The Journal of Supercomputing
https://doi.org/10.1007/s11227-024-05897-1

TGIE4REC: enhancing session-based recommendation with transition and global information

Shiwei Gao[1] · Jingyu Wang[1] · Yufeng Zeng[1] · Xiaohui Dong[1]

Accepted: 4 January 2024

Abstract

Predicting the next most likely interactive item based on the current session is the goal of session-based recommendation (SBR). In order to model the adjacent item transition information from previous session sequences and the current session sequences, the most advanced techniques in SBR use graph neural networks and attention mechanisms. Position-aware attention is used to incorporate the reversed position information in an item in order to learn the importance of each item in the session when generating the session representation. However, these methods have certain drawbacks. First, using data from previous sessions always introduces uncorrelated items (noise). Second, learning the sequence transition relations between items in the session sequence is challenging due to reverse position coding. This study presents a novel SBR technique called TGIE4Rec. Specifically, TGIE4Rec learns two levels of session embedding, global information enhanced session embedding and transition information enhanced session embedding. The global information enhanced session representation learning layer employs the information of other sessions and the current session to learn global-level session embedding, and the transition information enhanced session representation learning layer employs the items of the current session to learn new session embedding and integrates the time information into the item representation in the session sequence for neighbor embedding learning, so as to further enhance the sequential transition relations in the session sequence. Experiments on three benchmark datasets have demonstrated that TGIE4Rec is superior to other advanced methods.

Keywords Session-based recommendation · Graph neural network · Representation learning · Attention mechanisms

Extended author information available on the last page of the article

1 Introduction

The issue of information overload is growing in prominence due to the Internet's rapid expansion. Recommendation systems have been widely used to perform tailored information filtering in an effort to reduce the amount of information available on the Web (e.g., P2P lending recommendation [1], social networks [2], sequence recommender systems [3]). The recommendation system's primary function is to ascertain the user's interests and forecast whether or not they would interact with a certain item. In many real-world online applications, like e-commerce, music platforms, and short video platforms, the recommendation system is essential. It has enormous potential for growth in the future. The key to its remarkable success lies in its ability to capture the unique interests of individual users and provide them with personalized recommendations. Conventional recommendation approaches typically rely on the user's personal data and their entire historical activity, enabling a more accurate and tailored recommendation experience [4]. However, in some cases, this information is not accessible. In certain scenarios, accessing user-specific information may not be possible, leaving only the behavior of anonymous users within a short timeframe.

To enhance the recommendation quality for anonymous users, SBR has been proposed [5–7]. SBR is a fundamental recommendation task that has received extensive attention in academia and industry. It focuses on predicting the next click or action based on the sequence of anonymous sessions. This approach allows for improved recommendations by considering the chronological order of an anonymous user's interactions. SBR treats each item feature in a session as a vector, and obtains the final session feature representation by learning the relations between these vectors. Finally, recommendations are made based on these session feature representations. Figure 1 shows a sample session. Suppose a user visits five items on a certain platform in sequence. The task of SBR is to treat each item as an item, while using the previous items as training data to accurately predict the last item. Unlike other traditional recommendation methods that require learning the user's complete and clear historical interactions on a certain platform [4], SBRs only rely on anonymous user action logs (e.g., clicks) from current sessions to predict the user's next action. Its performance depends largely on the transition information between adjacent items in the sequence. For short sequences of sessions, it is difficult to make accurate recommendations only using the information from the current sequence itself.

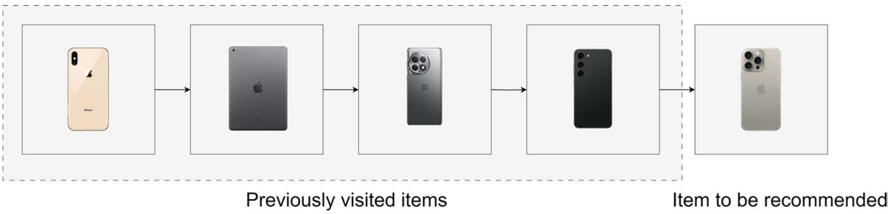


Fig. 1 An example of a session

In this case, several approaches have been proposed to solve the SBR task. Traditional SBR methods employ Markov chain to predict the next item based on the user's previous selected item [8]. In recent years, many methods based on deep learning have been proposed, which use paired item transition information models to simulate user preferences for a given session. GRU4REC [5]first applied RNN to the SBR and employed GRU to capture temporal signals for prediction. NARM [9]combined the attention mechanism and GRU to capture user's interests. STAMP employed MLP and attention to capture the general and current interests of users [10]. DSAN [11]employed a dual-channel attention network to mitigate the impact of unrelated items(noises) in the session sequence. These approaches have yielded encouraging results, but they still face some problems. For example, these methods usually model transition information only for consecutive items and do not capture higher-order transition information in the sequence of sessions.

Recently, due to the appearance and progress of graph neural networks (GNN) [12–18], more and more SBR methods based on graph neural networks have been proposed and achieved remarkable results. SR-GNN [6] first employed graph neural networks for SBR, learning the representation of the entire session by calculating the relative importance of pairwise item transformations between each item and the last item in the session. It focused on the current session when modeling the transition information. In this process, the sparsity of user behavior and the noise data have a great impact on the performance. To alleviate the problem, some methods try to improve the performance of recommendation tasks by employing the cooperation information from other sessions [19–21]. However, some irrelevant information or even noise will inevitably be introduced when utilizing transition information from other sessions, which will harm the model's performance. GCE–GNN [7] has been developed to learn global-level item embeddings by modeling the pairwise item transitions across all sessions. It utilized a combination of reverse position coding and a soft attention mechanism for more comprehensive assessment of each item's contribution in the session sequence. In comparison to previous methods, GCE–GNN demonstrates significant improvements. Among other aggregation methods, the position-aware attention method stood out by incorporating reverse position information into the item embeddings, enabling a more comprehensive understanding of each item's impact in the session sequence and achieving notable performance enhancements. Nevertheless, this method reverses the session sequence, so it cannot well learn the item order transition relations in the session sequence.

Despite the good performance these methods have attained, there are still certain issues. First, using item transition information from previous sessions can improve the inference of user preferences and the item representation in the current session sequence, but it can also introduce noise or unnecessary items that impair the performance of the model. Second, a more natural way to measure each item's contribution is to employ reverse position encoding, which aggregates item representations by flipping the session sequence. It does, however, destroy the relations of sequential transition between the objects in the session sequence. Furthermore, the majority of the current techniques concentrate on information aggregation from the standpoint of spatial structural data. In the session sequence, the temporal information of neighboring nodes is disregarded.

To address these issues, we have proposed a novel approach called enhancing session-based recommendation with transition and global information (TGIE4Rec). By combining the prediction results of the two levels of session embedding to make recommendations, the model can not only make use of the global information but also weaken the noise problem caused by the introduction of global information. In TGIE4Rec, we learn two levels of session embedding, global information enhanced session embedding (GIE) and transition information enhanced session embedding (TIE). GIE captures richer session information and improves model performance by combining reversed position information with global information. However, in order to better measure the contribution of the items in the session sequence, the GIE employs the position-aware attention method, resulting in insufficient learning of the sequential transition relations of the session sequence.

Therefore, we design a TIE to enhance the learning of sequential transition relations of session sequences. The session representation learning layer only uses the items in the current session sequence and their transition relations to learn a session embedding, it integrates the time information into the item representations in the session sequence for neighbor embedding learning, further enhancing the model's ability to learn the sequential transition relations of items. We then use the two session embeddings to make separate predictions, and finally, we sum the two prediction scores to get the final prediction. Since the session representation learning layer only uses the information of the current session to generate session embedding and does not introduce irrelevant items (noise) caused by other session sequences. When using other session information is not related items to the recommendation, combined predictive results of the two sessions' embedded joint recommendation can weaken the global information's noise to enhance session representation learning layer, thus improve the recommendation performance of the model. The main contributions of this work are as follows:

(1) Our method not only leverages the benefits of reverse position coding, which allows for a more comprehensive measurement of each item's contribution in the session sequence but also addresses the issue of insensitivity to the sequential transition relations between items.
(2) We propose a novel method to use both sequential-level session embeddings and global-level session embeddings for learning, thereby reducing the impact of noise introduced by global information on model performance. At the same time, an ingenious aggregation method is used to aggregate two levels of session information.
(3) We embed the temporal information of each item in the session sequence into the item representation for neighbor embedding learning and enhance the sequential transition information in the session sequence.
(4) We conduct extensive experiments on three public datasets, and the experimental results show that our model outperforms state-of-the-art methods.

