture the global dependencies between the input and output. The input of these layers is the total vectors of item embedding and position embedding $E = [e_1, e_2, ..., e_n]$. After they pass through the self-attention layers, the items latent vectors of $H = [h_1, h_2, ..., h_n]$ are learned and obtained as the output of the layers. Each SAN layer consists of two sub-layers: multi-head attention and point-wise feedforward network.

3.3.1 Multi-Head Attention

The multi-head attention sublayer encodes each item according to its relationship and dependence on other items in the input sequence. The embedding vectors of session items are initially input into the self-attention layer. For each item, the three vectors of query (Q), key (K), and value (V) are generated by multiplying the item embedding vector in three learnable projection matrices $W^Q, W^K, W^V \in R^{d \times d}$. These vectors are represented as $Q = EW^Q$, $K = EW^K$, and $V = EW^V$. Moreover, the attention function maps a query vector and a set of key-value vector pairs into an output vector. The output of the function is computed as a weighted sum of values, in which the weight of each value is obtained by the scaled dot-product attention mechanism of the query vector through the corresponding key vector. The function is defined as follows:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d}}\right)V \quad (2)$$

The SAN employs a multi-head attention mechanism so as to enable the model to attend to dependencies in various positions using multiple attention heads. The outcomes are derived by aggregating all attention heads

and multiplying the output in a weight matrix $W^O \in R^{d\times d}$. The mechanism is formulated as follows:

$$G = MultiHead(Q,K,V) = Concat(head_1,...,head_h)W^O \quad (3)$$

$$head_i = Attention(EW_i^Q, EW_i^K, EW_i^V) \quad (4)$$

in which h denotes the number of attention heads. Subsequently, the substrate's output is inputted into the feedforward network.

3.3.2 Point-Wise Feedforward Network

The multi-head attention sublayer output is passed through a two-layer fully connected feedforward network with a rectified linear unit (Relu) activation function between them. This network is used on each position in the input separately but equally. Lastly, the latent vectors of session items, i.e., $H = [h_1, h_2, ..., h_n]$, are obtained:

$$F = FFN(G) = \max(0, GW_1 + b_1)W_2 + b_2 \quad (5)$$

in which $W_1, W_2 \in R^{d\times d}$ represents weight matrices and $b_1, b_2 \in R^d$ denotes the bias vectors.

Following the output of both the multi-head attention sublayer and the feedforward network, a residual connection and normalization are applied to preserve the information of the main features, i.e., the information of the lower layer. L2 regularization is employed to prevent network overfitting. We define the SAN layer as follows:

$$F = SAN(G) \quad (6)$$

Several SAN layers can be stacked atop one another in the proposed model to capture various features and learn more complex transitions between items. The first layer is denoted by $F^1 = F$, and the Lth (L>1) layer is defined as follows:

$$F^L = SAN(F^{L-1}), \forall L \in \{1,2,...,n\} \quad (7)$$

where $F^L \in R^{n\times d}$ is the final output from the SAN layers.

3.4. Prediction Layer and Model Training

As described in the preceding section, the SAN generates the latent vector of each item in the session sequence based on the item's overall relationship with and dependence on all other items in the session without taking their distance into account. Given that these item latent vectors are an aggregate of all session items, they can represent the session's global interests. After the latent vectors of the items are captured, the latent vector of the last item clicked is assumed to represent each session, reflecting both the current preference and interest as well as the session's global interests. Subsequently, in the prediction layer, the recommendation score $\hat{z}_i$ for each candidate item $v_i \in V$ is calculated by multiplying the latent vector of the session's last item in the embedding vector of each item as follows:

$$\hat{z}_i = h_n^T x_i \quad (8)$$

In order to acquire the probability of each candidate item $\hat{y}$, the softmax function is later used:

$$\hat{y} = soft\max(\hat{z}) \quad (9)$$

in which $\hat{z} \in R^m$ represents recommendation scores on all candidate items, while $\hat{y} \in R^m$ indicates the probability of the items being clicked subsequently in the session S.

In order to train the proposed model, the cross-entropy loss function between the predicted value and the ground truth is utilized, as defined below:

$$L(\hat{y}) = -\sum_{i=1}^{m} y_i log(\hat{y}_i) + (1-y_i) log(1-\hat{y}_i) \quad (10)$$

where y represents the one-hot encoding vector of the ground truth item. The back-propagation through time (BPTT) is used to optimize this function.

