Personalized Graph Neural Networks With Attention Mechanism for Session-Aware Recommendation

Mengqi Zhang, Shu Wu, Meng Gao, Xin Jiang, Ke Xu, and Liang Wang, *Fellow, IEEE*

Abstract—The problem of session-aware recommendation aims to predict users' next click based on their current session and historical sessions. Existing session-aware recommendation methods have defects in capturing complex item transition relationships. Other than that, most of them fail to explicitly distinguish the effects of different historical sessions on the current session. To this end, we propose a novel method, named Personalized Graph Neural Networks with Attention Mechanism (A-PGNN) for brevity. A-PGNN mainly consists of two components: one is Personalized Graph Neural Network (PGNN), which is used to extract the personalized structural information in each user behavior graph, compared with the traditional Graph Neural Network (GNN) model, which considers the role of the user when the node embedding is updated. The other is Dot-Product Attention mechanism, which draws on the Transformer net to explicitly model the effect of historical sessions on the current session. Extensive experiments conducted on two real-world data sets show that A-PGNN evidently outperforms the state-of-the-art personalized session-aware recommendation methods.

Index Terms—Graph neural networks, attention, session-aware recommendation

1 Introduction

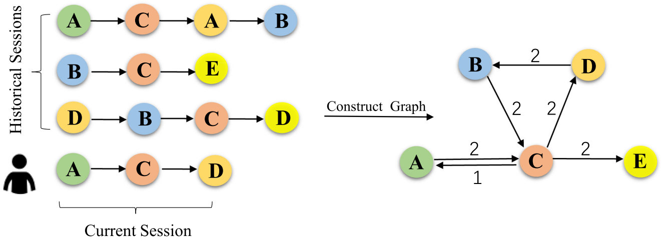
With the rapid growth of the amount of information on the Internet, recommender systems have become a fundamental technique to help users alleviate the problem of information overload. Usually, the user's interactions with the system within a given time frame are organized into a session. Predicting the next interaction for anonymous sessions is called session-based recommendation. Generally, the users' identifications and past behaviors can be utilized for the next-click prediction, which is called session-aware recommendation. In this scenario, users' recent behaviors in the current session often reflect their short-term preference, whereas historical session sequences imply the evolution of their long-term preferences over time. Combining the short- and long-term preferences of users have become a vital issue in the session-aware recommendation.

In recent years, most session-aware recommendation studies are conducted on the methods of session-based recommendation. Numerous researches based on deep learning [1], [2], [3] applied Recurrent Neural Networks (RNNs) in session-based recommendation scenarios and have obtained promising results. Attention networks are also a powerful tool to capture user interest in each session [4], [5]. Recently, graph-based models have gained increasing attention. SR-GNN [6] is the first to apply graph neural networks to capture complex item transition relationships in each session. However, these abovementioned session-based models can only leverage the current anonymous session to make the recommendation. Therefore, the session-aware recommenders came into being [7], [8], [9]. To capture the user's interest drift across sessions, recent session-aware works [10], [11] use a hierarchical RNN to capture the flow of user interest within and across sessions. The recently proposed HierTCN [12] employs a hierarchical architecture that contains GRU and Temporal Convolutional Network to capture both the long-term interests and short-term interactions.

Despite their effectiveness, we argue that these personalized models remain as the two critical limitations. First, personalized models have defects in capturing complex item transition relationships. Take Fig. 1 as an example. Fig. 1a represents all the interaction sessions of user u. Based on the transitions between items, the whole session can be converted into a graph as Fig. 1b, where B, C, and D compose a strongly connected component, which reflects their dense link relationships. Most existing sequence-based session-aware methods are challenging to capture that intricate pattern within and across sessions. Second, some

- *Mengqi Zhang, Shu Wu, and Liang Wang are with the Center for Research on Intelligent Perception and Computing (CRIPAC), Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, and also with the School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 101408, China. E-mail: mengqi.zhang@cripac.ia.ac.cn, {shu.wu, wangliang}@nlpr.ia.ac.cn.*
- *Meng Gao is with the School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing 100083, China. E-mail: gaomeng0527@icloud.com.*
- *Xin Jiang is with LMIB and School of Mathematics and Systems Science, Beihang University, Beijing 100083, China. E-mail: jiangxin@buaa.edu.cn.*
- *Ke Xu is with the State Key Laboratory of Software Development Environment, Beihang University, Beijing 100083, China. E-mail: kexu@nlsde.buaa.edu.cn.*

Manuscript received 26 Sept. 2019; revised 26 Aug. 2020; accepted 29 Sept. 2020. Date of publication 15 Oct. 2020; date of current version 7 July 2022.
(Corresponding author: Shu Wu.)
Recommended for acceptance by Raymond Chi-Wing Wong.
Digital Object Identifier no. 10.1109/TKDE.2020.3031329



(a) User interaction sessions (b) User behavior graph

Fig. 1. Construction of user behavior graph. In (a), user's interaction sessions includes historical sessions and current session. (b) is a schematic diagram of the user behavior graph. The number on the directed edge represents the number of times it appears.

session-aware methods fail to explicitly distinguish the effects of different historical user sessions on the current session. As a motivating example, suppose a user previously browsed or clicked a digital camera on a shopping site, and his current clicked items are SD and Micro-SD cards. In this case, there is a strong relationship between the historical session and the current session item. If his current interaction is with automotive products, the previously clicked items and the automotive fall into two unrelated categories, which shows that the historical sessions have a minor effect on the current session. However, recent work HierTCN and H-RNN encode historical user sessions into the initial representation of session-level TCN or RNN to assist in making predictions. They all ignore the fine-grained impact of historical sessions on the current session, making the model insufficient to utilize historical information.

To address these limitations, we consider improving the construction of session-graph in SR-GNN model [6]: building a personalized graph according to both user's current and historical sessions, as shown in Fig. 1b, called user behavior graph. To further reinforce associations between different sessions of each user, we design a personalized graph neural network (PGNN) that considers the role of the user when the node embedding is updated. Furthermore, we use the attention mechanism of Transformer [13] to model the explicit effect of historical session on each item of the current session.

