IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 7 February 2023, accepted 4 March 2023, date of publication 9 March 2023, date of current version 20 March 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3254897

RESEARCH ARTICLE

Attention-Enhanced Graph Neural Networks With Global Context for Session-Based Recommendation

YINGPEI CHEN, YAN TANG, AND YUAN YUAN
School of Computer and Information Science, Southwest University, Chongqing 400715, China
Corresponding author: Yan Tang (ytang@swu.edu.cn)

ABSTRACT Session-based recommendation is a crucial task aiming to predict users' interested items based only on anonymous user behaviors. Most recent solutions for session-based recommendation comprehensively consider the interactive information of all sessions but bring the problem of imbalanced positive and negative samples on model training. In this paper, we propose a novel approach, named Attention-enhanced Graph Neural Networks with Global Context for Session-based Recommendation (AGNN-GC), to learn and merge item transitions of all sessions in a cleverer way to enhance the recommendation effects. AGNN-GC first constructs global and local graphs based on all training sequences. Next, it uses graph convolutional networks with a session-aware attention mechanism to learn global-level item embedding in all sessions. Then it employs a graph attention networks module to learn local-level item embedding in the current sessions. After that, it fuses the learned two-level item embedding to enhance the feature presentations of items in the current session by a novel attention mechanism. Finally, applying the focal loss to balance positive and negative samples on model training accomplishes the prediction. Our experiments on three real-world datasets consistently show the superior performance of AGNN-GC over state-of-the-art methods.

INDEX TERMS Recommender systems, session-based recommendation, graph neural network, information fusion.

I. INTRODUCTION

Most existing recommenders depend on the user-item historical interactions [1]. In many online platforms, user identification may be anonymous, only with historical actions during the current session. To solve this problem, session-based recommendation (SBR) is proposed, which is a crucial task based only on limited and anonymous user behavior. It predicts interested items of users by implicit user feedback. Therefore, with insufficient user-item interaction data, SBR methods show better performance than the conventional recommendation methods.

Therefore, SBR has attracted extensive attention from researchers. Markov-based methods [2], [3], [4] treat the recommendation as a sequential optimization problem and deduces the user's next behavior by solving the problem. However, Markov-based methods only model the sequential transition of two consecutive items, disregarding other historical interaction information, which affects prediction accuracy. Probabilistic matrix factorization (PMF) [5] decomposes a user-item evaluation matrix into two low-rank matrices, where each low-rank matrix can represent the latent features of the user or item. However, it represents user preferences by considering positive user clicks, which cannot achieve fulfilling results.

Due to the impact of deep learning, recurrent neural network (RNN) has been successfully used in SBR. For instance, Hidasi et al. used RNN with gated recurrent units (GRUs) [6] in SBR for the first time and obtained promising performance [7]. And then Tan et al. further proposed an improved version by data augmentation to improve the robustness of training by pre-training to sufficiently consider the transitions of user behavior over time [8], [9]. Li et al. proposed NARM, a new method adding an attention

The associate editor coordinating the review of this manuscript and approving it for publication was Fabrizio Messina.

mechanism to RNN, which can simultaneously capture the users' continuous behaviors and interests [9], [10]. Considering the users' global and current preferences, Liu et al. proposed STAMP, which applies an attention mechanism and multilayer perceptron (MLP) networks [11]. Recently, RNN and variational auto-encoder (VAE) are integrated to extract user preferences in session by Song et al. [12]. Besides, Wu et al. proposed SR-GNN, which applies a graph neural network (GNN) combined with RNN in SBR [13]. Similarly, Xu et al. presented GC-SAN, using GNN and a self-attention mechanism to learn the long-term dependence between items in session sequences [14]. Qiu et al. proposed FGNN, which computes the information flow between items in the session using the multi-weight graph attention layer (WGAT) to achieve the item representation and then aggregates the item representation through the feature extractor to extract features [15]. Wang, Z et al. proposed GCE-GNN to learn the whole transitions of items from the current and historical sessions by enlarging the range of helpful information, achieving outstanding performance [16]. Chen, Y et al. presented MAE-GNN to select significant node information and capture user preferences from multiple dimensions by combining a dual-gated graph neural network and multi-head attention mechanisms [17]. More recently, Dong et al. presented GPAN, which obtains each item embedding of the current session by the high-low order session perceptron, combines the position embedding of the items to obtain short-term user preferences, and passes it to the self-attention layer to obtain long-term user preferences [18]. However, it does not achieve better results than GCE-GNN and MAE-GNN.

Compared with other SBR methods, GNN-based methods have achieved outstanding performance, but they still have some limitations. GCE-GNN, MAE-GNN, and GPAN started modeling user preferences based on all sessions while bringing the problem of imbalanced positive and negative samples on model training.

To this end, we propose a novel approach named Attention-enhanced Graph Neural Networks with Global Context for Session-based Recommendation (AGNN-GC). Global and local graphs are constructed based on all training sequences. Then it uses graph convolutional networks (GCNs) with a session-aware attention mechanism to learn global-level item embedding in all sessions. And it employs a graph attention networks (GATs) module to learn local-level item embedding in the current sessions. Particularly, It uses a novel attention mechanism to enhance fused information after learning global-level item embedding in all sessions and local-level item embedding in the current session. Besides, it applies the focal loss to balance positive and negative samples in model training. The main contributions of our work are summarized as follows:

- We propose a novel attention mechanism to process fused features after learning and merging the information of the global-level and local-level item embedding representations, which helps learn the final representation of a session sequence.
- We apply the focal loss to update the function for optimizing the model to solve the problem of imbalanced positive and negative samples on model training.
- We conduct extensive experiments on three real-world datasets, which consistently show the superior performance of AGNN-GC over state-of-the-art methods.

II. RELATED WORK

This section reviews the related work on SBR, including conventional methods, deep-learning-based methods, and graphs-based neural network methods.

A. CONVENTIONAL METHODS

There are many conventional studies on SBR, mainly based on the Markov chain or PMF. Markov-based methods [2], [3], [4] treat the recommendation as a sequential optimization problem and solve it to deduct a user's next behavior using the previous one. However, Markov-based methods can only model the sequential transition of two consecutive items, which cannot achieve fulfilling results. PMF [5] is common on SBR. It decomposes a user-item evaluation matrix into two low-rank matrices, where each low-rank matrix can represent the user's or item's latent features. However, it leads to user preferences being represented only by positive user clicks, which has certain limitations.