3.5. Combination Layer

For each user, sets of sessions are considered, including the current session and M historical sessions as $S^u = \{S_c^u, S_{h1}^u, S_{h2}^u, ..., S_{hM}^u\}$. S_c^u is the current session of the user u, and S_{hi}^u is the ith historical session of the same user. A personalized SBR system aims to predict the user's next item or click, given the user's set of sessions. After the probability of all candidate items is calculated for each user session, the scores of the user's current and historical sessions are accumulated. A user's current session and historical sessions are not of equal importance and priority. They are combined with different weights in a weighted manner. The aggregated ranking score for each item $\hat{r}_i$ is calculated through the following formula:

$$\hat{r}_i = \sum_{s=1}^{N_{ses}} \alpha_s \hat{y}_{i,s} \quad (11)$$

where α_s is the importance coefficient of each session, $\hat{y}_{i,s}$ is the item click probability i in the session s, and N_{ses} denotes the number of user sessions, including the user's current session and M historical sessions. Since the current session is the most important session and historical sessions have varying importance levels based on their distance from the current session, the formula provided is used to compute the importance coefficient of each session:

$$\alpha_s = \frac{N_{ses} - discurses}{N_{ses}} \quad (12)$$

in which N_{ses} represents the number of user sessions, while *discurses* indicates the session distance from the user's current session.

4. Experiments

We perform extensive experiments in this section to address the following research questions:

RQ1: How does the proposed PASAN model perform as compared to other cutting-edge methods?

RQ2: Do PASAN's key components, such as historical sessions and reverse-position embedding, enhance recommendation performance?

RQ3: How does the number of historical sessions impact our model's performance?

4.1. Experimental Settings

This section describes the datasets, baseline methods, and evaluation metrics employed in our experiments.

4.1.1 Datasets

The proposed model is assessed using two real-world datasets: Reddit[1] and Xing[2], which are widely used for SBR.

- **Reddit:** The Reddit dataset consists of user comments in the form of [username, subreddit, timestamp] records extracted from the Reddit social news website. According to the method used in [41], each user's records are manually split into sessions with a 60-minute time threshold. Sessions with less than 2 interactions are omitted to filter too short sessions and those with insufficient information. Users who have less than 3 sessions are excluded.
- **Xing:** The Xing dataset is released from RecSys Challenge 2016 and contains the interactions of 770,000 users on the job postings website over an 80-day period. User interaction data includes click, bookmark, reply, and delete. Similar to [6], we manually split each user's records into sessions with a 30-minute time threshold. Interactions of type "delete" are discarded. Repeated interactions of the same type within a session are removed to reduce noise. Also, sessions with less than 3 interactions and users with less than 5 sessions are excluded.

Then the following preprocessing is performed on both datasets: Each user's sessions are divided into training and test sets, with 80% of the sessions allocated to the training set and the remaining 20% allocated to the test set. Table I displays the statistical features of the two datasets after preprocessing.

Table I. Statistical features of the datasets.

Dataset	Reddit	Xing
Users	18271	11479
Items	27452	59121
Sessions	1135488	91683
Avg. sessions per user	62.15	7.99
Avg. session length	3.02	5.78
Train sessions	901161	69135
Test sessions	234327	22548

In order to enhance and broaden the datasets, as outlined in [15], the input sequences of the training and test sets are partitioned into sequences and labels. For instance, in the training set, an input session $S=[v_1,v_2,...,v_n]$ is split into sequences and corresponding labels $([v_1],v_2)$, $([v_1,v_2],v_3)$, ..., and $([v_1,v_2,...,v_{n-1}],v_n)$ so that $[v_1,v_2,...,v_{n-1}]$ is the created sequence and v_n denotes its corresponding label. Moreover, in the test data, the set of sessions of the user u, $S^u=\{\{v_{11},v_{12},v_{13}\},\{v_{21},v_{22}\},\{v_{31},v_{32},v_{33}\}\}$ is split into a set of the current session, historical sessions, and the corresponding labels, as follows:

$S^u_{h1}=\{\{v_{11},v_{12},v_{13}\}\}$, $S^u_{c1}=\{\{v_{21}\}\}$, $label_1=v_{22}$

$S^u_{h2}=\{\{v_{11},v_{12},v_{13}\},\{v_{21},v_{22}\}\}$, $S^u_{c2}=\{\{v_{31}\}\}$, $label_2=v_{32}$

$S^u_{h3}=\{\{v_{11},v_{12},v_{13}\},\{v_{21},v_{22}\}\}$, $S^u_{c3}=\{\{v_{31},v_{32}\}\}$, $label_3=v_{33}$

so that S^u_h denotes the user's set of historical sessions, S^u_c is the user's current session, and the last item in the current session is the labeled item.