To sum up, in this article, we propose a novel method Personalized Graph Neural Networks with Attention Mechanism (A-PGNN). It contains two main components: PGNN and Dot-Product Attention mechanism. We first convert all sessions of each user into a graph, and then feed it into PGNN and Dot-Product net in sequence. Fig. 2 illustrates the workflow of the proposed A-PGNN model. The details are introduced in Section 3. Extensive experiments conducted on real-world representative data sets demonstrate the effectiveness of the proposed method over the state-of-the-art methods. The main contributions of this work are summarized as follows:

- We design a new graph neural network PGNN for personalized recommendation scenario, which is able to capture complex item transitions in user-specific fashion.
- We use the attention mechanism to explicitly model the effect of the user's historical interests on the current session, which shows the superiority of our model in the session-aware recommendation task.
- We conduct empirical studies on two real-world data sets. Extensive experiments demonstrate the effectiveness of our proposed model and the contribution of each component.

2 Related Work

2.1 Session-Based Recommendation

Matrix factorization [14], [15], [16] is a general approach used in recommendation systems. The basic objective is to factorize a user-item rating matrix into two low-rank matrices, and each of them represents the latent factors of users or items. The item-based neighborhood methods [17] are a natural solution, in which item similarities are calculated on the co-occurrence in the same session. These methods have difficulty in considering the sequential order of items and generate prediction merely based on the last click. Then, the sequential methods based on Markov chains are proposed, which predict the users' next behavior based on the previous ones. Treating recommendation generation as a sequential optimization problem, [18] uses Markov decision processes (MDPs) for the solution. Via factorization of the personalized probability transition matrices of users, FPMC [19] models sequential behavior between every two adjacent clicks and provides a more accurate prediction for each sequence. The main drawback of Markov-chain-based models is that they combine past components independently. Such an independence assumption is too strong, and thus confines the prediction accuracy.

Recently, deep neural networks have become the most successful methods in modeling sequence, such as machine translation[20], [21], [22], and conversation machine[23]. For session-based and sequential recommendation, the work of [1] proposes the recurrent neural network approach, and then extends to an architecture with parallel RNNs [24], which could model sessions based on the clicks and features of the clicked items. After that, some work are proposed based on these RNN methods. An improved RNN [2] enhances the performance of recurrent model by using proper data augmentation techniques and taking temporal shifts in user behavior into account. The work of [25] combines the recurrent method with the neighborhood-based method together to mix the sequential patterns as well as the co-occurrence signals. What is more, convolutional neural networks are also used in sequential recommendations to incorporate session clicks with content features[26].

Furthermore, a neural attentive recommendation machine with an encoder-decoder architecture, that is, NARM [27], utilizes the attention mechanism on RNN to capture the users' features of sequential behavior and main purposes. SHAN model [5] uses a two-layer hierarchical attention network, which takes the long- and short-term preferences into account. Then, a short-term attention priority model (STAMP) [4] using a novel attention memory network, is proposed to efficiently capture both the users' general interests and current interests.

However, these abovementioned session-based or sequential models can only leverage the current anonymous session or single sequence to make the recommendation.

2.2 Session-Aware Recommendation

In session-aware recommendation scenarios, the user behavior in past sessions might provide valuable information for

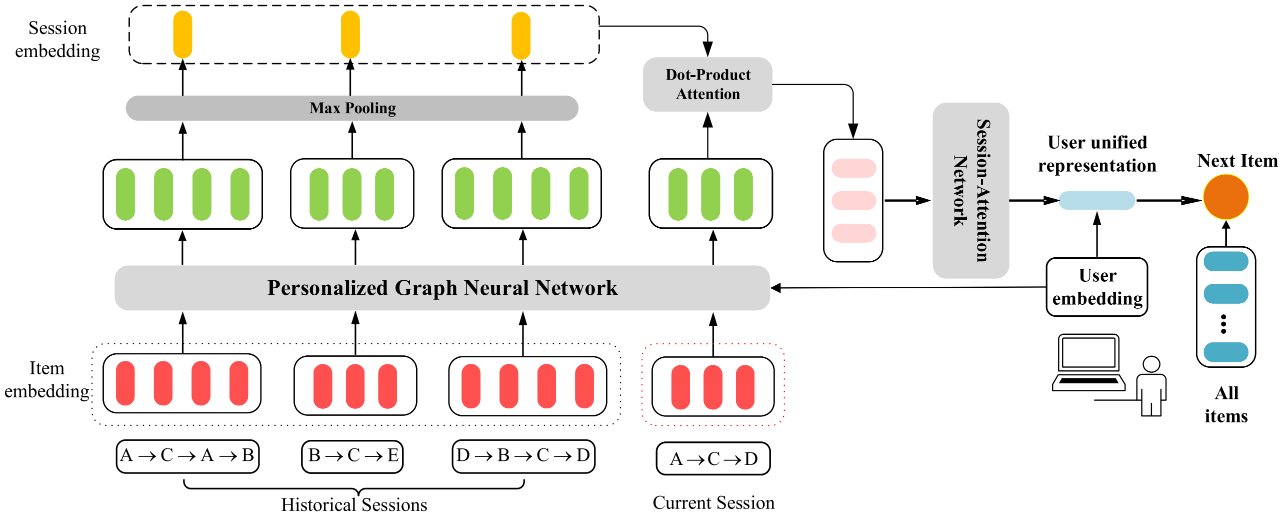


Fig. 2. The framework of A-PGNN. Based on the user's all sessions, we first construct a user behavior graph. We then input the user behavior graph along with item and user embedding into PGNN to obtain the item representations. Next, we utilize the max-pooling layer to get the historical session embedding. Following this, the representation of the current session that incorporates the user's historical preferences can be obtained by the Dot-product attention component. Finally, we utilize the Session-Attention network to generate the user's dynamic representation and combine it with the user's static embedding to get a unified user representation for final prediction.

providing recommendations in the next session. In [7], RNN-based approaches are proposed, which leverage additional item features to enhance recommendation capacity. A list-wise deep neural network [28] models the limited user behavior within each session and uses a list-wise ranking model to generate the recommendation for each session. The work of [29] proposes some strategies to integrate user expression with RNN models. However, they all fail to effectively use the user's historical information. To this end, the work[10] uses a hierarchical RNN to capture users' short- and long-term preferences for personalized session-based recommendation. Analogous to model Hierarchical RNN, II-RNN model [11] also utilizes multiple RNN to model interest relationships within current and historical sessions. DANN [30] exploits a dual attentive neural network to model user's personalized preference and primary purpose in his all sessions. The recently proposed HierTCN [12] employs a hierarchical architecture that contains GRU and Temporal Convolutional Network to capture both the long-term interests and short-term interactions. However, their encoding mechanism limits the model's capabilities. In addition, it is difficult to fully capture the complex patterns of user behavior by relying solely on the sequence relationship of the session.