B. DEEP-LEARNING-BASED METHODS

For SBR, Hidasi et al. proposed a recurrent neural network (RNN) approach for the first time [7]. In the same year, they proposed a parallel RNN approach, which considers the basic information of clicked items and uses some other features to improve the recommendation result [19]. On this basis, Tan et al. enhanced the performance of the model above by data augmentation, pre-training, and taking temporal shifts in user behavior into account [8]. Li et al. proposed NARM [10], similar to the Transformer [20], which is commonly used in natural language processing. It has an encoder-decoder neural network structure and uses an RNN approach with an attention mechanism to capture users' sequential behavior features. Liu et al. proposed STAMP, using simple multilayer perceptron (MLP) networks and attention mechanisms to achieve users' global preferences and current preferences [11]. To account for shifts in user interest, RNN and variational auto-encoder (VAE) are integrated to extract user preferences in session by Song et al. [12].

C. GNN-BASED METHODS

Recently, graph neural network (GNN) performs well in SBR models, which shows a promising direction for SBR. Most GNN-based methods construct a session sequence as a session graph and then utilize GNN to aggregate information on adjacent nodes.

For SBR, SR-GNN combines RNN, GGNNs, and attention network [13], [21]. It constructs user-item sequences into graph-structured data and captures underlying transitions of the items in sessions. Similarly, Xu et al. presented GC-SAN,

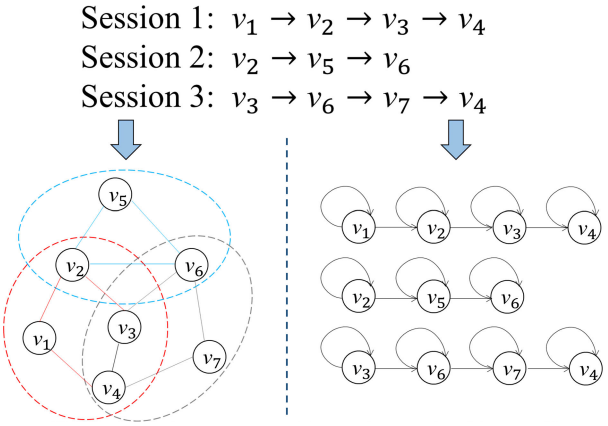

FIGURE 1. An example of global graph and local graph.

which utilizes GNN and the self-attention mechanism to learn long-term dependencies between items in session sequences [14]. Following SR-GNN, TAGNN improves the attentive module, further considering the relevance of the historical behaviors given a specific target item [22]. Qiu et al. presented FGNN, using the multi-weight graph attention layer (WGAT) to compute the information flow between items in the session to learn the item representation and then aggregate the item representation through the feature extractor to extract features [15]. Wang, Z et al. proposed GCE-GNN to learn the whole transitions of items from current and historical sessions by enlarging the range of helpful information, achieving outstanding performance [16]. Chen, Y et al. presented MAE-GNN to select significant node information and capture user preferences from multiple dimensions by combining a dual-gated graph neural network and multi-head attention mechanisms [17]. More recently, Dong et al. presented GPAN, which uses the high-low order session perceptron to model directed and undirected graphs respectively to obtain high-order and low-order item transitions in session, and session position information to enhance the relevance of sequence order to user preferences [18].

III. PRELIMINARIES

This section first defines the problem statement and then introduces two approaches of constructing graphs, the global graph that describes the sessions and the local graph that illustrates the current session [16].

A. PROBLEM STATEMENT

The purpose of the SBR is to predict the user's next click item, using only the user's historical session sequences rather than the long-term preference profile. Let $V = \{v_1, v_2, \ldots, v_m\}$ contain all user-clicked items in the session. In addition, the timestamp sorted $s = [v_{s,1}, v_{s,2}, \ldots, v_{s,n}]$ is used to represent an anonymous session sequence, and the length of S is l. Let $v_{s,n+1}$ denote the next user-click item. Ultimately, a probability ranking list of all candidate items is generated, and the top-k probability items will become the recommended candidate items. For example, FIGURE 1 illustrates three different session sequences labeled Session1, Session2, and Session3. They include all clicked items in the according sessions, with arrows describing the order of clicks one by one.

B. GLOBAL GRAPH AND LOCAL GRAPH

This subsection introduces two graph models, the global and the local graphs, to describe the transitions between items in current sessions at different levels [16]. FIGURE 1 shows an example of the global and the local graphs. From the figure, the image after the left arrow is the constructed global graph with different sequences of sessions circled by dotted lines of different colors. The image after the right arrow is the constructed local graph, corresponding to the three session sequences on the top of the figure.

1) GLOBAL GRAPH

Deep-learning-based methods (e.g., RNN-based SBR methods [7], [8], [10], [12]) aim to model the sequential patterns to learn the item representation of sessions. Different from RNN-based SBR methods, GNN-based SBR methods [13], [14], [15] learn item embedding of sessions by constructing the historical interaction sequence of a session into a session graph to capture information about adjacent nodes. Therefore, we choose to construct the global graph to model the global-level item transitions between items in all sessions.

We consider global-level item transitions for global-level item representation learning by integrating all pairwise item transitions over sessions [16], [23]. We define a concept (i.e., ε-neighbor set) for modeling the global-level item transition. For each item v_i^p in session S_p, the ε-neighbor set of v_i^p denotes a set of items, as follows:

$$N_\varepsilon\left(v_i^p\right) = \left\{v_j^p \mid v_i^p = v_{i'}^p \in S_p \cap S_q; v_j^p \in S_q; j \in \left[i' - \varepsilon, i' + \varepsilon\right]; S_p \neq S_q\right\} \quad (1)$$

where v_i^p is the i'-th item in session S_q, ε is the hyperparameter that controls the modeling scope of item transitions between v_i^p and other items in S_q. Besides, we use v_j^p to represent each item in ε-neighbor set $N_\varepsilon\left(v_i^p\right)$.

Based on ε-neighbor set, for item v_i global-level item transition is defined as $\left\{\left(v_i, v_j\right) \mid v_i, v_j \in V; v_j \in N_\varepsilon\left(v_i\right)\right\}$. Note that, to improve efficiency, we do not consider the direction of the global-level item transitions [16].

Next, the global graph is defined as $G_g = \left(V_g, E_g\right)$ which is an undirected weighted graph, where V_g represents the graph node set which has all items in V, and $E_g = \left\{e_{ij}^g \middle| \left(v_i, v_j\right) \middle| v_i, v_j \in V; v_j \in N_\varepsilon\left(v_i\right)\right\}$ represents the set of edges corresponding to two pairwise items in all sessions. Besides, we generate a weight for v_i's adjacent edges to emphasize the significance of its neighbors. The frequency of all the sessions is used as the weight of each corresponding edge, and we only keep top-k edges with the highest importance for each item v_i on graph G_g [16].