4.1.2 Baseline Methods

The proposed model's performance is compared with the following baseline models:

- **Pop**: It recommends popular items with the highest number of interactions in the training set.
- **Item-KNN** [1]: It recommends items similar to items clicked in sessions. It uses cosine similarity to calculate similarity.
- **FPMC** [9]: It uses a combination of MC and MF to predict the user's next basket.
- **SKNN** [23]: It calculates the score of the candidate items based on the set of user actions in the current session and *k* sessions from the training set, which is most similar to the current session.
- **VSKNN** [8]: It is a generalized model based on SKNN that considers the sequence and position of items in the session to calculate similarity.
- **GRU4Rec** [11]: It uses an improved RNN model, GRU, for SBR.
- **SR-GNN** [15]: It utilizes GNNs to model complex transitions between items and create SBR.
- **SR-SAN** [19]: It uses SANs to extract global dependencies between session items and provide SBR.

4.1.3 Evaluation Metrics

To assess the performance of the proposed model and those of other methods, we employ two widely-used evaluation criteria:

- **P@K** (Precision): It is employed to evaluate how accurate the prediction is in SBR. It displays the ratio of correctly predicted items among the top-*K* items in the ranking list.

$$P@K=\frac{n_{hit}}{N} \tag{13}$$

N denotes the number of test set sessions and n_{hit} denotes the number of sessions that include the target item among the top-*K* items in the ranking list.

- **MRR@K** (Mean Reciprocal Rank): It indicates the mean value of reciprocal ranks of target items. If the rank of the target item is larger than *K*, the *MRR* value is considered 0.

$$MRR@K=\frac{1}{N}\sum_{i=1}^{N}\frac{1}{rank_i} \tag{14}$$

N shows the number of test set sessions and $rank_i$ indicates the rank of the target item among the top-*K* items in the ranking list concerning the *i*th session. *MRR* accounts for the ranking list's order such that the higher the target item is ranked, the greater the *MRR* value.

4.1.4 Parameter Settings

[1] https://www.kaggle.com/colemaclean/subreddit-interactions

[2] http://2016.recsyschallenge.com/

We have implemented the proposed model using the Pytorch[3] framework. All model parameters are initialized using a Gaussian distribution with the mean 0 and standard deviation $1/\sqrt{d}$. We employ the Adam optimizer with an initial learning rate of 0.001, which is reduced by 0.1 after every 3 epochs. The L2 regularization is set to 10^{-5}, and the batch size is set to 100. The model's hyper-parameters are tuned on 10% of the training set selected as the validation set via grid search. The size of the embedding vectors (d) is set from {32, 64, 96, 128, 160, 256}, the number of layers from {1, 2, 3, 4}, the number of attention heads from {1, 2, 4, 8, 16}, and the dimension of feedforward network from {1, 2, 3, 4, 5, 6} multiplier of the embedding size. Table II presents the hyper-parameters used by the proposed model with the best results on two datasets.

Table II. Hyper-parameter settings.

Dataset	Reddit	Xing
Dimension of embedding vectors	128	128
SAN layers	1	1
Attention heads	8	4
Dimension of feedforward network	4	2
Historical sessions	5	3

The model's highest accuracy is obtained based on five historical sessions for Reddit and three for Xing. The maximum length of each session (maxlength) is 20. In order to prevent the model from overfitting, a dropout = 0.3 is applied to the sum of item embedding and position embedding, and a dropout = 0.1 is used for the output of the SAN sublayers. The results of baseline models are reported based on their default hyper-parameters. Each method has been executed 5 times, and the final results are determined from the average values.

4.2. Model Comparison

This section compares the performance of the proposed PASAN model with baseline models. Tables III and IV display the P@K and MRR@K values obtained by all methods for K=5, 10, and 20 on two datasets. The best results among all the methods are highlighted in bold. The best results on the baseline methods are marked in stars in each column. *Improv.* means the performance enhancement of our model compared to the best baseline method. As demonstrated, our proposed model has higher P@K and MRR@K values and better recommendation performance than competing methods on both datasets. The results are analyzed below.