2.3 Graph Neural Networks

Nowadays, neural network has been used for generating representation for graph-structured data, for example, social networks and knowledge bases. On one hand, extending the word2vec [20], an unsupervised algorithm DeepWalk [31] is designed to learn representations of graph nodes based on random walk. Following DeepWalk, unsupervised network embedding algorithms LINE [32] and node2vec [33] are most representative methods. On the other hand, the classical neural network CNN and RNN are also deployed on graph-structured data. Duvenaud *et al.* [34] introduce a convolution neural network that operates directly on graphs of arbitrary sizes and shapes. A scalable approach [35] chooses the convolution architecture via a localized approximation of spectral graph convolutions, which is an efficient variant, and it could operate on graphs directly as well. However, these methods can only be implemented on undirected graphs. Previously, in the form of recurrent neural networks, Graph Neural Networks (GNNs) [36], [37] are proposed to operate on directed graphs. As a modification of GNN, Gated Graph Neural Networks [38] uses gated recurrent units and employs back-propagation through time (BPTT) to compute gradients. Graph Attention Networks (GAT) [39] applies the attention mechanism to learn the weight of nodes and neighbor nodes. Recently GNN is broadly applied for the different tasks, for example, script event prediction [40], situation recognition [41], recommender system[6], and image classification [42].

GNN has advantages in processing graph structure data and can be used to capture more abundant information in sequence data. SR-GNN [6] is the first model to utilize the Gated Graph Neural Networks to capture the complex item transition relationships in session-based recommendation scenarios, but it ignores the role of user in item transition relationship, and fails to use user historical session information to improve recommendation performance. In this work, we propose a model A-PGNN based on improved GGNN, which is more suitable for personalized session-based recommendation scenarios.

3 The Proposed Method

In this section, we introduce the proposed A-PGNN[1] which applies personalized graph neural networks along with attention mechanism for session-aware recommendation. First, the problem is formulated. Then, we introduce the overview of our proposed method and give more details.

3.1 Problem Formulation

Let $V = \{v_i\}_{i=1}^{|V|}$ and $U = \{u_i\}_{i=1}^{|U|}$ be the set of items and users in the system, respectively. We describe item v_i with

[1] https://github.com/CRIPAC-DIG/A-PGNN

TABLE 1
Important Notations

Notation	Description
V	the set of items
U	the set of users
$\mathcal{S}^u$	user u's all sessions set
S_i^u	the ith session of user u's all sessions
$\mathcal{S}_{\text{h}}^u$	the historical sessions of user u
$\mathcal{S}_{\text{c}}^u$	the current session of user u
$\mathcal{G}^u$	the user behavior graph of user u
$\mathbf{e}_{v_i}$	the embedding of item v_i
$\mathbf{e}_{u_i}$	the embedding of user u_i
$\mathbf{z}_u$	the unified representation of user u

embedding vector $\mathbf{e}_{v_i} \in \mathbb{R}^d$ and user u_i with embedding vector $\mathbf{e}_{u_i} \in \mathbb{R}^{d'}$. d and d' are the dimensions of item and user embedding, respecively. For each user u, each session $S_i^u = \{v_{i,j}\}_{j=1}^{m_i} \in \mathcal{S}^u$ represents the sequence in the order of time occurrence, $v_{i,j} \in V$ represents an interactive item of the user within the session S_i^u, and m_i is the number of items in session S_i^u. All sessions of user u can be represented as $\mathcal{S}^u = \{S_i^u\}_{i=1}^{n_u}$, where n_u stands for the total number of sessions for a user u, for convenience, n_u is abbreviated as n and $\mathcal{S}^u = \{S_i^u\}_{i=1}^{n}$. Sessions and items are all ordered by timestamps. In general, the last interacted session, that is, $\{S_n^u\}$, is called current session. Other sessions that belong to $\{S_i^u\}_{i=1}^{n-1}$ are called historical sessions. Current session and historical session are denoted as $\mathcal{S}_{\text{c}}^u, \mathcal{S}_{\text{h}}^u$ respectively. Given users' all sessions $\mathcal{S}^u$, the goal of session-aware recommendation is to predict the next interactive item $v_{n,m+1}$ of the current session $\mathcal{S}_{\text{c}}^u$. The notations used throughout this paper are summarized in Table 1.

3.2 Overview

Fig. 2 is the overview of our proposed method. For each user u, all sessions $\mathcal{S}^u$ can be modeled as a user behavior graph $\mathcal{G}^u$ (Section 3.3). Then, $\mathcal{G}^u$ is fed into Personalized Graph Neural Network (PGNN) (Section 3.4) to capture transitions of items with respect to user u. After that, we use the max-pooling layer to get the session embedding, therefore, the historical session embedding matrix can be obtained. Then, we evaluate the explicit impact of the historical session on the current session through Dot-Product attention layer (Section 3.5.1). Thereby, we can get user's dynamic interest representations and concatenate it with users' embedding to obtain a unified representation (Section 3.5.2). Using the representations, we output probability $\hat{\mathbf{y}}$ for all candidate items, where the element $y_i \in \hat{\mathbf{y}}$ is the recommendation score of the corresponding item $v_i \in V$ (Section 3.6). The items with top-k values will be the candidate items for recommendation.