2) LOCAL GRAPH

The target of constructing a local graph is to model the transitions of adjacent items in the current session to learn

local-level item embedding. Inspired by [13] and [22], each session sequence is modeled into a directed local graph, which describes the click order of items in the session. It is defined as $G_l = (V_l, E_l)$, where V_l denotes the item set and E_l denotes the edge set. That is, each node $v_i^s \in V_l$ in G_l represents an item, and each edge $(v_{i-1}^s, v_i^s) \in E_l$ denotes that the user clicks the item v_{i-1}^s and item v_i^s in sequence, which is called local-level item-transition pattern. Like [15], each node has a self-loop to fuse its information in the following modeling.

Inspired by [16], there may be four types of edge connections in the local graph, which are represented as r_{in}, r_{out}, r_{in-out} and r_{self}. For example, in edge $\left(v_i^s, v_j^s\right)$, r_{in} denotes there is only a single transition from v_j^s to v_i^s. Similarly, r_{out} means there is only a single transition from v_i^s to v_j^s, and r_{in-out} indicates there are both transitions between v_i^s and v_j^s; r_{self} represents its transiting information.

IV. PROPOSED METHOD

This section covers our proposed method in detail. AGNN-GC aims to utilize global-level and local-level item transitions to capture the user preferences of the current session for recommendation. At first, based on the global graph structure, it uses graph convolutional networks (GCNs) with a session-aware attention mechanism to learn global-level item embedding in all sessions. Then it employs a graph attention networks (GATs) module to learn the local-level item embedding in the current sessions. After that, it fuses the learned two-level item embedding for modeling the user preference of the current session with a novel attention mechanism to process fused features. Ultimately, it outputs the probability that the top-k candidates are recommended. The overview of the proposed AGNN-GC method is shown in FIGURE 2.

A. LEARNING GLOBAL-LEVEL ITEM EMBEDDING

Referring to previous methods [24], [25], this module is based on GCNs, and we calculate the attention weights according to the importance of each connection. Since a single item may involve multiple sessions, from which we can obtain useful item transitions that are helpful for subsequent recommendation task. By mean pooling to obtain the first-order neighbor's features of item v is a simple and effective solution. However, not all items in v's ε-neighbor set are related to the user preferences of the current session, so we consider using session-aware attention to emphasize the importance of items in its $N_\varepsilon(v)$ [16]. In terms of session-aware attention, each item in $N_\varepsilon(v)$ is linearly combined, which is as follows:

$$\boldsymbol{h}_{N_{v_i}^g} = \sum_{v_j \in N_{v_i}^g} \pi\left(v_i, v_j\right) \boldsymbol{h}_{v_j} \tag{2}$$

where $\pi\left(v_i, v_j\right)$ denotes the weight of different neighbors, and $\boldsymbol{h}_{v_j}$ denotes the representation of item v_j in the unified embedding space. The closer an item is to the preference of the current session, the greater the significance of the item is to the recommendation, which is consistent with empirical judgment. Hence, $\pi\left(v_i, v_j\right)$ is formulated as follows:

$$\pi\left(v_i, v_j\right) = \boldsymbol{q}_1^{\mathrm{T}} \operatorname{LeakyReLU}\left(\boldsymbol{W}_1\left[\left(\boldsymbol{s} \odot \boldsymbol{h}_{v_j} \| w_{ij}\right)\right]\right) \tag{3}$$

Here $w_{ij} \in \mathbb{R}^1$ represents the weight of the edge v_1, v_2, $\odot$ denotes element-wise product, $\|$ represents the concatenation operation, $\boldsymbol{q}_1 \in \mathbb{R}^{d+1}$ and $\boldsymbol{W}_1 \in \mathbb{R}^{d+1 \times d+1}$ are trainable parameters. And we choose LeakyReLU as the activation function [23]. $\boldsymbol{s}$ represents the features of current sessions. And it can be obtained by calculating the mean value of the current session's item representation:

$$\boldsymbol{s} = \frac{1}{|S|} \sum_{v_i \in S} \boldsymbol{h}_{v_i} \tag{4}$$

After that, the coefficients across $N_\varepsilon(v)$ connected with v_i are normalized by the softmax function:

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

Finally, we aggregate the item representation $\boldsymbol{h}_v$ and its neighbor representation $\boldsymbol{h}_{N_v}^g$, the k-th representation of multiple aggregator layers is implemented as follows:

$$\boldsymbol{h}_v^{g,(k)} = \operatorname{ReLU}\left(\boldsymbol{W}_2^{(k)}\left[\boldsymbol{h}_v^{(k-1)} \| \boldsymbol{h}_{N_v^g}^{(k-1)}\right]\right) \tag{6}$$

where ReLU is the activation function, $\boldsymbol{h}_v^{(k-1)}$ is generated from previous $k-1$ steps, and the initial $\boldsymbol{h}_v^{(0)}$ is the same as $\boldsymbol{h}_v$ when $k = 1$. Besides, $\boldsymbol{W}_2^{(k)} \in \mathbb{R}^{d \times 2\,d}$ is the k-th layer aggregation weight.

Through the operations above, each global-level item embedding representation is dependent on itself and the connectivity information representation of the current session [16].

B. LEARNING LOCAL-LEVEL ITEM EMBEDDING

According to [25] and [26], we adopt GATs to learn the local-level item embedding representation. Specifically, the attention mechanism of GATs is used to compute the significant weights of different nodes. Given a node v_i, the significant weight of v_j on it can be calculated by element-wise product and non-linear transformation:

$$e_{ij} = \operatorname{LeakyReLU}\left(\boldsymbol{a}_{r_{ij}}^{\mathrm{T}}\left(\boldsymbol{h}_{v_i} \odot \boldsymbol{h}_{v_j}\right)\right) \tag{7}$$

where r_{ij} indicates the relationship between v_i and v_j, $\boldsymbol{a}_* \in \mathbb{R}^d$ represent the weight vectors, and LeakyReLU is the activation function.