Traditional recommendation methods rely on shallow and simple features to train the recommendation model and predict the target item, resulting in poor recommendation accuracy. The Pop method generates recommendations based solely on the simple feature of the product's popularity, disregarding specific user interactions with the items; it has performed the worst among the methods. Item-KNN predicts according to the similarity between items, but its performance is unsatisfactory because it disregards the order of session interactions. FPMC obtained a better result than Pop and Item-KNN on Reddit by using MC to consider session sequence information, demonstrating the significance of session sequence information. Given the assumption of independence of consecutive user interactions in a session, this method's ability to create accurate recommendations is limited.

SKNN and VSKNN outperform other traditional methods and have performance comparable to that of deep neural network models. These two methods have obtained high recommendation accuracy on Xing compared to other baseline models. It is because these methods consider sessions similar to the user's current session, which contain interactions similar to the current session's interactions. This demonstrates that user preferences play a crucial role in recommendation generation. Nonetheless, these methods disregard the arrangement of items and their sequential dependencies.

Compared with traditional methods, recommendation methods based on deep learning can model complex user behavior and have achieved significant performance in most cases. These methods can extract deep features of items in session sequences. Using RNNs, GRU4Rec has achieved relatively favorable results. This method relies solely on learning sequential session information and disregards the user's global interests in the session. SR-GNN models the complex transitions between session items by displaying session sequences as graph data structure and has achieved superior results to GRU4Rec by a large margin on Xing. In general, GNN-based methods account for the dependency between adjacent items in the session graph and disregard the more distant ones. Using SANs, SR-SAN models the global dependence between a session's items, i.e., the users' general interests in the session. As such, it has achieved superior results to other baseline models on both datasets. Nonetheless, SR-GNN and SR-SAN ignore information regarding the order of session interactions and the position of items.

The proposed PASAN model outperforms all traditional and deep-based methods in terms of all evaluation criteria on both datasets. On the Reddit dataset, PASAN has improved by 19.56% concerning P@5 and 8.16% regarding MRR@5 compared with the best baseline method. Also, in the Xing dataset, PASAN has an improvement of 11.48% and 10.09% in terms of P@5 and MRR@5 over the best baseline model, respectively, which demonstrates its high effectiveness.

In addition, we have measured the accuracy of the proposed model for values of K=1 and 3 on two datasets. The results are presented in Table V. As the results indicate, our proposed model has a satisfactory recommendation performance for small K values. In general, the PASAN model's success can be attributed to the following factors: 1) It simultaneously captures all types of session interaction dependencies. It uses the SAN to learn the global dependency and employs the reverse-position encoding mechanism to learn the sequential dependency between the session items. This enables it to model the user's behavior more accurately.

[3] https://pytorch.org/

Table III. Performance of PASAN compared to other baseline methods on the Reddit dataset.

Method	P@5	P@10	P@20	MRR@5	MRR@10	MRR@20
Pop	13.22	19.46	26.47	8.50	9.32	9.82
Item-KNN	21.71	30.32	38.85	11.74	12.88	13.49
FPMC	29.91	34.31	44.32	8.78	6.56	4.54
SKNN	34.29	42.17	49.68	19.11	20.16	20.68
VSKNN	34.25	42.17	49.67	19.09	20.14	20.67
GRU4Rec	33.72	41.73	50.04	24.36	25.42	26.00
SR-GNN	34.96	42.38	50.33	25.90	26.88	27.44
SR-SAN	36.24*	43.56*	51.13*	26.34*	27.32*	27.84*
PASAN	**43.33**	**52.37**	**61.39**	**28.49**	**29.69**	**30.32**
Improv.	19.56%	20.22%	20.07%	8.16%	8.67%	8.91%

Table IV. Performance of PASAN compared to other baseline methods on the Xing dataset.

Method	P@5	P@10	P@20	MRR@5	MRR@10	MRR@20
Pop	0.21	0.26	0.58	0.08	0.09	0.11
Item-KNN	8.79	11.85	14.67	5.01	5.42	5.62
FPMC	1.70	2.42	3.27	0.61	0.50	0.37
SKNN	14.36	19.42	24.12	9.29	9.80	10.22
VSKNN	14.46*	19.60*	24.25*	9.48	10.07	10.39
GRU4Rec	10.35	13.15	15.30	5.94	6.36	6.69
SR-GNN	13.38	16.71	19.25	8.95	9.39	9.64
SR-SAN	14.26	17.48	20.99	9.81*	10.24*	10.48*
PASAN	**16.12**	**20.39**	**24.34**	**10.80**	**11.32**	**11.61**
Improv.	11.48%	4.03%	0.37%	10.09%	10.55%	10.78%

Table V. Performance of PASAN in terms of P@1, 3 and MRR@1, 3 on two datasets.