3.3 User Behavior Graph

To fully capture the complex item transitions of each user, inspired by SR-GNN [6], we construct graph $\mathcal{G}^u$ for each user. As shown in Fig. 1a and 1b, for each user u, we model all of his/her sessions $\mathcal{S}^u$ as a directed graph $\mathcal{G}^u = (\mathcal{V}^u, \mathcal{E}^u)$. In each user behavior graph $\mathcal{G}^u$, node i represents an item $v_i \in V$ that user u interacted with. The edge $v_j \to v_i$ represents a user interacts item v_i after v_j in one of his sessions. For this case, we assume that the effect of v_i on v_j and the effect of v_j on v_i are different, which produces two types of edges that represent two different transition relationships. One directed edge called outgoing edge with weights of ω_{ij}^{out} and the other directed edge called incoming edge with weights of ω_{ji}^{in}. Their weights can be computed by

$$\omega_{ij}^{\text{out}} = \frac{\text{Count}(v_i, v_j)}{\sum_{v_i \to v_k} \text{Count}(v_i, v_k)}, \tag{1}$$

$$\omega_{ij}^{\text{in}} = \frac{\text{Count}(v_j, v_i)}{\sum_{v_k \to v_i} \text{Count}(v_k, v_i)}, \tag{2}$$

where function $\text{Count}(x, y)$ is used to calculate the number of occurrences that user interacts item y after interacts item x. The topological structure of user behavior graph $\mathcal{G}^u$ can be represented by two adjacency matrices, which can be written as

$$\mathbf{A}_u^{\text{out}}[i, j] = \omega_{ij}^{\text{out}}, \tag{3}$$

$$\mathbf{A}_u^{\text{in}}[i, j] = \omega_{ij}^{\text{in}}, \tag{4}$$

$\mathbf{A}_u^{\text{out}}$ and $\mathbf{A}_u^{\text{in}}$ represent adjacency matrix of outgoing and incoming edges in user behavior graph.

3.4 Personalized Graph Neural Network

SR-GNN [6] is the first model to apply GNNs in the session-based recommendation, which feeds the session graphs containing rich node connections into GGNN to automatically extract useful features of items. However, the GGNN used in SR-GNN is not suitable for personalized recommendation because it fails to inject the user's information into the graph model. To address this limitation, herein, we present personalized graph neural networks (PGNN), which is used to learn the complex item transition relationships between items interacted by each user, and then obtain the representation of items and users.

Various users have different behavior patterns, which results in different item transition relationships for each user. So, we consider the user factor when designing PGNN architecture. At each time of node update, we fuse user embedding $\mathbf{e}_u$ with the current representation of node $\mathbf{h}_i^{t-1}$. For example, at t time, the aggregated incoming and outcoming information of node i can be formulated as

$$\mathbf{a}_{\text{out}_i}^{(t)} = \sum_{v_i \to v_j} \mathbf{A}_u^{\text{out}}[i, j] \Big[\mathbf{h}_j^{(t-1)} \parallel \mathbf{e}_u\Big] \mathbf{W}_{\text{out}} \tag{5}$$

$$\mathbf{a}_{\text{in}_i}^{(t)} = \sum_{v_j \to v_i} \mathbf{A}_u^{\text{in}}[i, j] \Big[\mathbf{h}_j^{(t-1)} \parallel \mathbf{e}_u\Big] \mathbf{W}_{\text{in}}, \tag{6}$$

$$\mathbf{a}_i^{(t)} = \mathbf{a}_{\text{out}_i}^{(t)} \parallel \mathbf{a}_{\text{in}_i}^{(t)}, \tag{7}$$

where $\parallel$ is the concatenation operation. Because $\mathcal{G}^u$ is bidirectional, to embed bidirectional propagation information, we consider two parameters, $\mathbf{W}_{\text{in}}$ and $\mathbf{W}_{\text{out}} \in \mathbf{R}^{(d+d')\times\hat{d}}$, which transform the user and item connection vectors to two different $\hat{d}$-dimensional vectors, respectively. All users share parameters $\mathbf{W}_{\text{in}}$ and $\mathbf{W}_{\text{out}}$.

Then, we use gated recurrent units (GRUs) [22] to incorporate information from other nodes with hidden states of the previous timestep, and update each node's hidden state

$$\mathbf{z}_i^{(t)} = \sigma\Big(\mathbf{W}_z\mathbf{a}_i^{(t)} + \mathbf{U}_z\mathbf{h}_i^{(t-1)}\Big), \tag{8}$$

$$\mathbf{r}_i^{(t)} = \sigma\Big(\mathbf{W}_r\mathbf{a}_i^{(t)} + \mathbf{U}_r\mathbf{h}_i^{(t-1)}\Big), \tag{9}$$

$$\widetilde{\mathbf{h}_i^{(t)}} = \tanh\Big(\mathbf{W}_o\mathbf{a}_i^{(t)} + \mathbf{U}_o\Big(\mathbf{r}_i^{(t)} \odot \mathbf{h}_i^{(t-1)}\Big)\Big), \tag{10}$$

$$\mathbf{h}_i^{(t)} = \Big(1 - \mathbf{z}_i^{(t)}\Big) \odot \mathbf{h}_i^{(t-1)} + \mathbf{z}_i^{(t)} \odot \widetilde{\mathbf{h}_i^{(t)}}, \tag{11}$$

where $\mathbf{z}_i^t$ and $\mathbf{r}_i^t$ are update and reset gate, $\sigma(\cdot)$ is the sigmoid function, and $\odot$ is the element-wise multiplication operator. $\mathbf{W}_z$, $\mathbf{U}_z$, $\mathbf{W}_r$, $\mathbf{U}_r$, $\mathbf{W}_o$, $\mathbf{U}_o$ are GRU parameters shared by all users. After a total of T propagation steps, the final hidden state vector $\mathbf{h}_i^{(T)}$ of each node i can be obtained in graph $\mathcal{G}^u$. For convenience, we use $\mathbf{h}_i$ instead of $\mathbf{h}_i^{(T)}$. The final hidden state of each node not only contains its node features, but also aggregates the information from its T-order neighbors.