Referring to [16], according to the relationship of each node v_i, we train four weight matrices, $\boldsymbol{a}_{in}$, $\boldsymbol{a}_{out}$, $\boldsymbol{a}_{in-out}$ and $\boldsymbol{a}_{self}$ respectively, which can describe the impact of all nodes over v_i. And to make the weights of different nodes comparable, the softmax function is utilized to normalize them, and the attention weights coefficient α_{ij} is as follows:

$$\alpha_{ij} = \frac{\exp\left(e_{ij}\right)}{\sum_{v_k \in N_{v_i}^S} \exp\left(\operatorname{LeakyReLU}\left(\boldsymbol{a}_{r_{ik}}{}^{\mathrm{T}}\left(\boldsymbol{h}_{v_i} \odot \boldsymbol{h}_{v_k}\right)\right)\right)} \tag{8}$$

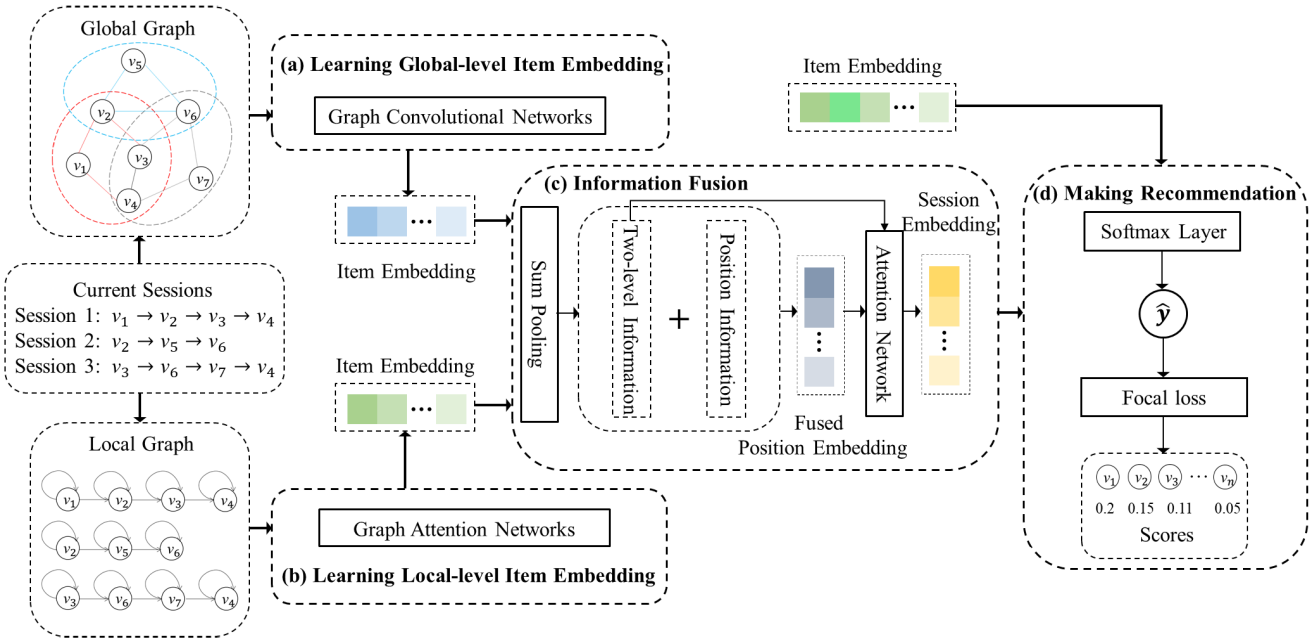

FIGURE 2. Overview of the proposed method. (a) Firstly, based on the global graph, global-level item embedding in all sessions is learned by graph convolutional networks with a session-aware attention mechanism. (b) Then, based on the local graph, a graph attention networks module learns local-level item embedding in the current sessions. (c) Then, the model fuses the learned two-level item embedding with a novel attention mechanism for processing fused features. (d) Finally, candidate items will be scored.

Due to different neighbors, α_{ij} is asymmetric. Hence, we need to calculate a linear combination of features of each node v_i to get the feature representations:

$$\boldsymbol{h}_{v_i}^{l} = \sum_{v_j \in N_{v_i}^{S}} \alpha_{ij} \boldsymbol{h}_{v_j} \tag{9}$$

After aggregating the critical information of the node itself and its neighbors in the current session, we obtain the local-level item embedding representation for each node.

C. INFORMATION FUSION

After obtaining the global-level and local-level item embedding representations, we need to fuse the information before making recommendation. With the dropout [27] on the global-level item embedding, the two-level information is extracted by sum pooling, which is as follows:

$$\boldsymbol{h}_{v_i}^{*} = \text{SumPooling}\left(\text{dropout}\left(\boldsymbol{h}_{v}^{g,(k)}\right), \boldsymbol{h}_{v_i}{}^{l}\right) \tag{10}$$

where $\boldsymbol{h}_{v_i}^{*}$ is the item representation with two-level information in the session.

Considering noise filtering and the items clicked later in the session show the greater significance for the recommendation [13], [22], we design a new position attention mechanism to compute soft-attention weights on all items in the session, which fuses two-level information with position information of items in the session.

Next, we can get the representations of items among the session, i.e., $\boldsymbol{H} = \left[\boldsymbol{h}_{v_1}^{*}, \boldsymbol{h}_{v_2}^{*}, \ldots, \boldsymbol{h}_{v_l}^{*}\right]$. We also employ reverse position embedding matrix $\boldsymbol{P} = [\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_l]$ to reveal the position information embeddings for all the items involved in the session, where $\boldsymbol{p}_1 \in \mathbb{R}^d$ is the first position vector, $\boldsymbol{p}_l \in \mathbb{R}^d$ is the last position vector [16].After these operations above, we leverage concatenation and non-linear transformation to generate the fused position embedding:

$$f_i^{*} = \tanh\left(\boldsymbol{W}_3\left[\boldsymbol{h}_{v_l}{}^{*} \| \boldsymbol{p}_{l-i+1}\right] + \boldsymbol{b}\right) \tag{11}$$

where $\boldsymbol{W}_3 \in \mathbb{R}^{d \times 2d}$ and $\boldsymbol{b} \in \mathbb{R}^d$ are the trainable parameters and $\|$ denotes the concatenation operation. Referring to [16], the reversed position information can more accurately suggest the significance of each item than the forward position information. In our work, the two-level information of items in the session is averaged:

$$\boldsymbol{s}^{*} = \frac{1}{l}\sum_{i=1}^{l} \boldsymbol{h}_{v_i}^{*} \tag{12}$$

Based on (11) and (12), different from previous work, we apply a novel attention mechanism to calculate soft-attention weights:

$$\beta_i = \boldsymbol{q}_2^{\mathrm{T}} \text{ReLU}\left(\boldsymbol{W}_4 \boldsymbol{f}_i^{*} + \boldsymbol{W}_5 \boldsymbol{s}^{*} + \boldsymbol{c}\right) \tag{13}$$

where $\boldsymbol{W}_4, \boldsymbol{W}_5 \in \mathbb{R}^{d \times d}$ and $\boldsymbol{q}_2, \boldsymbol{c} \in \mathbb{R}^d$ are learnable parameters. Exceptionally, to release the vanishing gradient problem [28], we choose ReLU as the activation function [29].

Next, it is normalized by a softmax function:

$$\alpha_i = \text{softmax}\left(\beta_i\right) = \frac{\exp\left(\beta_i\right)}{\sum_{i=1}^{n} \exp\left(\beta_i\right)} \tag{14}$$

Ultimately, the session embedding representation can be generated through linear combination operations:

$$\boldsymbol{F} = \sum_{i=1}^{l} \alpha_i \boldsymbol{h}_{v_i}^{*} \tag{15}$$

The session embedding representation $\boldsymbol{F}$ can represent the session features, by fusing the global-level and local-level information and considering the order and position information of all involved items.