The rest of this paper are organized as follows: In Sect. 2, we introduce the related work research of the SBR method. Section 3 covers the basics of SBR and how to construct session graphs from the session sequence. The detailed description of our proposed model TGIE4Rec is presented in Sect. 4. In Sect. 5, we introduce the relevant datasets and experimental results. At last, we summarize the main work of this paper in Sect. 6.

2 Related work

In this section, we review some related work on SBRs. We first discuss the traditional Markov chain-based approach, then we introduce the deep learning-based approach and finally we introduce the latest graph neural network-based approach.

2.1 Methods based on Markov chain

Since a session is a sequence of items clicked in chronological order, Markov chain-based methods are proposed in SBR. It map the current session to a Markov chain and then infer the user's next action based on the previous action. Rendle et al. [8] proposed FPMC to model the local sequential behaviors and long-term user preferences by combine matrix factorization and first-order Markov chains. It can accommodate the SBR by ignoring the user latent representation because the user latent representation is not available for the anonymous session. However, FPMC focus on modeling the sequential transition of two adjacency items, ignoring the long-term dependencies.

2.2 Methods based on deep learning

These deep learning-based methods [5, 9–11, 22–24] are mainly based on recurrent neural networks and attention mechanisms. GRU4REC [5] was the first to apply RNN to the SBR and employ GRU to capture temporal signals for prediction. However, methods based on recurrent neural networks have a strong assumption of order, which means that they cannot capture transition relations between items that are far away. NARM [9] adopted a hybrid encoder that uses the attention mechanism and RNN to capture the user's primary purpose. To alleviate the deviation of time series, STAMP [10] proposed a short-term attention/priority model, the model proposed a novel attention mechanism to capture the user's interest, rather than using RNN. DSAN [11] employed a dual-channel attention network to mitigate the impact of unrelated items(noises) in the session sequence. Compared with traditional methods, these deep learning-based methods are significantly improved, which implies that deep learning methods have stronger representation ability.

2.3 Methods based on graph neural network

Graph neural networks have attracted more and more attention in SBRs because of their ability to model complex relations between items. These methods transform session sequences into graphs and use graph neural networks to make predictions. SR-GNN [6] applied graph neural network to SBR task for the first time and achieved significant performance improvement compared with deep learning-based methods. It transformed the session sequence into an unweighted directed graph and applied the gated graph neural network (GGNN) [12] to capture complex item transitions in the session. FGNN [25] defined SBR as a graph classification problem. It employed a graph weighted attention network to obtain embeddings of items. GC-SAN [26] learned local dependencies among items through the gated graph neural network and employed self-attention network to extract long-term dependencies among items. SGNN-HN [27] designed a new session graph called star graph to capture the non-adjacent item transition relations. GCE–GNN [7] learned two levels of item representation from the session graph and the global graph, and it employed reverse position coding and soft attention mechanism to aggregate the items in the session sequence. TRASA [28] employed GRU to model item transition relations and adopted graph neural networks to learn item representations. GC-HGNN [29] employed hyper-graph convolution neural network and graph attention network to obtain global context information and local information, and employed attention mechanism to learn the final representation of session sequence. S2-DHCN [30] introduced self-supervised learning into SBR, where it constructed two different levels of views without using random dropout, and compared the representations of the two views through contrastive learning. COTREC [31] proposed a co-training framework based on self-supervised learning that preserves the entire session information and provides significant improvement. MCGNN [32] performs item prediction and category prediction by capturing multi-level information in sessions to build a multi-task learning framework.

3 Preliminaries

In this section, we first introduce the problem definition based on SBR. Then, we show how to convert the session sequences into session graph and global graph, and describe in detail how to construct the session graph and global graph.

3.1 Problem definition

SBRs focus on providing anonymous users with the next recommendation for a short sequence of clicks. Therefore, it requires the model to accurately capture the user's interest by using short sessions rather than a full history of interactions.

Let $V = \{v_1, v_2, ..., v_m\}$ and $S = \{s^1, s^2, ..., s^n\}$ represent the set of items and the whole session sequences, respectively. V is made up of all the unique items in S. Each session sequence $s^i = [v^i_1, v^i_2, ..., v^i_l]$ordered by timestamps.$v^i_j \in V$ represents the j-th clicked item in session i, and l is the length of the current session sequence. The goal of the SBR is to predict the next click $v_{l+1}^{i}$for the session s^{i}. SBR model first generates item embeddings of session sequences. These item embeddings are then further learned through a graph neural network to obtain the final session embeddings. Finally, candidate items are obtained by calculating the scores of the items.

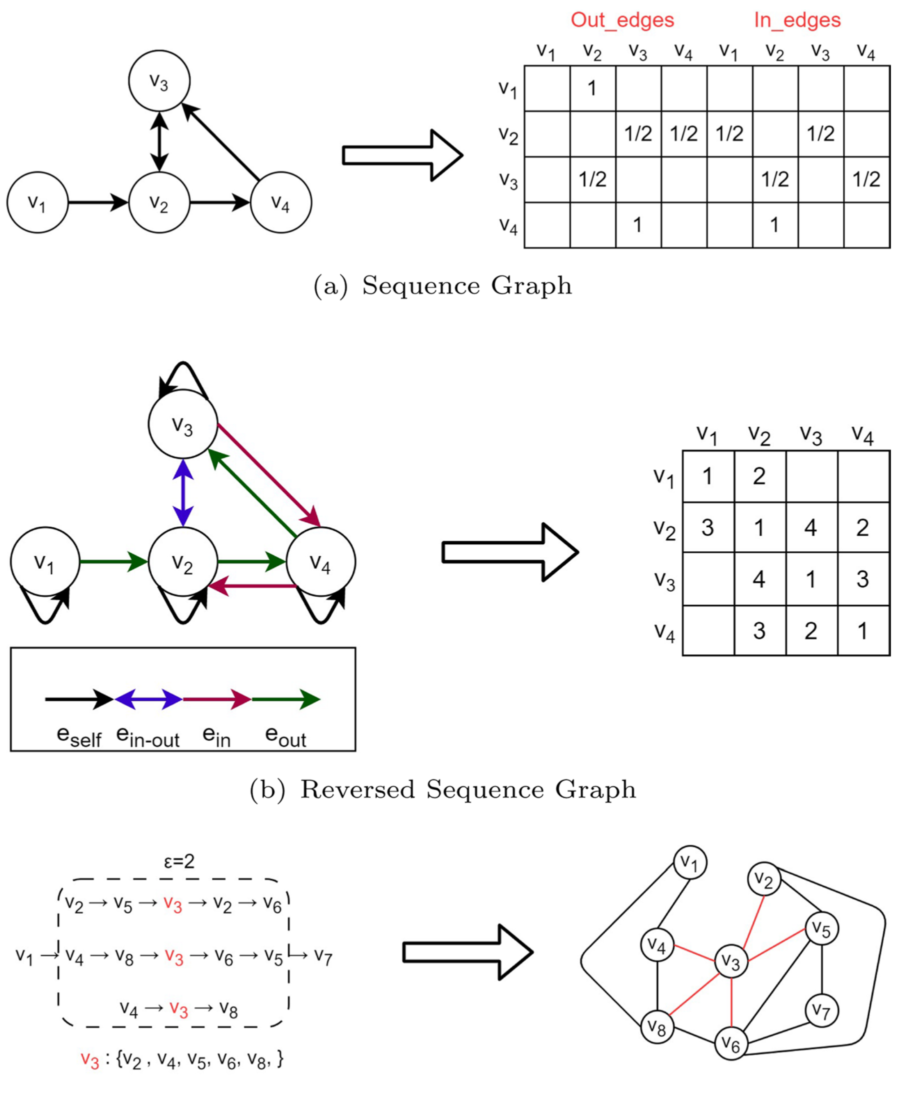
3.2 Constructing graphs

3.2.1 Constructing sequence graph

In order to get the items' sequential information in a session, we need to construct a sequence graph which contains the item transition sequence. For a sequence of sessions $S = \left\{v_1^s, v_2^s, \ldots, v_m^s\right\}$, we can model it as a directed graph $G_s = \left(V_s, E_s\right)$, where$V_s$ represents the set of nodes in session S and E_s denotes the edge set,$\left(v_i^s, v_{i+1}^s\right)$ represents two adjacent items in the session. For a sequence graph, we divide an edge into an input edge and an output edge and assign a normalized weight. The normalized weight is set as the value of the occurrences of the edge divided by the out degree of the edges start node. In this way, we can learn the propagation information of the previous item and the propagation information of the next item. An example of the construction of a sequence graph is shown in Fig. 2a.