Similar to most graph-based models [6], [43], PGNN is suitable for the scenarios that the user repeatedly click the same items within sessions or across sessions. When the same user repeatedly clicks the same items across sessions, all sessions of the user can convert to a fully connected graph structure. Obviously, PGNN can capture the items transition pattern across sessions. For the other extreme case, the same user has no repeated interaction items across sessions. In this case, the user's behavior graph contains many disconnected sub-graphs, where each sub-graph corresponds to a session. Since all sessions of each user in PGNN share the same user embedding, each sub-graph in user's behavior graph can be related through the user embedding when the node embedding updates. Therefore, PGNN can still capture the association across sessions.

3.5 Generating User's Unified Representation via Attention Networks

In this section, we propose to use the Transformer's attention mechanism [13] to calculate the explicit effect of historical sessions on the current session and then get the dynamic representation of each user through the attention network. Finally, the user's unified representation can be obtained for personalized recommendation.

3.5.1 Calculating the Impact of Historical Sessions on the Current Session

In our model, we resort to Transformer network [13] which is widely used in some popular neural machine translation models to complete the calculation of the impact of historical sessions on current session. The scaled dot-product attention mechanism is the core of Transformer network.

Transformer Attention. The input of Transformer attention consists of queries and keys of dimension d_k, and values of dimension d_v. We compute the dot products of the *query* with all *keys* divide each by $\sqrt{d_k}$, then apply a softmax function to obtain the weight on values. The scaled dot-product attention is formally defined as

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}}\right)\mathbf{V}, \tag{12}$$

where $\mathbf{Q}, \mathbf{K}, \mathbf{V}$ represent the queries, keys, and values respectively, and the scale factor $\sqrt{d}$ is to avoid exceedingly large dot products and speed up convergence.

The embedding representations of user u's historical sessions $\mathcal{S}_{\text{h}}^u$ and current session S_{C}^u can be obtained through the output of PGNN. The embedding vector of historical session $S_i^u = \{v_{i,1}, v_{i,2}, \ldots, v_{i,m_i}\}$ in $\mathcal{S}_{\text{h}}^u$ can be represented as $\mathbf{f}_i^u \in \mathbb{R}^d$, which can be calculated by max-pooling

$$\mathbf{f}_{i,j}^u = \max_{1\le j\le d}\big(\mathbf{h}_{1,j}, \mathbf{h}_{2,j}, \ldots, \mathbf{h}_{m_i,j}\big). \tag{13}$$

Therefore, historical session sequence $\mathcal{S}_{\text{h}}^u = \{S_1^u, S_2^u, \ldots, S_{n-1}^u\}$ can be expressed as an embedded matrix $\mathbf{F}^u = \big[\mathbf{f}_1^u, \mathbf{f}_2^u, \ldots, \mathbf{f}_{n-1}^u\big]$. For current session S_{C}^u, we simply denote the embedding matrix as $\mathbf{H}^u = [\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_m]$. In our context, we use current session embedding to query historical session embedding, where the queries $\mathbf{Q}$ are determined by $\mathbf{H}^u$, the keys and values are determined by $\mathbf{F}^u$. In specical, we project $\mathbf{F}^u$ and $\mathbf{H}^u$ to the same latent space through nonlinear transformation

$$\begin{aligned} \mathbf{Q}^u &= \text{Relu}\big(\mathbf{H}^u\mathbf{W}^Q\big), \\ \mathbf{K}^u &= \text{Relu}\big(\mathbf{F}^u\mathbf{W}^K\big), \\ \mathbf{V}^u &= \text{Relu}\big(\mathbf{F}^u\mathbf{W}^V\big), \end{aligned} \tag{14}$$

where $\mathbf{W}^Q, \mathbf{W}^K, \mathbf{W}^V \in \mathbb{R}^{d\times d}$ are the projection matrices and shared by all users. The effect of the historical sessions on current session can be calculated by

$$\mathbf{H}_h = \text{Attention}(\mathbf{Q}^u, \mathbf{K}^u, \mathbf{V}^u). \tag{15}$$

After the effect of history session on each item in the current session sequence is calculated, we then compute the embedding of the current session as follows:

$$\mathbf{H}^{u'} = \mathbf{H}_h + \mathbf{H}^u. \tag{16}$$

Then, the current session embedding $\mathbf{H}^{u'}$ can be rewritten as $\big[\mathbf{h}_1', \mathbf{h}_2', \ldots, \mathbf{h}_m'\big]$.

3.5.2 Generating the User's Unified Representation

The current session embedding $\mathbf{H}^{u'}$ combines long- and short-term interests of users. In the following part, we describe how to encode $\mathbf{H}^{u'}$ to the user unified representation vector for next-item recommendation task.

Similar to SR-GNN [6], we first use the attention mechanism to encode the current embedding matrix to local representation and global representation, respectively, where local representation $\mathbf{z}_l$ denotes the user's recent interest and global representation $\mathbf{z}_g$ denotes the user's general interest. $\mathbf{z}_l$ can be simply defined as $\mathbf{h}_m'$, which is the embedding of last clicked item within the current session. $\mathbf{z}_g$ is defined as

$$\mathbf{z}_g = \sum_{i=1}^{m} \alpha_i \mathbf{h}'_i, \tag{17}$$

$$\alpha_i = \mathbf{W}_0 \sigma\big(\mathbf{W}_1 \mathbf{h}'_m + \mathbf{W}_2 \mathbf{h}'_i + \mathbf{b}_c\big), \tag{18}$$

where parameters $\mathbf{W}_0 \in \mathbb{R}^d$, $\mathbf{W}_1, \mathbf{W}_2 \in \mathbb{R}^{d \times d}$ control the weights of item embedding vectors, $\mathbf{b}_c \in \mathbb{R}^d$ is a bias vector, $\sigma(\cdot)$ denotes the sigmoid function and weighted coefficient α_i determines the weights of items of current session when making predictions. After that, we compute the user's dynamic representation $\mathbf{z}_d$ as follows,

$$\mathbf{z}_d = \mathbf{z}_g \parallel \mathbf{z}_l. \tag{19}$$

The embedding $\mathbf{e}_u$ implies the inherent attributes of the user and can be regarded as a static representation. So, we concatenate the dynamic and the static representation into one vector, then get the unified representation of users z_u through linear transformation

$$\mathrm{z}_u = \mathbf{B}[\mathbf{z}_d \parallel \mathbf{e}_u], \tag{20}$$

where matrix $\mathbf{B} \in \mathbb{R}^{d \times (2d+d')}$ compresses two combined embedding vectors into the latent space $\mathbb{R}^d$, and d, d' are the dimension of item and user embedding respectively.