D. MAKING RECOMMENDATION

Based on the obtained session embedding representation $\boldsymbol{F}$, let $\widehat{\boldsymbol{y}}_{\boldsymbol{i}}$ denote the final recommendation probability for each item v_i based on the original embedding representation and the current session embedding representation. We first take dot product and next add a softmax function to get the results:

$$\widehat{\boldsymbol{y}}_{\boldsymbol{i}} = \text{softmax}\left(\boldsymbol{F}^{\mathrm{T}}\boldsymbol{h}_{v_i}\right) \tag{16}$$

Different from the previous work, to solve the problem of imbalanced positive and negative samples, the focal loss [30] is innovatively applied to replace the conventional cross-entropy loss for optimizing the model, which is defined as follows:

$$\mathcal{L}(\hat{\mathbf{y}}) = \begin{cases} -\alpha \sum_{i=1}^{m} (1-\widehat{\boldsymbol{y}}_{\boldsymbol{i}})^{\gamma} \log \widehat{\boldsymbol{y}}_{\boldsymbol{i}}, & y_i = 1 \\ -(1-\alpha)\widehat{\boldsymbol{y}}_{\boldsymbol{i}}^{\gamma} \log (1-\widehat{\boldsymbol{y}}_{\boldsymbol{i}}), & y_i = 0 \end{cases} \tag{17}$$

where y is the one-hot vector, which denotes the ground truth of the target item, α is a factor that can balance the ratio of positive and negative samples, and γ is a factor that can solve the problem of imbalance between distinguishable and indistinguishable samples. Hence, it ensures that in the training process, the model will pay more attention to those small and indistinguishable samples, reducing the impact of the gradient superposition of a significant number of distinguishable samples on model training.

V. EXPERIMENTS

This section describes the experiments' datasets, baseline methods, evaluation metrics, and parameter settings. To verify the validity of the proposed model AGNN-GC, we conducted a range of experiments by answering the subsequent questions:

- RQ1: Does AGNN-GC outperform state-of-the-art methods on real-world datasets?
- RQ2: Does the focal loss perform better than the conventional cross-entropy loss in training AGNN-GC? Does our novel scheme for representing user interests improve the performance?
- RQ3: How do different hyperparameters settings of the focal loss affect the performance of AGNN-GC?
- RQ4: How do different hyperparameters settings of dropout affect the performance of AGNN-GC?

A. EXPERIMENTAL SETUP

1) DATASETS

We conducted extensive experiments on three representative public datasets, i.e., Diginetica, Nowplaying, and Tmall. Diginetica dataset comes from CIKM Cup 2016,[1] and we only select the public transactional data. Nowplaying dataset comes from [31], which contains users' music-listening behavior. Tmall dataset comes from IJCAI-15 competition,[2] which records the shopping logs of anonymous users on Tmall e-commerce platform [16].

TABLE 1. Statistics of the datasets.

Dataset	Diginetica	Nowplaying	Tmall
total clicks	982961	1367963	818479
training sessions	719470	825304	351268
test sessions	60858	89824	25898
total items	43097	60417	40728
average session length	5.12	7.42	6.69

Following previous methods [16], [32], we conduct the same data preprocessing step on the datasets above to make it fair. Notably, we filter out the sessions with a length of 1 and items with less than five occurrences, and we choose the sessions of last week (latest data) as the test set. Besides, we split a sequence of session data $S = [s_1, s_2, \ldots, s_n]$ into a series of sequences and corresponding labels, i.e., $([s_1], s_2), ([s_1, s_2], s_3), \ldots, ([s_1, s_2, \ldots, s_{n-1}], s_n)$ for training and testing on all three datasets. After preprocessing, the statistics of the datasets are shown in TABLE 1.

2) BASELINES

We compare the proposed AGNN-GC method with the following representative baseline methods, including three conventional and ten latest deep-learning-based recommendation methods.

- POP: It always recommends the most popular items in the training set.
- Item-KNN [33]: It is a conventional recommendation method based on cosine similarity between session vectors.
- FPMC [3]: It is a Markov-based recommendation method.
- GRU4REC [7]: It is the first SBR method based on RNN that uses Gated Recurrent Units (GRUs).
- NARM [10]: It is a deep-learning-based method that extracts sequential action features of users by an attentive RNN-based network.
- STAMP [11]: It is an SBR method with short-term attention memory priority that effectively captures user preferences.
- SR-GNN [13]: It is the first GNN-based SBR method that captures the user's global and current preferences.
- CSRM [34]: It applies the memory networks to learn the latest m sessions for better predicting the intentions of the current session.
- GC-SAN [14]: It is an SBR method with self-attention networks to learn global and local dependency information between items in a session.
- FGNN [15]: It employs a weighted attention graph layer to learn the item representations and utilizes a graph feature encoder to extract the final representation of the session.

[1]Data files of CIKM Cup 2016: https://competitions.codalab.org/competitions/11161

[2]Data files of IJCAI-15 competition: https://tianchi.aliyun.com/dataset/42

TABLE 2. Comparisons of HR@20 and MRR@20 between AGNN-GC and baselines.

Method	Diginetica		Nowplaying		Tmall	
	HR@20	MRR@20	HR@20	MRR@20	HR@20	MRR@20
POP[a]	0.89	0.20	2.28	0.86	2.00	0.90
Item-KNN	35.75	11.57	15.94	4.91	9.15	3.31
FPMC	26.53	6.95	7.36	2.82	16.06	7.32
GRU4REC	29.40	8.31	7.92	4.48	10.93	5.89
NARM	49.70	16.17	18.59	6.93	23.30	10.70
STAMP	45.64	14.32	17.66	6.88	26.47	13.36
SR-GNN	50.77	17.62	18.87	7.47	27.57	13.72
CSRM	50.55	16.38	18.14	6.42	29.46	13.96
GC-SAN	48.58	16.55	17.31	6.80	19.14	8.54
FGNN	50.58	16.84	18.78	7.15	25.24	10.39
GCE-GNN	54.22	**19.04**	22.37	8.40	33.42	15.42
MAE-GNN[b]	51.61	17.77	-	-	33.44	15.37
GPAN	53.96	18.84	22.64	7.66	28.37	13.86
AGNN-GC	**54.35**	19.00	**23.07**	**8.62**	**33.68**	**15.54**

a The results of POP are quoted from SR-GNN and GCE-GNN.
b The authors of MAE-GNN do not release the results on Nowplaying dataset. We only refer to the results on Diginetica and Tmall datasets.