Dataset	P@1	P@3	MRR@1	MRR@3
Reddit	19.47	36.32	19.47	26.85
Xing	7.42	13.28	7.42	10.05

Table VI. Performance of PASAN compared to various ablation models on the Reddit dataset.

Method	P@5	P@10	P@20	MRR@5	MRR@10	MRR@20
PASAN(-P)	40.46	49.85	59.46	26.34	27.60	28.26
PASAN(-H)	35.98	43.43	51.26	26.48	27.47	28.02
PASAN(-P-H)	36.24	43.56	51.13	26.34	27.32	27.84
PASAN	**43.33**	**52.37**	**61.39**	**28.49**	**29.69**	**30.32**
PASAN(Fix)	39.67	48.88	58.12	25.75	26.99	27.63
PASAN(Forward)	42.00	51.19	60.30	27.39	28.62	29.26

Table VII. Performance of PASAN compared to various ablation models on the Xing dataset.

Method	P@5	P@10	P@20	MRR@5	MRR@10	MRR@20
PASAN(-P)	15.70	19.95	23.80	10.54	10.96	11.38
PASAN(-H)	14.62	17.90	21.36	9.98	10.42	10.59
PASAN(-P-H)	14.26	17.48	20.99	9.81	10.24	10.48
PASAN	**16.12**	**20.39**	**24.34**	**10.80**	**11.32**	**11.61**
PASAN(Fix)	15.76	19.91	23.76	10.51	10.98	11.32
PASAN(Forward)	15.86	20.07	23.99	10.67	11.09	11.51

2) The baseline methods only utilize the user's current session data. In contrast, the PASAN model employs both the user's current session, which represents the user's short-term interest, and the user's historical sessions as the user's long-term interest to predict and create personalized recommendations. This contributes to a significant increase in prediction precision and a strengthening of recommendations.

4.3. Effects of Key Components

The proposed PASAN model's superiority over the baseline models presented in the previous section can be attributed to the effect of the model's two key components: reverse-position encoding and historical sessions. This section examines the impact of each of these proposed key components on the performance of the recommendation.

We develop three distinct versions of the proposed model for this purpose. PASAN(-P): PASAN without the use of reverse-position encoding; PASAN(-H): PASAN without consideration of the user's historical sessions and based only on the current session; PASAN(-P-H): PASAN, which lacks reverse-position encoding and historical sessions and is equivalent to SR-SAN. These models are compared with the proposed PASAN model, which incorporates both key components. To investigate the

impact of various position encoding techniques, we replace the reverse-position encoding in the PASAN model with two methods of fixed position encoding [18] and forward position encoding and compare them to PASAN. They include PASAN(Fix): PASAN with fixed position encoding and PASAN(Forward): PASAN with forward position encoding. Tables VI and VII display the performance of various ablation models on the Reddit and Xing datasets, respectively.

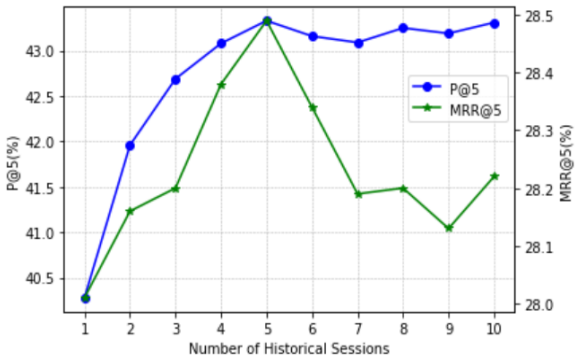
The results indicate that PASAN outperforms all other models, which demonstrates its capability and effectiveness. PASAN(-P) and PASAN(-H) produce results that are inferior to PASAN, and the performance of the PASAN model decreases in both metrics when the reverse-position encoding or historical sessions are removed. This demonstrates that both components of reverse-position encoding and historical sessions are effective and beneficial and enhance the performance of recommendations. In addition, PASAN(-P) achieved improved outcomes than PASAN(-H), indicating that the influence of historical sessions is greater than that of reverse-position embedding. This also highlights the significance of past user interactions in enhancing the performance of the recommendation.