3.2.2 Constructing reversed sequence graph

Since in a session sequence, the last item often contains more valid information, we can more accurately represent the importance of each item by reversing the position information. Let's take session $\left[v_1, v_2, v_4, v_3, v_2\right]$ as an example, introduce how to model each session sequence as two directed graphs: the reverse sequence graph and the sequence graph. For the reverse sequence graph, we first flip the sequence to become $\left[v_2, v_3, v_4, v_2, v_1\right]$. Then, we add a self-connection edge to each item and divide the edges into four types: $e_{\mathrm{in}}, e_{\mathrm{out}}, e_{\mathrm{in-out}}, e_{\mathrm{self}}$. For edge $\left(v_i^s, v_j^s\right)$, e_{in} indicates there is only transition from v_i^s to v_j^s, e_{out} implies there is only transition from v_j^s to v_i^s, and e_{in-out} reveals the transition between v_i^s and $v_i^s$is bidirectional;e_{self} represents a self-connection edge. The construction of the reverse sequence graph is shown in Fig. 2b, where different numbers in the matrix represent different types of edges. The number 1 represents a self-loop edge(v_1 to v_1, e_{self}), 2 represents an out-degree edge(v_1 to v_2, e_{out}), 3 represents an in-degree edge(v_1 to v_2, e_{in}), and 4 represents a bidirectional edge(v_2 to v_3 and v_3 to v_2, $e_{\mathrm{in-out}}$).



(a) Sequence Graph

(b) Reversed Sequence Graph

(c) Global Graph

Fig. 2 Illustrates the construction of the session graph and the global graph

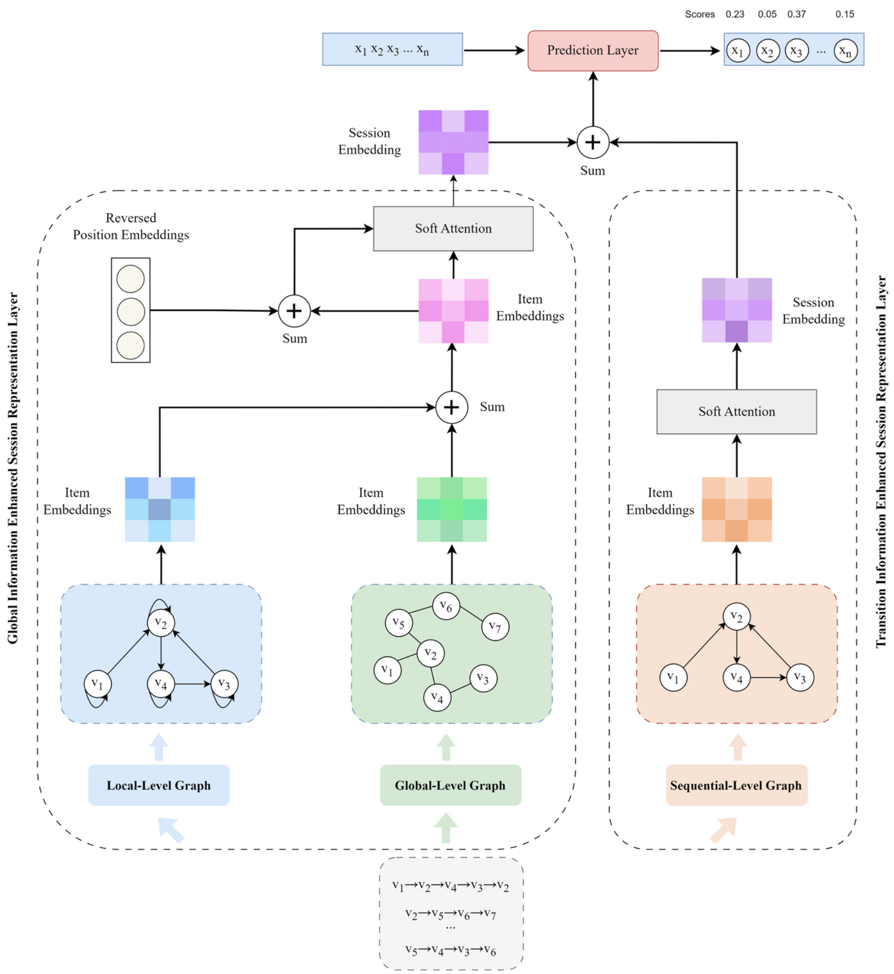


Fig. 3 Overview of the TGIE4Rec model. The overall model is divided into two parts: the global information enhanced session representation layer (GIE) and the transformation information enhanced representation layer (TIE)

3.2.3 Constructing global graph

Since the session graph can only extract the item transition information within the current session. In order to better extract the item transition relations, we construct a global graph to extract global information from the items in all sessions, so as to improve the generalization of the model. Let $G_g = \left(V_g, \varepsilon_g\right)$ represent the nodes that make up the global graph are all items in V, ε_g is the edges constituted by the adjacent items in all session sequences. We use the frequency of the neighbor items in all sessions as the weight of the corresponding edge. For efficiency reasons, we keep only the N edges with the highest edge weight for each item. For a more detailed description of global graph construction, refer to the description of global graphs in [7]. An example of the construction of global graph is shown in Fig. 2c.

4 The proposed method

In this section, we introduce our novel model Enhancing Session-based Recommendation with Transition and Global Information (TGIE4Rec). The workflow is shown in Fig. 3. First of all, we employ global information enhanced session represents learning layer and transition information enhanced session represents learning layer to learn two different levels of session embedding. Global information enhanced session represents learning layer employs other session information and the reverse sequence of the current session to learn session-level items embedding and global-level items embedding, it aggregates the item representation into a global-level session embedding through the reverse position encoding and soft attention mechanism. The transition information enhanced session learning layer only employs the current session sequence to learn the local-level session embedding to supplement the learning of the sequential transition relations of the session sequence ignored by the GIE and integrates the time information into the item representation in the session sequence for neighbor embedding learning. Then, we use these two session representations to predict respectively and obtain their corresponding probabilities. Finally, we sum the two scores to get the final prediction result.

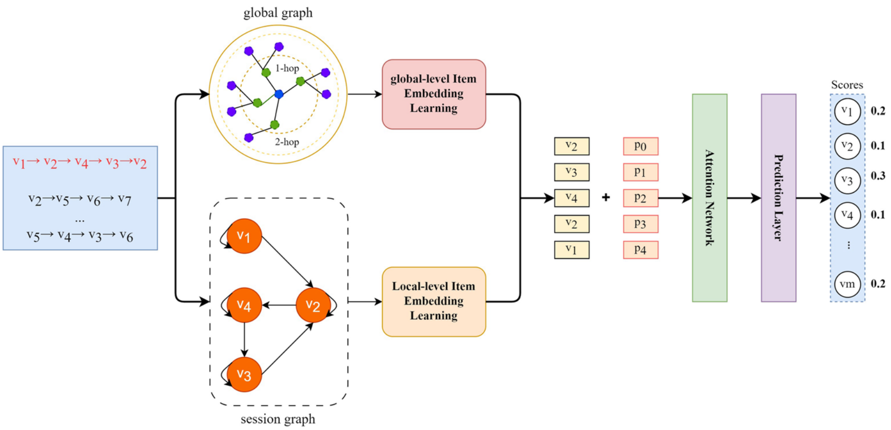


Fig. 4 Global information enhanced session representation layer. First, we construct session sequences as local-level graphs and global-level graphs, and learn their item embeddings. The item embeddings are then combined with reversed position encoding, and the final session embeddings are obtained through the attention network for prediction

4.1 Global information enhanced session representation learning layer (GIE)

In this section, we focus on the GIE, which is the left part in Fig. 3. The workflow is shown in Fig. 4. We employ session graph and global graph to learn session-level item representations and global-level item representations respectively. And aggregate the item representations learned from the two levels to form session representations through reverse position coding and soft attention mechanism.