- GCE-GNN [16]: It utilizes GNN to learn two levels of item embeddings from global and session graphs, and next aggregates the learned item representations considering the position embeddings.
- MAE-GNN [17]: It combines a dual-gated graph neural network and multi-head attention mechanisms for SBR to select significant node information and capture user preferences from multiple dimensions.
- GPAN [18]: It utilizes the high-low order session perceptron to model directed and undirected graphs respectively to obtain high and low order item transitions in session, and session position information to enhance the relevance of sequence order to user preferences.

3) EVALUATION METRICS

Following previous methods [13], [15], [16], [17], [18], we adopt the commonly used HR@20 (Hit Rate)[3] and MRR@20 (Mean Reciprocal Rank) as evaluation metrics [9], [35].

4) PARAMETER SETTINGS

All the experiments below were run on Ubuntu 16.04.6 LTS docker system with pytorch 1.10.1.

In our experiments, the dimension of embedding vectors is set to 100, and the batch size is set to 100, the L2 penalty is set to 10^{-5}. The dropout ratio is set to 0.5 in Diginetica, 0.7 in Tmall, and no dropout in Nowplaying. The hyperparameters α and β of the focal loss are set to 0.9 and 2, respectively. Moreover, we select a random 10% subset of the training set as the validation set. All parameters are initialized using a Gaussian distribution with a mean value of 0 and a standard deviation of 0.1. After that, the mini-batch Adam optimizer with the initial learning rate of 0.001 is adopted, which will decay at a rate of 0.1 every three epochs. Besides, the number of neighbors and the maximum distance of items are set to 12 and 3, respectively. Otherwise, for fairness, the parameters are also set as the ones when the model performs best.

[3]Note that [10], [11], [13], [17], [18], and [22] used different metric names for HR@20 (e.g., Precision@20 and Recall@20). However, they used the same formula to obtain this measurement (i.e., the proportion of cases when the desired item is among the top-20 items in all cases).

B. COMPARISON WITH BASELINES (RQ1)

To verify the overall performance of AGNN-GC, we compare it with existing representative baselines. The overall performance in HR@20 and MRR@20 is shown in TABLE 2, where the best results are highlighted in bold.

As shown in TABLE 2, compared with the baselines, it can be observed that AGNN-GC achieves superior performance in most metrics across all three datasets, which shows the superiority and effectiveness of AGNN-GC.

From TABLE 2, conventional methods generally do not perform well. POP and Item-KNN are early conventional recommendation methods, while FPMC is recommended based on the Markov chain [13]. Their performance is inferior to AGNN-GC because they are not based on advanced deep neural networks.

Compared with conventional methods, the latest deep-learning-based methods significantly perform better due to their greater ability to capture complicated user behaviors. Although GRU4REC performs inferior to Item-KNN on Diginetica and Nowplaying datasets, it still ensures the effectiveness of RNN in modeling sequences. Since GRU4REC only considers sequential relationships rather than the remaining information in the sequence, missing the meaningful shift of user preferences, it performs worse than NARM and STAMP. NARM is a sequence method based on RNN, which considers the unidirectional transitions between adjacent items. STAMP uses an attention mechanism and multilayer perceptron (MLP) networks to achieve the

TABLE 3. Performance of different session embedding strategies.

Strategy	Diginetica		Nowplaying		Tmall	
	HR@20	MRR@20	HR@20	MRR@20	HR@20	MRR@20
AGNN-GC-S-C	54.22	**19.04**	22.37	8.40	33.42	15.42
AGNN-GC-T-C	54.16	18.90	22.30	8.49	33.30	15.51
AGNN-GC-L-C	54.09	18.90	22.47	8.57	33.40	15.37
AGNN-GC-R-C	54.29	18.90	22.53	8.54	33.62	15.41
AGNN-GC-D-C	51.14	17.37	20.93	6.99	31.04	14.56
AGNN-GC-S-F	54.30	19.01	22.50	8.54	32.73	15.19
AGNN-GC-T-F	54.08	18.86	22.53	8.56	32.89	15.19
AGNN-GC-L-F	54.19	18.92	22.57	8.57	33.02	15.16
AGNN-GC	**54.35**	19.00	**23.07**	**8.62**	**33.68**	**15.54**
AGNN-GC-D-F	51.17	17.36	21.08	7.09	30.63	14.31

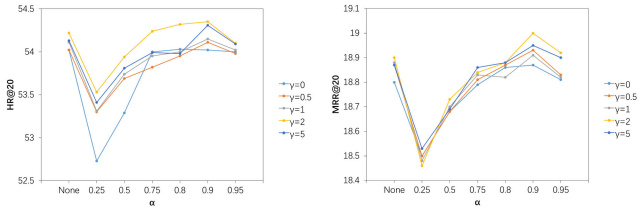

FIGURE 3. Performance effects of varying hyperparameters on Diginetica.

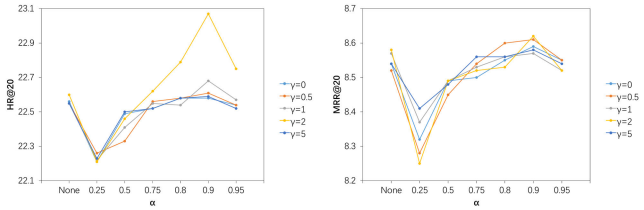

FIGURE 4. Performance effects of varying hyperparameters on Nowplaying.

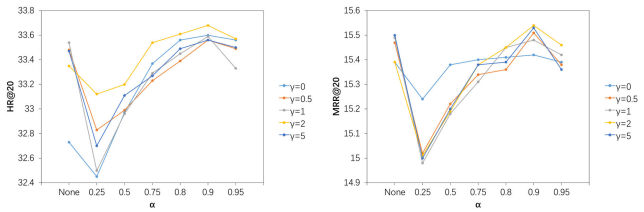

FIGURE 5. Performance effects of varying hyperparameters on Tmall.

user's global and current preferences. The following CSRM method outperforms NARM and STAMP on Diginetica and Tmall. CSRM regards other sessions as a whole, showing the effectiveness of using item transitions from other sessions.

According to TABLE 2, it is obvious that after introducing GNN to SBR, the performance of methods can be observably improved, especially on Diginetica and Nowplaying. This is because constructing session sequences into graph-structured data is adequately capable of considering the complex transitions among items in sessions rather than just considering the unidirectional transitions between adjacent items. In other words, GNN has a more excellent capability than RNN of capturing more complex inter-item dependencies in a session sequence. SR-GNN, GC-SAN, and FGNN construct each session sequence as a simple graph and employ GNN to encode the items, which verifies the effectiveness of using GNN in SBR.