3.6 Making Recommendation

After obtaining the unified representation of user u, we compute the recommendation score $\hat{\mathbf{z}}_i$ for each item $v_i \in V$. The score function is defined as

$$\hat{\mathbf{z}}_i = {\mathbf{z}_u}^{\top} \mathbf{e}_{v_i}, \tag{21}$$

where $\mathbf{z}_u$ and $\mathbf{e}_{v_i}$ denote the user's unified representation and item v_i's embedding, respectively. Then we apply a $\mathrm{softmax}$ function to get the output vector

$$\hat{\mathbf{y}} = \mathrm{softmax}(\hat{\mathbf{z}}), \tag{22}$$

where $\hat{\mathbf{z}} \in \mathbb{R}^{|V|}$ denotes the recommendation scores over all candidate items V and $\hat{\mathbf{y}}$ denotes the probabilities that items will be interacted by user u in the next time of the current session S^u_c.

For any user behavior graph, the loss function is defined as the cross-entropy of the prediction and the ground truth. It can be written as follows,

$$\mathcal{L}(\hat{\mathbf{y}}) = -\sum_{i=1}^{|V|} \mathbf{y}_i \log(\hat{\mathbf{y}}_i) + (1 - \mathbf{y}_i) \log(1 - \hat{\mathbf{y}}_i), \tag{23}$$

where $\mathbf{y}$ denotes the one-hot encoding vector of the ground truth item. Finally, we use the back-propagation through time (BPTT) algorithm to train the proposed A-PGNN.

4 Experiments

We first describe the experimental setting from Sections 4.1, 4.2, 4.3, and 4.4, and then compare A-PGNN against state-of-the-art methods in Section 4.5. To verify the effectiveness of two important components in our model, we perform ablation studies in Section 4.6. In Section 4.7, we further give analysis about the effect of test session's characteristics on the model's performance. The hyper-parameter study is finally presented in Section 4.8. We intend to answer the following questions through experiments.

- *RQ1*: How does A-PGNN perform compared with other SOTA models?
- *RQ2*: What is the effect of various components in A-PGNN?
- *RQ3*: How does A-PGNN perform on test sessions with varied lengths and numbers of historical sessions?
- *RQ4*: What are the effects of different hyper-parameter settings (parameter initialization methods, maximum historical session and PGNN propagation step) on A-PGNN?

4.1 Data Sets

We used two different real-word data sets for our experiments. The first is the Xing data [10], which is released from RecSys Challenge 2016.[2] The second is a data set [11] extracted from the social news and discussion website Reddit.[3]

Xing. The Xing data set contains interactions on job postings for 770,000 users over an 80-day period. In these data, user behaviors include click, bookmark, reply, and delete. Following the preprocessing procedure of [10]: We split the Xing data into session by 30-minute idle threshold and discarded interactions having typed "delete." Also discarded are repeated interactions of the same type within sessions to reduce noise (e.g., repeated clicks on the same job posting within a session). Removed sessions having less than 3 interactions to filter too short and poorly informative sessions, and kept users having 5 sessions or more to have sufficient cross-session information.

Reddit. The Reddit data set contains tuples of user name, a subreddit where the user makes a comment to a thread, and a timestamp for the interaction. We split each user's records into sessions manually by using the same approach as mentioned in [11]. The time threshold turn to be 60-minutes this time.

Then we preprocessed both data sets as follows: For each user, we hold the first 80 percent of his sessions as the training set. The remaining 20 percent constitutes the test set. We tune the hyper-parameters of the algorithms on the last 10 percent of the training set. The statistics of two data sets after the preprocessing steps are shown in Table 2. Referring to [6], we segment each user's sessions $\mathcal{S}^u$ into a series of sequences and labels. For example, for an input $\mathcal{S}^u = \{\{v_{1,1}, v_{1,2}, v_{1,3}\}, \{v_{2,1}, v_{2,2}\}, \{v_{3,1}, v_{3,2}, v_{3,3}\}\}$ of user u, where $\mathcal{S}^u_{\mathrm{h}} = \{\{v_{1,1}, v_{1,2}, v_{1,3}\}, \{v_{2,1}, v_{2,2}\}\}$, $\mathcal{S}^u_{\mathrm{c}} = \{\{v_{3,1}, v_{3,2}, v_{3,3}\}\}$. We generate historical sessions, current sessions, and labels, $\mathcal{S}^u_{\mathrm{h}_1} = \{\{v_{1,1}, v_{1,2}, v_{1,3}\}\}$, $\mathcal{S}^u_{\mathrm{c}_1} = \{\{v_{2,1}\}\}$, $\mathrm{label}_1 = v_{2,2}$; $\mathcal{S}^u_{\mathrm{h}_2} = \{\{v_{1,1}, v_{1,2}, v_{1,3}\}, \{v_{2,1}, v_{2,2}\}\}$, $\mathcal{S}^u_{\mathrm{c}_2} = \{\{v_{3,1}\}\}$, $\mathrm{label}_2 = v_{3,2}$; $\mathcal{S}^u_{\mathrm{h}_3} = \{\{v_{1,1}, v_{1,2}, v_{1,3}\}, \{v_{2,1}, v_{2,2}\}\}$, $\mathcal{S}^u_{\mathrm{c}_3} = \{\{v_{3,1}, v_{3,2}\}\}$, $\mathrm{label}_3 = v_{3,3}$, where the label is the next interacted item within the current session.

4.2 Compared Methods

We compared the performance of our proposed A-PGNN with nine compared methods, including conventional methods and deep neural methods.