4.1.1 Session-level item representation learning

Before learning item representation, all items in V need to be encoded into a unified embedding space R^d, where d is the size of the embedding dimension. We embed session S and item into the same space. We use graph attention network to learn session-level item representation and employ attention mechanism to aggregate neighbors to learn the importance of different neighbors. The attention coefficients between v_i^s and v_j^s is calculated by the element-wise product and nonlinear transformation. For different relations, different weight vectors are trained, namely a_{in}, a_{out}, $a_{\text{in-out}}$, a_{self}. To make the coefficients between different nodes comparable, we normalized the weights by the Softmax function:

$$\alpha_{ij} = \frac{exp\Big(\text{ LeakyRelu }\Big(a_{r_{ij}}^T\Big(h_{v_i} \odot\ h_{v_j}\Big)\Big)\Big)}{\sum_{v_k \in N_{v_i}^s} exp\Big(\text{LeakyRelu}\Big(a_{r_{ij}}^T\big(h_{v_i} \odot\ h_{v_k}\big)\Big)\Big)}. \tag{1}$$

In formula (1), $a_{r_{ij}}^T \in R^d$ is the weight vector of the relations between v_i^s and v_j^s, and employ LeakyReLu as activation function, $\odot$ indicates element-wise product. $N_{v_i}^s$ is the first-order neighbor of item v_i.

Next, we use a linear combination of features based on the previously calculated attention score to get the output features for each node:

$$h_{v_i}^s = \sum_{v_j \in N_{v_i}^s} \alpha_{ij} h_{v_j}. \tag{2}$$

The item representations in session sequence are obtained by aggregating the features of item itself and its neighbors through graph attention mechanism.

4.1.2 Global-level item representation learning

In this section, we describe how to utilize item transition information from other sessions to help with recommendations. We generate attention weights based on the importance of each connection to distinguish the importance of different neighbors of the current item. We first describe the single layer that consists of two components: information propagation and information aggregation, and then show how to extend the single layers to multiple layers.

Information propagation An item may be involved in multiple sessions, and we get richer item transition information from other session sequences to help predict the interest of the current session. Since different neighbor items have different importance, we consider using session-aware attention scores to distinguish the importance of the current item's neighbors $\mathbf{N}_{\boldsymbol{\varepsilon}}(\mathbf{v})$. As a result, the current item's neighbors$\mathbf{N}_{\boldsymbol{\varepsilon}}(\mathbf{v})$ are linearly combined according to the session-aware attention coefficient.

$$h_{N_{v_i}^g} = \sum_{v_j \in N_{v_i}^g} \pi\left(v_i, v_j\right) h_{v_j}, \tag{3}$$

where $\pi\left(v_i, v_j\right)$ represents the importance weight of different neighbors, the Softmax function is used to normalize them. Intuitively, the closer an item is to the preferences of the current session, the closer the item is to the interests of the user. Therefore, we implement $\pi\left(v_i, v_j\right)$ as follows:

$$\pi\left(v_i, v_j\right) = q_1^T \mathrm{LeakyRelu}\left(W_1\left[\left(s \odot h_{v_j}\right) \| w_{ij}\right]\right) \tag{4}$$

$$\pi\left(v_i, v_j\right) = \frac{\exp\left(\pi\left(v_i, v_j\right)\right)}{\sum_{v_k \in N_{v_i}^g} \exp\left(\pi\left(v_i, v_k\right)\right)}, \tag{5}$$

LeakyRelu is selected as the activation function, || indicates concatenation operation, $\odot$ indicates element-wise product,$w_{ij} \in R^1$ is the weight of edge $\left(v_i, v_j\right)$ in the global graph, $W_1 \in R^{d+1 \times d+1}$ and $q_1 \in R^{d+1}$ are trainable parameters. s is the average of item representations of the current session,

$$s = \frac{1}{|S|} \sum_{v_i \in S} h_{v_i}. \tag{6}$$

The mean value of the current session's item representations is used to calculate the importance weight of related items, which means neighbors that match the preference of current session sequence will be more important.

Information aggregation Finally, we need to aggregate the item representation h_v and its neighbors $h_{N_v^g}$. We implement the aggregation function as follows:

$$h_v^g = \mathrm{Relu}\left(W_2\left[h_v \| h_{N_v^g}\right]\right), \tag{7}$$

where we employ Relu as the activation function, $W_2 \in R^{d \times 2d}$ is the trainable parameter, || represents the concatenation operation.

After the single layer of information propagation and aggregation, we aggregated the representations of the item itself and its neighbors to form a new item representation. We could also scale from one layer to multiple layers to capture high-order connection information. This combines more useful information from the other sessions into the item representation. At step k, the embedding of an item is represented as:

$$h_v^{g(k)} = \mathrm{agg}\Big(W_3\Big[h_v^{(k-1)}, h_{N_v^g}^{(k-1)}\Big]\Big), \tag{8}$$

where $h_v^{(k-1)}$ is representation of item v which is generated from previous information propagation steps, h_v^0 is set as h_v at the initial representation. Therefore, the k-order representation of an item is the aggregation of its initial representation and its k-order neighbors, so that the high-order information can be aggregated into the current session.

4.1.3 Generate fusion session embeddings

For each item in the session sequence, its representation is obtained by combining the global context and the session context. We add the global-level item representation to the session-level item representation to get the final item representation. It is computed by sum pooling:

$$h_v^{g,(k)} = \mathrm{dropout}\big(h_v^{g,(k)}\big) \tag{9}$$

$$h_v^s = \mathrm{dropout}\big(h_v^s\big) \tag{10}$$

$$h_v = h_v^g + h_v^s, \tag{11}$$

where we employ dropout [33] on global-level and session-level representations to prevent overfitting.

We now present how to obtain a session representation based on the item representation learned above. After feeding a session sequence into graph neural networks, we can obtain the representation of the items involved in the session, i.e., $H_s = \{h_1^s, h_2^s, ..., h_l^s\}$. l is the length of the session sequence. We employ reversed position coding to retain the position information of the session. Intuitively, the closer an item is to the last item, the more likely it is to represent the user's current interest. Positional embedding corresponds to the learnable positional embedding matrix $p_s^g = \{p_1^g, p_2^g, ..., p_l^g\}$, which is added to the item representation to obtain the final item representation. Specifically, we concatenate the position embedding matrix with the item representation in the session sequence, and then use the feature transformation and the nonlinear activation function tanh to get the final coefficient:

$$z_i = \tanh\big(W_3\big[h_i^s \| p_{l-i+1}^g\big] + b\big), \tag{12}$$

where $W_3 \in R^{d\times 2d}$ and $b \in R^d$ are trainable parameters, $\|$ represents concatenation operations. Most of the previous methods focused on the importance of the last item in the session, which indirectly affected the contribution of other items to the current item. We get the session information by averaging the item represents in the session sequence,

$$\bar{s} = \frac{1}{l}\sum_{i=1}^{l} h_i^s. \tag{13}$$

Then, we use a soft attention mechanism to calculate the weights,

$$\beta_i = q_2^T \sigma\left(W_4 z_i + W_5 s\right), \tag{14}$$

where $W_4, W_5 \in R^{d\times d}$ and $q_2 \in R^d$ are trainable parameters. Finally, the session embeddings are formed from a linear join represented by items,

$$S_g = \sum_{i=1}^{L} \beta_i h_i^s, \tag{15}$$

where session embeddings S_g is composed of all the items representation in the current session sequence, and the contribution of each item is determined by the information in the current session and the order in the session sequence.

4.2 Transition information enhanced session representation learning layer(TIE)

Next, we describe how to employ gated graph neural networks (GGNN) to learn item transitions relations of session sequences and how to generate session embedding, which is the right part in Fig. 3. Follow the previous study [34, 35], before using GGNN for item representation learning, we employed position coding to embed temporal information into the item representation in the session sequence to enhance the sequential transition relations of the session sequence. The position embedding corresponds to a learnable position embedding matrix $p_s^t = \left\{p_1^t, p_2^t, ..., p_l^t\right\}$, l is the length of the current session sequence.

Unlike the previous Sect. 4.1.3, which employed reverse position coding to obtain an attention coefficient for the generation of session embedding, we add position embedding to the initial embedding of items for neighbor aggregation to generate item representations. For a session sequence $S = \left\{v_1^s, v_2^s, ..., v_m^s\right\}$, its initial embedding in the TIE is $C^t = \left\{c_1^s, c_2^s, ..., c_m^s\right\}$, c_i^s is composed of v_i^s and p_i^t. The workflow of TIE is shown in Fig. 5.