Following GNN-based methods, GCE-GNN makes a significant breakthrough in performance on the three datasets. GCE-GNN can learn two levels of context information and incorporate relative position information, achieving better performance than previous methods [16]. Besides, MAE-GNN filters out the noise interference of irrelevant nodes [17]. However, the performance of MAE-GNN on Diginetica is not satisfactory. This is because most sessions are too short, and the performance of MAE-GNN will deteriorate.

Our proposed method AGNN-GC performs better than GPAN and MAE-GNN on all datasets and outperforms GCE-GNN on Nowplaying and Tmall. It applies a novel attention mechanism to process fused features and the focal loss to update the function for optimizing the model. It can solve the problem of imbalanced positive and negative samples, which is why AGNN-GC has superior performance.

C. COMPARISON WITH VARIANTS OF THE PROPOSED MODEL (RQ2)

To demonstrate the impact of different strategies on the recommendation results, we compare AGNN-GC with several variants of AGNN-GC, AGNN-GC-S-C, AGNN-GC-T-C, AGNN-GC-L-C, AGNN-GC-R-C, AGNN-GC-D-C, AGNN-GC-S-F, AGNN-GC-T-F, AGNN-GC-L-F, and AGNN-GC-D-F, which are tested on three datasets, Diginetica, Nowplaying, and Tmall. The evaluation metrics are HR@20 and MRR@20, respectively. The detailed description of the variants above is as follows:

- AGNN-GC-S-C: AGNN-GC-S-C adopts the sigmoid-adding attention mechanism to process fused features and retains the conventional cross-entropy loss in training.

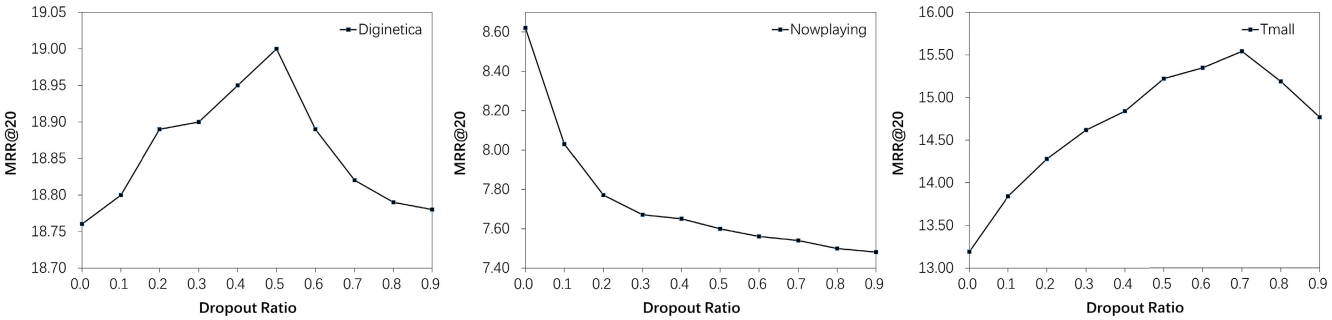

FIGURE 6. Impact of different dropout settings on recommended performance.

- AGNN-GC-T-C: AGNN-GC-T-C adopts the tanh-adding attention mechanism to process fused features and retains the conventional cross-entropy loss in training.
- AGNN-GC-L-C: AGNN-GC-L-C adopts the leakyrelu-adding attention mechanism to process fused features and retains the conventional cross-entropy loss in training.
- AGNN-GC-R-C: AGNN-GC-R-C adopts the relu-adding attention mechanism to process fused features and retains the conventional cross-entropy loss in training.
- AGNN-GC-D-C: AGNN-GC-D-C adopts the dot product attention mechanism to process fused features and retains the conventional cross-entropy loss in training.
- AGNN-GC-S-F: AGNN-GC-S-F adopts the sigmoid-adding attention mechanism to process fused features and applies the focal loss in training.
- AGNN-GC-T-F: AGNN-GC-T-F adopts the tanh-adding attention mechanism to process fused features and applies the focal loss in training.
- AGNN-GC-L-F: AGNN-GC-L-F adopts the leakyrelu-adding attention mechanism to process fused features and applies the focal loss in training.
- AGNN-GC-D-F: AGNN-GC-D-F adopts the dot product attention mechanism to process fused features and applies the focal loss in training.

It can be found from TABLE 3 our proposed method AGNN-GC performs best. Obviously, with the focal loss rather than the conventional cross-entropy loss in the training process, AGNN-GC achieves better performance on three datasets, especially on Nowplaying and Tmall datasets, which indicates that using the focal loss in the training process can better train the positive samples and samples that are difficult to be trained and classified. Besides, on Nowplaying and Tmall datasets, the relu-adding attention mechanism outperforms other methods with the same loss function in the training process, which suggests the superiority of the relu-adding attention mechanism to process fused features in information fusion module. On Diginetica dataset, the method that uses the relu-adding attention mechanism performs close to other methods with the same loss function in the training process, which may be because the average length of sessions in Diginetica dataset is shorter than that in the other two datasets. Therefore, it can help the information fusion module to compute soft-attention weights.

D. IMPACT OF FOCAL LOSS SETTING (RQ3)

In the training process, we use two hyperparameters, α and γ, to control the focal loss. From FIGURE 3, FIGURE 4, and FIGURE 5, taking the factor $\gamma = 2$ achieves the best performance than taking the factor γ from $\{0, 0.5, 1, 5\}$ on the three datasets. Due to the extreme imbalance between positive and negative samples in the training process, the model generally achieves better performance as α increases. On the one hand, when setting $\alpha = 0.25$, both HR@20 and MRR@20 reach the bottom, which is because it exacerbates the imbalance between positive and negative samples. On the other hand, when setting $\alpha = 0.9$, both HR@20 and MRR@20 reach the peak, but increasing α will degrade the performance of the model.

E. IMPACT OF DROPOUT SETTING (RQ4)

We apply the dropout regularization strategy to prevent our model from overfitting, referring to GCE-GNN [16]. Specifically, the dropout regularization strategy randomly drops neurons with probability p during training, where all neurons are in the test set. FIGURE 6 illustrates the impact of the dropout setting of (10) on Diginetica, Nowplaying, and Tmall datasets. It is easy to find that our model performs poorly when the dropout ratio is small on Diginetica and Tmall datasets. It reaches peak performance when the dropout ratio is 0.5 on Diginetica and 0.7 on Tmall because it is prone to overfitting on the two datasets. However, as the dropout ratio increases, its performance worsens because it is challenging to learn from data with few accessible neurons. Besides, it gets the best performance without any dropout setting on Nowplaying because it is challenging to overfit.