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

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

TABLE 2
Statistics of Data Sets After Preprocessing

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

- *POP* recommends the top K frequent items in the training set.
- *Item-KNN* [17] computes an item-to-item cosine similarity based on the co-occurrence of items within sessions.
- *FPMC* [19] is a sequential prediction method based on the personalized Markov chain.
- *SKNN* [44] selects the K most similar sessions from the training set to retrieve candidate items for recommendation.
- *VSKNN* [45] is a sequential extension based SKNN.
- *GRU4Rec* [2] applies improved RNNs in session-based recommendation scenario.
- *SR-GNN* [6] utilizes the Gated Graph Neural Networks to capture the complex transition relationships of items for the session-based recommendation.
- *H-RNN* [10], [11] use a hierarchical RNNs consisting of a session-based and a user-level RNN to model the cross-session evolution of the user's interest. Due to [10] and [11] are similar in model architecture, we only select the best results of the two models as a comparison, collectively referred to as H-RNN.
- *HierTCN* [12] utilizes the hierarchical architecture that contains RNN and Temporal Convolutional Network to capture both the long-term interests and short-term interactions.

4.3 Evaluation Metrics

Following the metrics are used to evaluate each methods, which are also widely used in other related works [10], [11].

Recall@K (Precision) is widely used as a measure for predictive accuracy. It represents the proportion of correctly recommended items among the top-K items.

MRR@K (Mean Reciprocal Rank) is the average of reciprocal ranks of the correctly-recommended items. When all the rank positions exceed K, the reciprocal rank is set to 0. The MRR measure considers the order of recommendation ranking, where large MRR value indicates that correct recommendations are at the top of the ranking list.

We used Recall@K and MRR@K with $K = 5, 10, 20$ to evaluate all compared methods.

4.4 Parameter Setup

We set the dimension of item embedding $d = 100$ for Xing as [10], $d = 50$ for Reddit as [11], and set user embedding dimension $d' = 50$ for both data sets. According to the data processing method of Section 4.1, the maximum length of current session is 20. Because of the limitation of computing resources, for each user, we limit the number of historical sessions that feed into our model, that is, only his M most recent historical sessions can be utilized to assist with making prediction for current session. We set the M to a be a hyper-parameter named as ″maximum historical session″. For Xing, M is 50, whereas for Reddit, it is 30. As for the PGNN's propagation step T, we set T to be 1 for Xing and 3 for Reddit. All parameters are initialized by using Uniform distribution $\mathcal{U}(-1/\sqrt{d}, 1/\sqrt{d})$. The model is trained with Adam [46] optimizer, with learning rate 0.001. The coefficient of L2 normalization is set to 0, and the batch size is 100. In particular, we use batch normalization [47] between dot-attention layer and session-attention layer to prevent from overfitting on smaller Xing data. For the baseline methods, we use the default hyperparameters except for dimensions. We run the evaluation 5 times with different random seeds and report the mean value per algorithm.

4.5 Performance Comparison (RQ1)

First, for question *RQ1*, we compare it with other state-of-the-art personalized and pure session-based recommendation

TABLE 3
Performance of A-PGNN and Nine Compared Models in Terms of Recall@5, 10, 20, and Mrr@5, 10, 20 on Two Data Sets

Data	Xing						Reddit					
	Recall@5	Recall@10	Recall@20	Mrr@5	Mrr@10	Mrr@20	Recall@5	Recall@10	Recall@20	Mrr@5	Mrr@10	Mrr@20
Pop	0.21	0.26	0.58	0.08	0.09	0.11	13.22	19.46	26.47	8.50	9.32	9.82
Item-KNN	8.79	11.85	14.67	5.01	5.42	5.62	21.71	30.32	38.85	11.74	12.88	13.49
FPMC	1.70	2.42	3.27	0.61	0.50	0.37	29.91	34.31	44.32	8.78	6.56	4.54
SKNN	14.36	19.42	24.12	9.29	9.8	10.22	34.29	42.17	49.68	19.11	20.16	20.68
VSKNN	14.46	19.60	24.25	9.48	10.07	10.39	34.25	42.17	49.67	19.09	20.14	20.67
GRU4Rec	10.35	13.15	15.30	5.94	6.36	6.69	33.72	41.73	50.04	24.36	25.42	26.00
SR-GNN	13.38	16.71*	19.25	8.95	9.39	9.64	34.96	42.38	50.33	25.90	26.88	27.44
H-RNN	10.74	14.36	17.64	7.22	7.78	8.83	44.76	53.44	61.80	32.13	33.29	33.88
HierTCN	13.57*	16.55	19.93*	9.23*	9.48*	10.23*	47.15*	55.37*	63.96*	32.18*	33.79*	34.27*
A-PGNN	**14.38**	**17.06**	**19.98**	**10.36**	**10.71**	**10.91**	**49.19**	**59.43**	**68.00**	**33.54**	**34.92**	**35.52**
Improvement$^-$	-	-	-	9.28%	6.36%	5.00%	43.45%	40.93%	36.88%	85.96%	73.21%	71.76%
Improvement$*$	5.97%	2.09%	0.25%	12.24%	12.97%	6.65%	4.33%	7.33%	6.32%	4.23%	3.34%	3.65%

*The * and underlined numbers mean the best results on traditional and deep neural methods, respectively. Improvement$^-$ means improvement over the best conventional methods. Improvement* means improvement over the best deep neural methods.*

TABLE 4
Performance of A-PGNN Compared With Four Ablation Models in Terms of Recall@5,10 and Mrr@5,10 on Two Data Sets