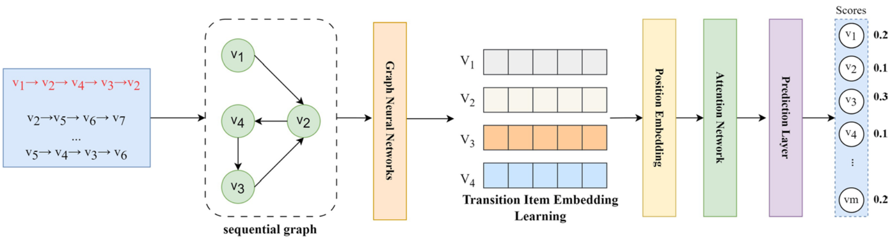


Fig. 5 Transition information enhanced session representation layer. After constructing the sequence graph, the corresponding transformation item embeddings are learned through a graph neural network. Then by performing position embedding, the attention network obtains the final session embedding for prediction

4.2.1 Item representation learning

Gated graph neural network (GGNN) has been widely used in SBR and achieved remarkable results. Therefore, we employ GGNN to learn item representations in the TIE. For each session graph G_s, the gating graph neural network processes the nodes simultaneously. Under the constraint of matrix A_s, Eq. (16) is used to disseminate information among different nodes. Specifically, it extracts the latent vectors of the neighbor nodes and takes them as input to the gated graph neural network. For the node v_i on the graph, the information propagation between item nodes is formalized as:

$$a_{s,i}^{t} = A_{s,i:}\left[c_1^{(t-1)}, \cdots, c_m^{(t-1)}\right]^T H + b, \tag{16}$$

where $H \in R^{d\times 2d}$ is the weight coefficient, $\left[c_1^{(t-1)}, \cdots, c_m^{(t-1)}\right]$ is the list of node vectors in session S^t after added position embedding, $A_s \in R^{n\times 2n}$ is the adjacency matrix of session graph, $A_{s,i:} \in R^{1\times 2n}$ are the two columns of blocks in matrix A_s corresponding to node $v_{s,i}$.

Then, the update gate and reset gate decide what information to be preserved and discarded respectively. The update gate is used to control the extent to which the state information from the previous moment is brought into the current state. Reset gate determine how new information can be combined with previous information to capture short-term dependencies in the session sequence.

After that, we use reset gate to construct candidate states as described in Eq. (19). Finally, the update gate is used to combine the previous hidden with the candidate state to get the final state. The update functions are formalized as:

$$z_{s,i}^{t} = \sigma\left(W_z a_{s,i}^{t} + U_z c_i^{(t-1)}\right) \tag{17}$$

$$r_{s,i}^{t} = \sigma\left(W_r a_{s,i}^{t} + U_r c_i^{(t-1)}\right) \tag{18}$$

$$\tilde{c}_i^{t} = \tanh\left(W_o a_{s,i}^{t} + U_o\left(r_{s,i}^{t} \odot c_i^{(t-1)}\right)\right) \tag{19}$$

$$c_i^{t} = \left(1 - z_{s,i}^{t}\right) \odot c_i^{(t-1)} + z_{s,i}^{t} \odot \tilde{c}_i^{t}, \tag{20}$$

where W_* and U_* are trainable parameters, $\odot$ indicates element-wise product, σ is the Sigmoid activation function. $z_{s,i}$ and $r_{s,i}$ are update gate and reset gate, which control what representation vector will be discarded or preserved.

4.2.2 Generate local session embeddings

Different from the previous methods which employ the last item of the session sequence as the user's current interest and employ the soft attention mechanism to generate session embedding. We generate the session embedding by averaging the item represents in the session sequence,

$$S_t = \frac{1}{l}\sum_{i=1}^{l} c_i^s. \tag{21}$$

4.3 Prediction layer

Based on the obtained session representations S_g and S_t, we first employ the dot product based on the initial embedding and current session representation of each candidate, and apply the Softmax function to get the output $\hat{y}$:

$$\hat{y_g} = \mathrm{Softmax}\left(S_g^T v_i\right) \tag{22}$$

$$\hat{y}_t = \mathrm{Softmax}\left(S_t^T v_i\right), \tag{23}$$

where $\hat{y}_i \in \hat{y}$ denotes the probability of item v_i appearing as the next click in the current session, $\hat{y}$ is the probability of all items in the next click in the current session. By the prediction layer, we can get two probability vectors $\hat{y_g}$ and $\hat{y}_t$, and then use the hyper-parameter k to add two probabilities:

$$\hat{y} = k * \hat{y_g} + (1 - k) * \hat{y}_t. \tag{24}$$

The loss function is defined as the cross-entropy of the prediction result $\hat{y}$:

$$L(\hat{y}) = -\sum_{i=1}^{m} y_i \log \hat{y}_i + \left(1 - y_i\right)\log\left(\left(1 - \hat{y}_i\right)\right), \tag{25}$$

where y denotes the one-hot encoding vector of the ground truth item. Finally, the whole training procedure of TGIE4Rec is summarized in Algorithm 1.

Algorithm 1 Training Process of TGIE4Rec

Input: Sessions S, Item Embeddings Vs, Embedding_size d, Parameter K
Output: Top-k recommendation items

```
Construct sequential-level graph and global-level graph;
for epoch in range(epoches) do
    for batch in DataLoader do
        for each session s in batch do
            //GIE;
            Compute the coefficients and features between different nodes using Eq. (1)-(2);
            Information propagation & Information aggregation using Eq. (3)-(8);
            Generate global information enhanced session embedding S_g using Eq. (9)-(15);
            //TIE;
            for l in range (L) do
                c_i^t= GGNN (initial embedding,c_{s,i} ) based on Eq. (16)-(20);
            end for
            Generate transition information enhanced session embedding S_t using Eq. (21);
            Session embedding Fusion S_h using Eq. (22)-(24);
            Next-item Prediction loss L using Eq. (25);
        end for
    end for
    Using Adam optimize loss L
end for
```

5 Experiments

We have conducted some experiments to evaluate the accuracy of the proposed TGIE4Rec method by answering the following four key questions:

RQ1: Does TGIE4Rec outperform other SBR baselines in real datasets?

Table 1 The statistical results of the processed data set

Dataset	Diginetica	Tmall	Nowplaying
# click	982,961	818,479	1,367,963
# train	719,470	351,268	825,304
# test	60,858	25,898	89,824
# items	43,097	40,728	60,417
avg.len	5.12	6.69	7.42

RQ2: How does each component of TGIE4Rec affect the performance?

RQ3: What is the impact of different aggregation methods on model performance?

RQ4: How do different hyper-parameter settings affect the model performance?

5.1 Datasets and preprocessing

We use three public datasets: Diginetica, Tmall and Nowplaying to verify the performance of our model. The Diginetica datasets from CIKM Cup 2016 consist of a typical transaction data. Tmall datasets from IJCAI 2015 competition, which contain the anonymous user shopping log on Tmall online shopping platform. The Nowplaying datasets come from [36] and describe the user's music listening behavior.

Our processing of the datasets is consistent with previous work [6, 37, 38]. Specifically, sessions of length one and items that appear less than five times are filtered in all three datasets. In addition, for a session sequences, we divided the process into the sequence and the corresponding labels, i.e., $S = \left\{v_{s,1}, v_{s,2}, \ldots, v_{s,n}\right\}\left(\left[v_{s,1}\right]v_{s,2}\right)\left(\left[v_{s,1}v_{s,2}\right]v_{s,3}\right),\left(\left[v_{s,1}v_{s,2\ldots}v_{s,n-1}\right]v_{s,n}\right)$. The statistical results of the processed data set are shown in Table 1.

5.2 Evaluation metric

P@20 (Precision) [14] is widely used as a measure of forecast accuracy. It represents the proportion of correctly recommended items among the top-20 items.

MRR@20(Mean Reciprocal Rank) [14] is the average reciprocal ranking of correct recommended items. When the rank exceeds 20, the reciprocal rank is set to 0. The MRR metric considers the order of the recommendations sort, with the larger MRR value indicates the correct recommendation in front of the sorted list.

5.3 Baseline algorithm

To evaluate the performance of the proposed model, we compare it with the following representative nine baselines:

FPMC [8]: This method combined matrix factorization and first-order Markov chains to capture sequence effects and user preferences. According to the previous work, we also ignored potential user representations when calculating recommendation ratings.

GRU4REC [5]: This is a deep learning model based on recurrent neural networks. It stacked multiple GRU layers to model sessions and designed a loss function to help RNN models adapt to SBR problems.