VI. CONCLUSION

This paper presents a novel approach for session-based recommendation based on graph neural networks. Specifically, it first constructs global and local graphs based on all training sequences. Next, it learns global-level and local-level item embedding information and fuses them to enhance the feature presentations of items by a novel attention mechanism. Finally, applying the focal loss to balance positive and negative samples on model training

accomplishes the prediction. Our experiments over three real-world datasets prove the superiority over most advanced methods.

REFERENCES

[1] J. B. Schafer, J. Konstan, and J. Riedl, "Recommender systems in e-commerce," in *Proc. 1st ACM Conf. Electron. Commerce*, Nov. 1999, pp. 158–166.
[2] G. Shani, D. Heckerman, R. I. Brafman, and C. Boutilier, "An MDP-based recommender system," *J. Mach. Learn. Res.*, vol. 6, no. 9, pp. 1–31, 2005.
[3] S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme, "Factorizing personalized Markov chains for next-basket recommendation," in *Proc. 19th Int. Conf. World Wide Web*, Apr. 2010, pp. 811–820.
[4] A. Zimdars, D. M. Chickering, and C. Meek, "Using temporal data for making recommendations," 2013, *arXiv:1301.2320*.
[5] A. Mnih and R. R. Salakhutdinov, "Probabilistic matrix factorization," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 20, 2007, pp. 1–8.
[6] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder–decoder for statistical machine translation," 2014, *arXiv:1406.1078*.
[7] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk, "Session-based recommendations with recurrent neural networks," 2015, *arXiv:1511.06939*.
[8] Y. K. Tan, X. Xu, and Y. Liu, "Improved recurrent neural networks for session-based recommendations," in *Proc. 1st Workshop Deep Learn. Recommender Syst.*, Sep. 2016, pp. 17–22.
[9] Y. Chen and Y. Tang, "Attentive capsule graph neural networks for session-based recommendation," in *Proc. Int. Conf. Knowl. Sci., Eng. Manag.* Cham, Switzerland: Springer, 2022, pp. 602–613.
[10] J. Li, P. Ren, Z. Chen, Z. Ren, T. Lian, and J. Ma, "Neural attentive session-based recommendation," in *Proc. ACM Conf. Inf. Knowl. Manag.*, Nov. 2017, pp. 1419–1428.
[11] Q. Liu, Y. Zeng, R. Mokhosi, and H. Zhang, "STAMP: Short-term attention/memory priority model for session-based recommendation," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2018, pp. 1831–1839.
[12] J. Song, H. Shen, Z. Ou, J. Zhang, T. Xiao, and S. Liang, "ISLF: Interest shift and latent factors combination model for session-based recommendation," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 5765–5771.
[13] S. Wu, Y. Tang, Y. Zhu, L. Wang, X. Xie, and T. Tan, "Session-based recommendation with graph neural networks," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 346–353.
[14] C. Xu, P. Zhao, Y. Liu, V. S. Sheng, J. Xu, F. Zhuang, J. Fang, and X. Zhou, "Graph contextualized self-attention network for session-based recommendation," in *Proc. IJCAI*, vol. 19, 2019, pp. 3940–3946.
[15] R. Qiu, J. Li, Z. Huang, and H. Yin, "Rethinking the item order in session-based recommendation with graph neural networks," in *Proc. 28th ACM Int. Conf. Inf. Knowl. Manag.*, Nov. 2019, pp. 579–588.
[16] 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 *Proc. 43rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2020, pp. 169–178.
[17] Y. Chen, Q. Xiong, and Y. Guo, "Session-based recommendation: Learning multi-dimension interests via a multi-head attention graph neural network," *Appl. Soft Comput.*, vol. 131, Dec. 2022, Art. no. 109744.
[18] L. Dong, G. Zhu, Y. Wang, Y. Li, J. Duan, and M. Sun, "A graph positional attention network for session-based recommendation," *IEEE Access*, vol. 11, pp. 7564–7573, 2023.
[19] B. Hidasi, M. Quadrana, A. Karatzoglou, and D. Tikk, "Parallel recurrent neural network architectures for feature-rich session-based recommendations," in *Proc. 10th ACM Conf. Recommender Syst.*, Sep. 2016, pp. 241–248.
[20] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–11.
[21] Y. Li, D. Tarlow, M. Brockschmidt, and R. Zemel, "Gated graph sequence neural networks," 2015, *arXiv:1511.05493*.
[22] F. Yu, Y. Zhu, Q. Liu, S. Wu, L. Wang, and T. Tan, "TAGNN: Target attentive graph neural networks for session-based recommendation," in *Proc. 43rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2020, pp. 1921–1924.
[23] Z. Wang, W. Wei, G. Cong, X.-L. Li, X.-L. Mao, M. Qiu, and S. Feng, "Exploring global information for session-based recommendation," 2020, *arXiv:2011.10173*.
[24] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, *arXiv:1609.02907*.
[25] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," 2017, *arXiv:1710.10903*.
[26] Y. Xie, Z. Li, T. Qin, F. Tseng, K. Johannes, S. Qiu, and Y. Lu Murphey, "Personalized session-based recommendation using graph attention networks," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2021, pp. 1–8.
[27] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, Jan. 2014.
[28] S. Hochreiter, "The vanishing gradient problem during learning recurrent neural nets and problem solutions," *Uncertainty Fuzziness Knowl.-Based Syst.*, vol. 6, no. 2, pp. 107–116, Apr. 1998.
[29] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proc. 14th Int. Conf. Artif. Intell. Statist.*, 2011, pp. 315–323.
[30] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2980–2988.
[31] E. Zangerle, M. Pichl, W. Gassler, and G. Specht, "#nowplaying music dataset: Extracting listening behavior from Twitter," in *Proc. 1st Int. Workshop Internet-Scale Multimedia Manag.*, Nov. 2014, pp. 21–26.
[32] L. Feng, Y. Cai, E. Wei, and J. Li, "Graph neural networks with global noise filtering for session-based recommendation," *Neurocomputing*, vol. 472, pp. 113–123, Feb. 2022.
[33] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proc. 10th Int. Conf. World Wide Web*, Apr. 2001, pp. 285–295.
[34] 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 *Proc. 42nd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2019, pp. 345–354.
[35] T. Chen and R. C.-W. Wong, "Handling information loss of graph neural networks for session-based recommendation," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2020, pp. 1172–1180.

YINGPEI CHEN was born in 1997. He is currently pursuing the M.S. degree with the School of Computer and Information Science, Southwest University, China. His research interests include data mining and recommender systems.

YAN TANG was born in 1965. She is mainly engaged in intelligent science and big data analysis technology research. In the face of mass data processing, she studies the theories and technologies related to artificial intelligence and big data analysis and applies them to web mining, natural language processing, and computer vision.

YUAN YUAN was born in 1996. She received the M.S. degree from Southwest University, China, in 2022. Her research interests include data mining and recommender systems.