The PASAN(Fix) model performs worse than PASAN(-P) on Reddit, indicating that adding fixed position encoding to our model negatively affected the recommendation performance. However, fixed position encoding did not affect recommendation performance in Xing. The PASAN(Forward) model outperforms the PASAN(-P) model. Besides, the results show that the forward position embedding enhances performance but to a limited degree. The PASAN model with reverse-position embedding has produced superior results compared to the other two versions. Because the length of the sequence of sessions varies, reverse-position encoding can more effectively demonstrate the significance of each item and learn position embedding vectors.

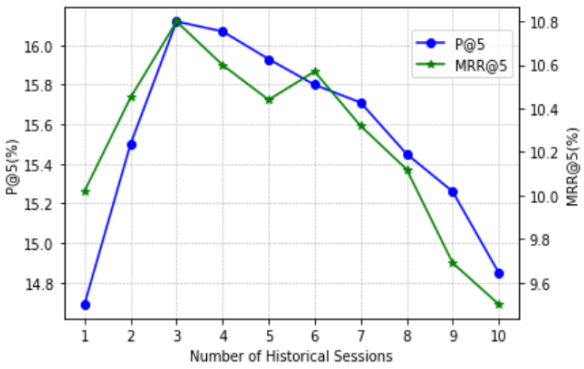
4.4. Effect of the Number of Historical Sessions

This section examines the impact of the number of historical sessions on the performance of PASAN. The range of 1 to 10 is considered for the number of historical sessions. Fig. 2 depicts the outcomes of PASAN with varying numbers of historical sessions based on the P@5 and MRR@5 criteria.

Based on the results, the optimal performance is achieved with five historical sessions for Reddit and three for Xing. Increasing the number of historical sessions initially improves the performance of our model in terms of both criteria. This demonstrates that historical session data is useful for improving recommendation precision and validates the efficacy of our model. However, the model's performance degrades when using excessive historical sessions or considering a longer history of previous user interactions. In fact, considering an excessive number of historical sessions and past user interactions results in noise and information unrelated to the current session and user actions. As a result, it can be concluded that selecting the appropriate number of historical sessions is crucial and that considering an excessive number of historical sessions does not necessarily increase the recommendation's precision.



(a) Reddit data



(b) Xing data

Fig. 2. Performance of PASAN with different numbers of historical sessions concerning P@5 and MRR@5.

4.5. Time Performance Comparison

This section compares the size and runtime of the proposed PASAN model with two baseline models, SR-GNN and SR-SAN, which have demonstrated superior accuracy among baseline models. For all models, the embedding vector dimension is set to 128, and the batch size is set to 100. Table VIII displays the size and number of parameters for the different models on the Reddit data. The runtime is measured on a GPU server (NVIDIA GeForce RTX 3060). The average runtime is recorded for 10 epochs. The runtime per epoch on the Reddit data is presented in Table IX.

Among all models, SR-GNN requires the least memory and the least number of parameters. SR-SAN has faster training and testing times compared to SR-GNN due to the parallel processing capabilities of SANs, which are faster than GNNs. Comparing the proposed PASAN model with the SR-SAN model, it can be concluded that the inverse-position encoding component added to the SANs in the proposed model does not significantly increase the model size and runtime. This indicates that the added key component is computationally efficient.

Table VIII. Size and number of parameters of different models.

Method	Model Size	Number of Parameters
SR-GNN	15.1 MB	3778048
SR-SAN	15.6 MB	3910528
PASAN	15.7 MB	3913088

Table IX. Runtime of each epoch in different models.

Method	Training Time	Testing Time
SR-GNN	5min38s	2min12s
SR-SAN	4min30s	1min45s
PASAN	6min	2min20s

5. Conclusion and Future Work

This article proposes a position-aware SAN model to provide personalized SBR. The proposed model concurrently extracts types of global and sequential dependencies between session interactions. The SAN is used to model the global dependencies among session interactions. The sequential dependence between session interactions is modeled by employing a reverse-position encoding mechanism. Besides, the proposed model incorporates a weighted combination of the current session and the user's recent historical sessions to strengthen the recommendations and extract the user's short- and long-term interests. The proposed model is evaluated using two real datasets. By combining two types of global and sequential dependencies, our model is able to generate more accurate and effective item embedding vectors, as demonstrated by the outcomes of tests and comparisons with other baseline methods. In addition, it achieves high precision in predicting the user's subsequent action and providing recommendations based on information from both the current and historical sessions.

Future work will involve examining and assessing the effectiveness of our proposed model on datasets from other domains. To enhance recommendation accuracy and reduce noise, we propose integrating the user's current session with historical sessions adaptively. This will be achieved by employing an adaptive attention network to learn the importance weights of the sessions.