Table 2 Hyper-parameter settings

Dataset	Diginetica	Tmall	Nowplaying
Embedding_size(d)	100	300	128
Dropout_gcn	0.2	0.6	0
Dropout_local	0	0.3	0
K	0.6	0.1	0.3

Table 3 Performance comparison of the baseline model

	Diginetica		Tmall		Nowplaying	
Model	P@20	MRR@20	P@20	MRR@20	P@20	MRR@20
FPMC	22.31	6.95	9.15	3.31	7.36	2.82
GRU4REC	30.85	8.32	10.93	5.89	7.92	4.48
NARM	48.32	16.03	23.30	10.70	18.59	6.93
STAMP	45.98	14.52	26.47	13.36	17.66	6.88
SR-GNN	51.30	17.80	27.57	13.72	18.87	7.47
GC-SAN	49.11	16.73	21.80	10.17	18.85	7.43
FGNN	51.36	18.47	25.36	10.39	18.78	7.15
E-GNN	51.28	18.34	25.34	10.27	18.67	7.11
CT-GNN	52.71	18.56	31.42	15.50	22.28	8.06
GCE-GNN	54.22	19.04	35.09	15.80	22.43	8.40
GCL	54.26	19.12	35.73	16.21	22.59	8.32
GC-HGNN	54.10	18.64	36.83	17.37	**23.65**	7.83
TGIE4Rec	**54.31**	**19.21**	**37.49**	**17.50**	22.81	**8.81**

The bold indicates the best result

NARM [9]: It introduced the attention mechanisms to capture the user's primary purpose. The main purpose is to combine with continuous behavioral features and use them to generate the final representation of the next item.

STAMP [10]: STAMP employed the multi-layer perceptron (MLP) enhanced by attention mechanism to capture the general interests of users and the current interests of the current session.

SR-GNN [6]: It was the first method to employ graph neural networks for SBR, which introduced gated graph neural networks (GGNN) to capture complex item transitions, and employed the attention mechanism to capture the user's general and current interest.

GC-SAN [26]: It combined graph neural network and multi-layer self-attention network to improve recommendation performance by modeling local neighborhood item transition and extracted the long-range dependencies.

FGNN [25]: It proposed the weighted attention layer and a graph-level feature extractor to learn item embedding and session embedding.

E-GNN [39]: E-GNN reconstructed the local conversation graph by designing multiple interaction patterns, and then uses a new fusion algorithm to fuse conversation information.

CT-GNN [40]: CTGNN combined global information and time-related information to capture user's interests.

GCE-GNN [7]: It employed graph attention networks to capture item transition relations from local and global contexts and employed reverse position coding and soft attention mechanism to generate session representations.

CGL [41]: It combined training of self-supervised signals with supervised signals to alleviate the data sparsity problem.

GC-HGNN [29]: It employed hyper-graph convolution neural network and graph attention network to obtain global context information and local information, and employed attention mechanism to process the final representation of fusion feature learning session sequence.

5.4 Parameter settings

In order to make a fair comparison, we adopt the data preprocessing methods and parameter settings provided in the respective papers to obtain the best performance. Referring to the previous method [7], we initialize all parameters using Gaussian distribution with mean of 0 and standard deviation of 0.1. The L_2 regularization as penalty term with a value of 10^{-5}, and choose Adam with initial learning rate of 0.001, which will decay by 0.1 after every 3 epochs to optimize parameters. The epoch is set to 30. And the stopping criteria remain consistent within the identical dataset.

In addition, for the setting of global graph, we set the number of neighbors and the maximum distance between neighboring items to 12 and 1, respectively. Some hyper-parameters that have a significant impact on the model need to be adjusted separately in each dataset. Table 2 shows the details of hyper-parameter settings.

5.5 Full comparison (RQ1)

The performance comparison of each model is shown in Table 3. It can be seen that our model achieves the best performance on all three datasets, which proves the effectiveness of our model.

The traditional Markov chain-based FPMC method has the worst performance among the SBR methods shown in Table 3, which shows that the assumption of independence of continuous terms in the Markov chain-based method is not enough in SBR. Compared with the methods based on deep learning and graph neural network, the competitiveness of traditional methods is insufficient.

For deep learning-based methods, GRU4REC made predictions by simply stacking multiple gated loop cells, so it performs worse than other deep learning-based methods. NARM introduced the attention mechanisms to capture the user's primary purpose, and then combines them into a unified session representation to achieve better performance. STAMP explicitly considered users' general and current interests by using an attention mechanism, which has advantages and disadvantages compared with NARM. Compared with traditional methods, the recommendation performance of these models is significantly improved, which indicates that the methods based on deep learning have a more powerful representation ability.

With the development of graph neural networks, SBR also began to use graph neural network to model the transition relations of session sequences and achieved good results. SR-GNN is the first attempt to apply gated graph neural networks to SBR and achieves better performance than deep learning-based methods. GC-SAN dynamically constructed the graph structure of session sequences and captures rich local dependencies, and each session learns long-term dependencies through a self-attention mechanism. Finally, each session is represented as a linear combination of the global preference and the current interest. FGNN proposed a weighted attention layer and a graph-level feature extractor to learn item embedding and session embedding. GCE–GNN employed inter-session information to generate item representations by building global graphs, and employed reverse position encoding and soft attention mechanism to generate the session representations. CGL combined training of self-supervised signals with supervised signals to alleviate the data sparsity problem. GC-HGNN proposed a hyper-graph enhanced neural network with global context support, which can effectively learn global and local context information and obtain better session features by fusing the two layers of information. Compared with the previous methods, these two models have achieved better results than the previous models on three public datasets, indicating that considering two levels of item embedding from the session graph and the global graph is conducive to prediction.

Our method, TGIE4Rec, achieves the best results on all three public datasets, which proves the superiority of our method. Different from the common GNN-based methods, our method not only integrates global information into the item representation of session sequence but also adds time information into the item

Table 4 Effects of different components

Dataset	Diginetica		Tmall		Nowplaying	
Measures	P@20	MRR@20	P@20	MRR@20	P@20	MRR@20
TGIE4Rec-OG	54.06	18.92	35.32	15.74	22.45	8.40
TGIE4Rec-OT	51.30	17.92	29.91	14.34	19.20	7.52
TGIE4Rec-NP	54.25	19.16	36.98	16.85	22.65	8.68
TGIE4Rec	54.31	19.21	37.86	17.49	22.82	8.81

representation of session sequence to enhance the sequential transition relations in session sequence.

Moreover, we generate two levels of session embedding to weaken the impact of the noise caused by integrating global information on the performance of the model and get better performance.

5.6 Model analysis (RQ2)

Further analysis of the model is carried out to explore the impact of various components in our proposed model on model performance.

The GIE mainly uses the item transition relations in all sessions to better infer the user preferences of the current session. The TIE is mainly used to learn the sequential transition relations of session sequences.

We verified the impact of these two modules on the performance of the model on three datasets and specially designed three sets of comparative experiments:

- TGIE4Rec-OG: Only employ the GIE for recommendations
- TGIE4Rec-OT: Only employ the TIE for recommendations
- TGIE4Rec-NP: The transition information enhanced session representation learning module does not add time information

As can be seen from Table 4, TGIE4Rec achieves the best results by integrating the global information enhanced representation learning module and the session representation learning module enhanced by the transfer information added with time information. Compared with TGIE4Rec-OT, TGIE4Rec-OG and TGIE4Rec-NP can employ global information to learn richer item representations, so they also achieve better results. Compared with TGIE4Rec-NP, TGIE4Rec integrates the time information into the item representation in the session sequence for neighbor embedding learning in the TIE, and it achieves better results.

Therefore, we can conclude that all three components have a certain contribution to the improvement of model performance. Among them, the contribution of global information and time information is the most prominent. Because in actual application scenarios, an item is often affected by multiple sessions. At the same

Table 5 Effects of different aggregation approaches

Dataset	Diginetica		Tmall		Nowplaying	
Measures	P@20	MRR@20	P@20	MRR@20	P@20	MRR@20
M-Pooling	48.72	16.54	30.22	14.5	19.41	6.45
NP-Attn	52.76	18.43	27.99	13.71	20.21	7.48
NP-Mean	54.31	19.21	37.86	17.49	22.82	8.81

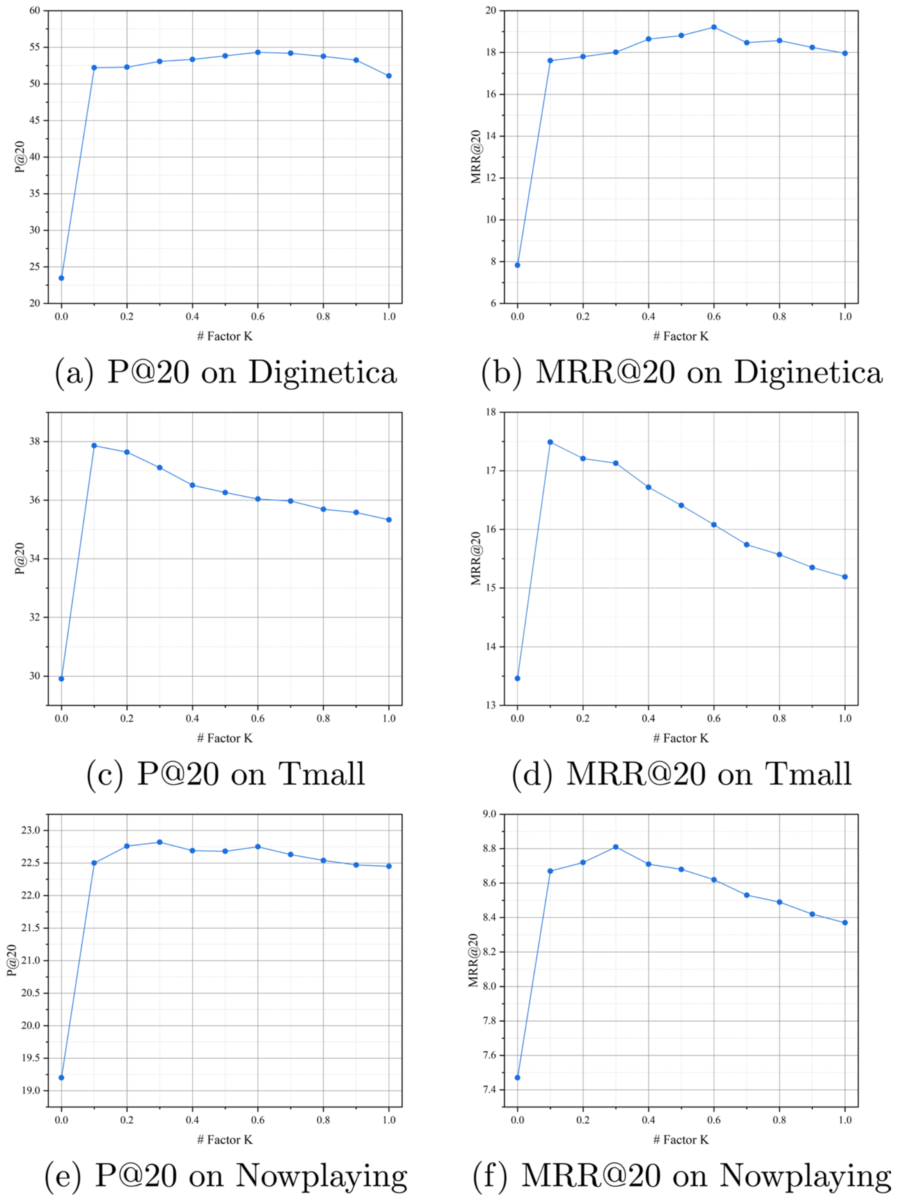


Fig. 6 Effect of different K values

time, the time information of the user's selected items is also crucial because it is directly related to the user's short-term preferences.

5.7 Impact of aggregation operations (RQ3)

The model is further analyzed to explore the impact of different aggregation methods on the performance of the model when two different levels of session embedding are aggregated.

Aggregating the item representations in the session sequence to form session embedding is crucial for the SBR. Therefore, we designed several comparative experiments to compare the effects of different aggregation methods on model performance.

- M-Pooling: In both the GIE and the TIE, the average pooling is used to aggregate the items represents in the session sequence to generate session embedding
- NP-Attn: In both the GIE and the TIE, the soft attention mechanism with reverse position coding is used to aggregate the items represents in the session sequence to generate session embedding
- NP-Mean: The GIE employs the soft attention mechanism with reverse position coding, and the TIE employs the average pooling mechanism to aggregate the items represents in the session sequence to generate session embedding.

As can be seen from Table 5, the aggregate methods M-pooling and NP-Attn do not achieve satisfactory results. Compared with the above two aggregation methods, the aggregate methods NP-Mean can get better results.

The reason for achieving such results is that NP-Mean uses different aggregation methods for GIE and TIE. Since GIE contains global information, position encoding information needs to be used to distinguish different neighbors, so a soft attention mechanism with reversed position encoding is used. Since TIE contains time sequence, it can directly use mean pooling to aggregate the items of the session sequence.

Table 6 Effects of GIE and TIE

Dataset	Diginetica		Tmall		Nowplaying	
Measures	P@20	MRR@20	P@20	MRR@20	P@20	MRR@20
TIE	51.30	17.80	27.57	13.72	18.87	7.47
GIE	54.22	19.04	35.09	18.80	22.43	8.40
TGIE4Rec	54.31	19.21	37.86	17.49	22.82	8.81

5.8 Effect of hyper-parameter K (RQ4)

The hyper-parameter K is used to control the proportion of the prediction scores obtained from the two modules in forming the final prediction score. Figure 6 shows the impact of different k value on Diginetica, Tmall and Nowplaying datasets.

As can be seen from Fig. 6, after getting the best results, the model performance begins to decrease as K increases. It achieves the best performance when the hyper-parameter is set to 0.6 on Diginetica, 0.1 on Tmall and 0.3 on Nowplaying. We think the cause of this result is the predictive score of the global information enhanced session representation learning layer is generally higher than that of the TIE. If the final score is calculated with a greater weight, it will lead to the ignorance of the learning results of the TIE, which will affect the performance of the model. Meanwhile, GIE contains a large amount of information and also contains some noise. Therefore, it is necessary to introduce the time information of TIE to alleviate the noise problem. If the K value is too small, it is easy to lose global information, and if the K value is too large, it is easy to introduce too much noise.

In addition, we simply use a weight coefficient K to control the importance of the two predicted scores, and we leave it to the future to work on how to better use the two predicted scores to get the final score.

5.9 Impact of GIE and TIE module

The model is further analyzed to explore the impact of GIE and TIE module on the performance of the model.

- TIE: Make predictions using the TIE module only.
- GIE: Make predictions using the GIE module only.
- TGIE4Rec: Make predictions using the combined TIE and GIE modules.

As can be seen from Table 6, the performance of using the TIE module alone for prediction is significantly lower than using TIE for prediction. This suggests that incorporating global information from other sessions can significantly enhance the representation capability of session embeddings. However, utilizing information from other sessions can introduce irrelevant items(noise), leading to suboptimal model performance. Therefore, we combine TIE and GIE to alleviate the noise interference issue present in GIE.

6 Conclusions

In this paper, we propose a novel architecture called Enhancing Session-based Recommendation with Transition and Global Information (TGIE4Rec). Our model learns two levels of session embedding: global information enhanced session embedding (GIE) and transition information enhanced session embedding (TIE).

GIE enriches the item representation of the current session by incorporating other session information. It uses reversed position coding and a soft attention mechanism to generate session embedding, providing a more intuitive measure of each item's contribution in the session sequence. TIE complements GIE by learning sequential transition relations among session sequences. By utilizing two different levels of session embedding for predictions and combining their recommendations, our model reduces the influence of irrelevant items (noise) introduced by GIE when considering other session information. We evaluate our model on three public datasets and achieve the best performance, demonstrating the effectiveness of our approach.

For the future work, on the one hand, we consider filtering sessions to extract relevant sessions for learning. On the other hand, introducing contrastive learning to obtain richer session representation is also a good improvement direction.

Acknowledgements The authors would like to thank the associate editor and the reviewers for their useful feedback that improved this paper, and thank to the providers of the public datasets used in this manuscript.

Author contributions SG contributed to conceptualization and methodology. JW helped in software and writing—original draft preparation. YZ helped in reviewing and editing. XD helped in writing.

Funding This work is supported by the Natural Science Foundation of Gansu Province (Grant No.22JR5RA156), the National Natural Science Foundation of China (Grant No.62341314) and the Backbone Fund of Youth Teachers' Capability Promotion (Grant No.NWNU-LKQN2020–14).

Data availability The Diginetica data that support the findings of this study are available from "Codalab","https://competitions.codalab.org/competitions/11161" The Tmall data that support the findings of this study are available from "Tiannchi", "https://tianchi.aliyun.com/dataset/dataDetail?dataId=42" The Nowplaying data that support the findings of this study are available from "DBIS", "http://dbis-nowplaying.uibk.ac.at/#nowplaying"

Declarations

Conflict of interest We declare that none of the authors have any financial or scientific conflicts of interest with regard to the research described in this manuscript.