6. References

[1] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proceedings of the 10th International Conference on World Wide Web*, pp. 285-295, 2001.

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

[3] J. B. Schafer, D. Frankowski, J. Herlocker, and S. Sen, "Collaborative filtering recommender systems," in *The adaptive web: methods and strategies of web personalization: Springer*, pp. 291-324, 2007.

[4] M. Rajabzadeh and R. Rafeh, "Proposing a hybrid recommender system for e-commerce," *Tabriz Journal of Electrical Engineering*, vol. 45, no. 4, pp. 85-91, 2015 (in persian).

[5] F. Tahmasebi, M. Meghdadi, and S. Ahmadian, "A novel hybrid approach based on profile expansion technique to improve cold start problem in recommender systems," *Tabriz Journal of Electrical Engineering,* vol. 48, no. 1, pp. 151-159, 2018 (in persian).

[6] M. Quadrana, A. Karatzoglou, B. Hidasi, and P. Cremonesi, "Personalizing session-based recommendations with hierarchical recurrent neural networks," in *Proceedings of the 11th ACM Conference on Recommender Systems*, pp. 130-137, 2017.

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

[8] M. Ludewig and D. Jannach, "Evaluation of session-based recommendation algorithms," *User Modeling and User-Adapted Interaction*, vol. 28, no. 4-5, pp. 331-390, 2018.

[9] S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme, "Factorizing personalized markov chains for next-basket recommendation," in *Proceedings of the 19th International Conference on World Wide Web*, pp. 811-820, 2010.

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

[11] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk, "Session-based recommendations with recurrent neural networks," in *Proceedings of the 4th International Conference on Learning Representations (ICLR)*, pp. 1-10, 2016.

[12] Y. K. Tan, X. Xu, and Y. Liu, "Improved recurrent neural networks for session-based recommendations," in *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*, pp. 17-22, 2016.

[13] B. Hidasi and A. Karatzoglou, "Recurrent neural networks with top-k gains for session-based recommendations," in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management (CIKM)*, pp. 843-852, 2018.

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

[15] S. Wu, Y. Tang, Y. Zhu, L. Wang, X. Xie, and T. Tan, "Session-based recommendation with graph neural networks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 346-353, 2019.

[16] C. Xu et al., "Graph contextualized self-attention network for session-based recommendation," in *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI-19)*, pp. 3940-3946, 2019.

[17] W. Song, Z. Xiao, Y. Wang, L. Charlin, M. Zhang, and J. Tang, "Session-based social recommendation via dynamic graph attention networks," in *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining (WSDM)*, pp. 555-563, 2019.

[18] A. Vaswani et al., "Attention is all you need," in Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS), pp. 5998-6008, 2017.

[19] J. Fang, "Session-based recommendation with self-attention networks," *arXiv preprint arXiv:2102.01922*, 2021.

[20] P. H. Anh, N. X. Bach, and T. M. Phuong, "Session-based recommendation with self-attention," in *Proceedings of the 10th International Symposium on Information and Communication Technology*, pp. 1-8, 2019.

[21] R. Mehta and K. Rana, "A review on matrix factorization techniques in recommender systems," in