Ethical approval This manuscript does not contain any studies with animals performed by any of the authors.

Consent to participate All authors read and approved the final manuscript.

Consent for publication All authors approved the final manuscript and submission to this journal.

References

1. Liu Y, Ma H, Jiang Y, Li Z (2022) Modelling risk and return awareness for p2p lending recommendation with graph convolutional networks. Appl Intell 52:1–16
2. Jiang Y, Ma H, Zhang X, Li Z, Chang L (2022) An effective two-way metapath encoder over heterogeneous information network for recommendation. In: Proceedings of the 2022 International Conference on Multimedia Retrieval, pp 90–98

3. Singer U, Roitman H, Eshel Y, Nus A, Guy I, Levi O, Hasson I, Kiperwasser E (2022) Sequential modeling with multiple attributes for watchlist recommendation in e-commerce. In: Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining, pp 937–946
4. Sarwar B, Karypis G, Konstan J, Riedl J (2001) Item-based collaborative filtering recommendation algorithms. In: Proceedings of the 10th International Conference on World Wide Web, pp 285–295
5. Hidasi BH, Karatzoglou A, Baltrunas L, Tikk D (2016) Session-based recommendations with recurrent neural networks. In: Proceedings of the 4th International Conference on Learning Representations, pp 1–10
6. Wu S, Tang Y, Zhu Y, Wang L, Xie X, Tan T (2019) Session-based recommendation with graph neural networks. In: AAAI, pp 346–353
7. Wang Z, Wei W, Cong G, Li XL, Mao X, Qiu M (2020) Global context enhanced graph neural networks for session-based recommendation. In: SIGIR, pp 169–178
8. Rendle S, Freudenthaler C, Schmidt-Thieme L (2010) Factorizing personalized markov chains for next-basket recommendation. In: WWW, pp 811–820
9. Jing L, Ren P, Chen Z, Ren Z, Ma J (2017) Neural attentive session-based recommendation. In: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, pp 1419–1428
10. Liu Q, Zeng Y, Mokhosi R, Zhang H (2018) Stamp: short-term attention/memory priority model for session-based recommendation. SIGKDD explorations, pp 1831–1839
11. Yuan J, Song Z, Sun M, Wang X, Zhao WX (2021) Dual sparse attention network for session-based recommendation. In: Thirty-Fifth Conference on Artificial Intelligence, AAAI 2021, pp 4635–4643
12. Li Y, Tarlow D, Brockschmidt M, Zemel RS (2016) Gated graph sequence neural networks. In: Bengio Y, LeCun Y (eds) ICLR (Poster)
13. Scarselli F, Gori M, Tsoi AC, Hagenbuchner M, Monfardini G (2009) The graph neural network model. IEEE Trans Neural Netw (TNN) 20(1):61–80
14. Kipf TN, Welling M (2017) semi-supervised classification with graph convolutional networks. In: Proceedings of the 5th International Conference on Learning Representations. ICLR '17
15. Veličković P, Cucurull G, Casanova A, Romero A, Liò P, Bengio Y (2017) Graph attention networks. In: ICLR 2018
16. Ding C, Zhao Z, Li C, Yu Y, Zeng Q (2023) Session-based recommendation with hypergraph convolutional networks and sequential information embeddings. Expert Syst Appl 223:119875
17. Wang J, Xie H, Wang FL, Lee L-K, Wei M (2023) Jointly modeling intra-and inter-session dependencies with graph neural networks for session-based recommendations. Inf Process Manag 60(2):103209
18. Hou Y, Hu B, Zhang Z, Zhao WX (2022) Core: simple and effective session-based recommendation within consistent representation space. In: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp 1796–1801
19. Wang M, Ren P, Mei L, Chen Z, Ma J, de Rijke M (2019) A collaborative session-based recommendation approach with parallel memory modules. In: SIGIR, pp 345–354
20. Luo A, Zhao P, Liu Y, Zhuang F, Wang D, Xu J, Fang J, Sheng VS (2020) Collaborative self-attention network for session-based recommendation. In: IJCAI, pp 2591–2597
21. Feng L, Cai Y, Wei E, Li J (2022) Graph neural networks with global noise filtering for session-based recommendation. Neurocomputing 472:113–123
22. Kang WC, McAuley J (2018) Self-attentive sequential recommendation. In: 2018 IEEE international conference on data mining (ICDM), pp 197–206
23. Wang H, Liu G, Liu A, Li Z, Zheng K (2019) DMRAN: a hierarchical fine-grained attention-based network for recommendation. In: IJCAI, pp 3698–3704
24. Wang S, Hu L, Wang Y, Sheng QZ, Orgun MA, Cao L (2019) Modeling multi-purpose sessions for next-item recommendations via mixture-channel purpose routing networks. In: IJCAI, pp 3771–3777
25. Qiu R, Li J, Huang Z, Yin H (2019) Rethinking the item order in session-based recommendation with graph neural networks. In: CIKM, pp 579–588
26. Xu C, Zhao P, Liu Y, Sheng VS, Xu J, Zhuang F, Fang J, Zhou X (2019) Graph contextualized self-attention network for session-based recommendation. In: IJCAI, pp 3940–3946

27. Pan Z, Cai F, Chen W, Chen H, De Rijke M (2020) Star graph neural networks for session-based recommendation. In: Proceedings of the 29th ACM International Conference on Information & Knowledge Management, pp 1195–1204
28. Zhu G, Hou H, Chen J, Yuan C, Huang Y (2022) Transition relation aware self-attention for session-based recommendation. CoRR **abs/2203.06407**
29. Peng D, Zhang S (2022) GC–HGNN: a global-context supported hypergraph neural network for enhancing session-based recommendation. Electron Commer Res Appl 52:101129
30. Xia X, Yin H, Yu J, Wang Q, Cui L, Zhang X (2021) Self-supervised hypergraph convolutional networks for session-based recommendation. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol 35, pp 4503–4511
31. Xia X, Yin H, Yu J, Shao Y, Cui L (2021) Self-supervised graph co-training for session-based recommendation. In: Proceedings of the 30th ACM International Conference on Information & Knowledge Management, pp 2180–2190
32. Zhang Z, Yang B, Xu H, Hu W (2024) Multi-level category-aware graph neural network for session-based recommendation. Expert Syst Appl 242:122773
33. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R (2014) Dropout: a simple way to prevent neural networks from overfitting. J Mach Learn Res 15(1):1929–1958
34. Ye W, Wang S, Chen X, Wang X, Qin Z, Yin D (2020) Time matters: sequential recommendation with complex temporal information. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2020, Virtual Event, pp 1459–1468
35. Yang Z, Ding M, Xu B, Yang H, Tang J (2022) Stam: a spatiotemporal aggregation method for graph neural network-based recommendation. In: The ACM Web Conference 2022, Virtual Event, pp 3217–3228
36. Zangerle E, Pichl M, Gassler W, Specht GS (2014) Nowplaying music dataset: extracting listening behavior from twitter. In: Proceedings of the First International Workshop on Internet-Scale Multimedia Management, pp 21–26
37. Lai S, Meng E, Zhang F, Li C, Wang B, Sun A (2022) An attribute-driven mirror graph network for session-based recommendation. In: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp.1674–1683
38. Zangerle E, Pichl M, Gassler W, Specht GS (2014) Nowplaying music dataset: extracting listening behavior from twitter. In: Proceedings of the First International Workshop on Internet-Scale Multimedia Management, pp 21–26
39. Sheng Z, Zhang T, Zhang Y, Gao S (2023) Enhanced graph neural network for session-based recommendation. Expert Syst Appl 213:118887
40. Chen Q, Jiang F, Guo X, Chen J, Sha K, Wang Y (2024) Combine temporal information in session-based recommendation with graph neural networks. Expert Syst Appl 238:121969
41. Pan Z, Cai F, Chen W, Chen C, Chen H (2022) Collaborative graph learning for session-based recommendation. ACM Trans Inf Syst 40(4):72–17226

Authors and Affiliations

Shiwei Gao[1] · Jingyu Wang[1] · Yufeng Zeng[1] · Xiaohui Dong[1]

✉ Shiwei Gao
gaoshiwei@nwnu.edu.cn

Jingyu Wang
2022222297@nwnu.edu.cn

Yufeng Zeng
2021222199@nwnu.edu.cn

Xiaohui Dong
dxh@nwnu.edu.cn

[1] College of Computer Science and Engineering, Northwest Normal University, Lanzhou 730070, Gansu, China