Proceedings of the 2nd International Conference on Communication Systems, Computing and IT Applications (CSCITA), pp. 269-274, 2017.
[22] R. Salakhutdinov and A. Mnih, "Probabilistic matrix factorization," in *Advances in Neural Information Processing Systems*, pp. 1257-1264, 2008.
[23] D. Jannach and M. Ludewig, "When recurrent neural networks meet the neighborhood for session-based recommendation," in *Proceedings of the 11th ACM Conference on Recommender Systems*, pp. 306-310, 2017.
[24] R. He and J. McAuley, "Fusing similarity models with markov chains for sparse sequential recommendation," in *Proceedings of the 16th International Conference on Data Mining (ICDM)*, pp. 191-200, 2016.
[25] S. Wang, L. Hu, Y. Wang, L. Cao, Q. Z. Sheng, and M. Orgun, "Sequential recommender systems: challenges, progress and prospects," in *Proceedings of the International Joint Conferences on Artificial Intelligence (IJCAI'19)*, pp. 6332–6338, 2019.
[26] Q. Liu, Y. Zeng, R. Mokhosi, and H. Zhang, "STAMP: short-term attention/memory priority model for session-based recommendation," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1831-1839, 2018.
[27] M. Wang, P. Ren, L. Mei, Z. Chen, J. Ma, and M. de Rijke, "A collaborative session-based recommendation approach with parallel memory modules," in *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 345-354, 2019.
[28] Z. Zhang and B. Wang, "Learning sequential and general interests via a joint neural model for session-based recommendation," *Neurocomputing*, vol. 415, pp. 165-173, 2020.
[29] C. Kumar and M. Kumar, "Session-based recommendations with sequential context using attention-driven LSTM," *Computers and Electrical Engineering*, vol. 115, p. 109138, 2024.
[30] Z. Wang, W. Wei, G. Cong, X.-L. Li, X.-L. Mao, and M. Qiu, "Global context enhanced graph neural networks for session-based recommendation," in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 169-178, 2020.
[31] F. Yu, Y. Zhu, Q. Liu, S. Wu, L. Wang, and T. Tan, "TAGNN: Target attentive graph neural networks for session-based recommendation," in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1921-1924, 2020.
[32] T. Chen and R. C.-W. Wong, "Handling information loss of graph neural networks for session-based recommendation," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1172-1180, 2020.
[33] Z. Sheng, T. Zhang, Y. Zhang, and S. Gao, "Enhanced graph neural network for session-based recommendation," *Expert Systems with Applications*, vol. 213, p. 118887, 2023.
[34] W. Liu, Z. Zhang, Y. Ding, and B. Wang, "Global heterogeneous graph enhanced category-aware attention network for session-based recommendation," *Expert Systems with Applications*, vol. 243, p. 122907, 2024.
[35] J. Qiao, L. Wang, and L. Duan, "Sequence and graph structure co-awareness via gating mechanism and self-attention for session-based recommendation," *International Journal of Machine Learning and Cybernetics*, vol. 12, no. 9, pp. 2591-2605, 2021.
[36] G. Zhu, H. Hou, J. Chen, C. Yuan, and Y. Huang, "Transition relation aware self-attention for session-based recommendation," *arXiv preprint arXiv:2203.06407*, 2022.
[37] W. Pan and K. Yang, "Enhanced multi-head self-attention graph neural networks for session-based recommendation," *Engineering Letters*, vol. 30, no. 1, 2022.
[38] J. Wang, Q. Xu, J. Lei, C. Lin, and B. Xiao, "PA-GGAN: Session-based recommendation with position-aware gated graph attention network," in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1-6, 2020.
[39] S. Sang, W. Yuan, W. Li, Z. Yang, Z. Zhang, and N. Liu, "Position-aware graph neural network for session-based recommendation," *Knowledge-Based Systems*, vol. 262, p. 110201, 2023.
[40] T. Lu, X. Xiao, Y. Xiao, and J. Wen, "GTPAN: Global target preference attention network for session-based recommendation," *Expert Systems with Applications*, vol. 243, p. 122900, 2024.
[41] M. Ruocco, O. S. L. Skrede, and H. Langseth, "Inter-session modeling for session-based recommendation," in *Proceedings of the 2nd Workshop on Deep Learning for Recommender Systems*, pp. 24-31, 2017.
[42] J. You, Y. Wang, A. Pal, P. Eksombatchai, C. Rosenburg, and J. Leskovec, "Hierarchical temporal convolutional networks for dynamic recommender systems," in *Proceedings of the World Wide Web Conference (WWW'19)*, pp. 2236-2246, 2019.
[43] M. Zhang, S. Wu, M. Gao, X. Jiang, K. Xu, and L. Wang, "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, 2020.
[44] Y. Pang et al., "Heterogeneous global graph neural networks for personalized session-based recommendation," in *Proceedings of the fifteenth ACM International Conference on Web Search and Data Mining (WSDM)*, pp. 775-783, 2022.
[45] H. Wang, H. Bai, J. Huo, and M. Yang, "A graph convolution neural network for user-group aided personalized session-based recommendation," in *Proceedings of the International Conference on Neural Information Processing (ICONIP)*, pp. 332-345, 2023.
[46] H. Vaghari, M. H. Aghdam, and H. Emami, "DIARec: Dynamic intention-aware recommendation with attention-based context-aware item attributes modeling," *Journal of Artificial Intelligence and Soft Computing Research*, vol. 14, no. 2, pp. 171-189, 2024.
[47] A. Ramazani and A. M. Zareh-Bidoki, "Provide a personalized session-based recommender system with self-attention networks," *Iranian Journal of Electrical and Computer Engineering (IJECE)*, vol. 20, no. 3, pp. 236-244, 2022 (in persian).