Datasets	Xing				Reddit			
	Recall@5	Recall@10	Mrr@5	Mrr@10	Recall@5	Recall@10	Mrr@5	Mrr@10
A-PGNN(**-U)**	14.20(-1.25%)	16.69(-2.16%)	10.29(-0.68%)	10.81(-0.91%)	48.97(-0.46%)	59.24(-0.32%)	33.37(-0.51%)	34.75(-0.49%)
A-PGNN(**-A)**	11.89(-17.3%)	14.74(-13.6%)	8.08(-2.21%)	8.46(-21.0%)	49.04(-0.32%)	59.31(-0.20%)	33.35(-0.56%)	34.73(-0.54%)
A-PGNN(**-P)**	13.84(-13.5%)	16.67(-2.28%)	9.67(-6.71%)	10.05(-6.16%)	48.77(-0.87%)	58.75(-1.14%)	33.46(-0.24%)	34.79(-0.37%)
A-PGNN(**-A-P)**	13.15(-8.55%)	16.29(4.51%)	8.63(-16.7%)	9.33(-12.89%)	32.60(-33.7%)	40.06(-32.6%)	23.09(-31.2%)	24.09(-31.0%)
A-PGNN	**14.38**	**17.06**	**10.36**	**10.71**	**49.20**	**59.43**	**33.54**	**34.92**

The numbers in parentheses indicate the percentage of performance degradation of the ablation model compared to A-PGNN.

methods. Table 3 reports the performance comparison results. We have the following observations:

Compared to ”pure” session-based methods SR-GNN and GRU4Rec, the performance of A-PGNN and HierTCN verify that incorporating historical session information can improve the recommendation ability. HierTCN performs better than the H-RNN model on both data sets, indicating that Temporal Convolution Network has a more powerful sequence encoding capability than that of the GRU net. SR-GNN outperforms GRU4Rec and H-RNN by large margins on Xing, which attributes to the superiority of the graph-based model. In particular, prior study[10] on Xing has shown that users' activity within and across sessions has a high degree of repetitiveness, which means their behaviors are more easily to form graph structure. Therefore, the graph-based methods are more effective [6], [43].

It is obvious that deep neural methods perform better than conventional methods in most cases. However, the non-neural model VSKNN and SKNN exhibit strong competitive performance in Xing, which is better than most neural models. A possible explanation is that, in a job-seeking website, users are interested in the same kind of jobs, and their interacted items among sessions are quite similar. So sequence KNN model produced excellent results by retrieving items in the K most similar past sessions in the training data. However, A-PGNN still outperforms VSKNN by 9.28, 6.36, 5.00 percent in Mrr@5/10/20. In contrast, VSKNN and SKNN perform poorly on Reddit than neural models such as A-PGNN, HierTCN, and H-RNN, especially in terms of Mrr values. One possible reason is that the average session length of Reddit is relatively smaller than that of Xing. Regarding entertainment, social, and news websites, user activity within-session has a high degree of variability, making it difficult to extract useful information from similar sessions only.

A-PGNN consistently yields the best performance on all the data sets compared with the state-of-the-art session-aware method HierTCN. We attribute the success of A-PGNN and HierTCN to their ability to model the effect of the user's historical interests on the current session. However, they act explicitly or implicitly. As for HierTCN, it fuses the representations of all historical sessions into a single vector to represent the user's long term interest, which limits the effective use of historical information. In contrast, A-PGNN overcomes this deficiency by utilizing the dot-product attention mechanism to explicitly calculate the impact of historical sessions on the current session, which makes it better to integrate long-term and short-term preferences of users.

4.6 Ablation Study (RQ2)

Next, turn to *RQ2*, we compare our method with different variants to verify the effectiveness of two critical components, the Dot-Product Attention mechanism and PGNN. *A-PGNN(-U)*: APGNN without using user embedding; *A-PGNN(-A)*: A-PGNN without the Dot-Attention mechanism, that is, it does not consider the explicit impact of historical sessions on the current session; *A-PGNN(-P)*: A-PGNN has no PGNN component; *A-PGNN(-A-P)*: A-PGNN neither has the PGNN component nor the Dot-Product attention mechanism, which is equivalent to the session-based method. We show the Recall@5/10, Mrr@5/10 results in Table 4, and have the following findings.

A-PGNN is consistently superior to A-PGNN(-P) and A-PGNN(-A). It illustrates the importance of personalized Graph Neural Network and explicit modeling of historical information. The A-PGNN performs relatively better than A-PGNN(-U), which suggests that PGNN is more capable of capturing relationships between items than vanilla GNN.

A-PGNN(-A-P) outperforms A-PGNN(-A) on Xing. One possible reason is that directly combining historical sessions with the current session to construct user behavior graph may bring noise to the prediction of current session, and does not necessarily lead to improvement. A-PGNN performs better than A-PGNN(-A-P) and A-PGNN(-P), this might be that the PGNN and Dot-Product attention mechanism can mutually reinforce each other: PGNN could be used to capture the complex transition between items, while Dot-Product attention can distinguish important historical session information.

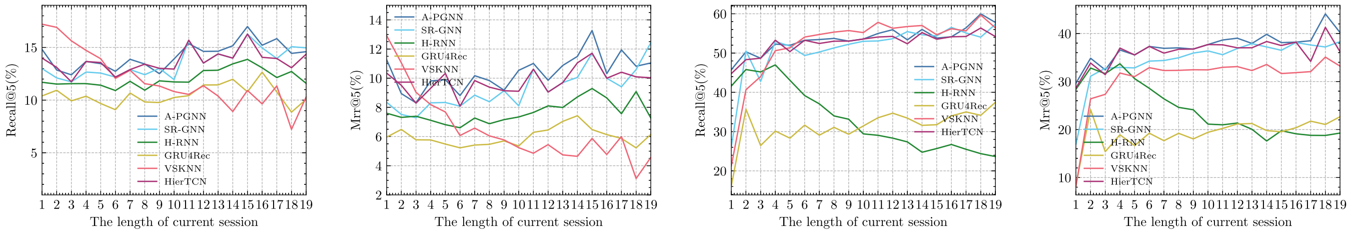
A-PGNN(-A) and A-PGNN(-P) improve by more than 30 percent compared to A-PGNN(-A-P) on Reddit. It again verifies the significance of Dot-Product attention and PGNN but they use different ways to utilize historical session information. Dot-Product attention explicitly calculates the impact of historical sessions on the current session, and PGNN can capture the complex item transition in each user's sessions in user special.

4.7 The Effects of Current Session Length and the Number of Historical Sessions (RQ3)

To answer *RQ3*, we further analyze the effects of current session length and the numbers of historical sessions on performance.

4.7.1 The Effect of Current Session Length

First, we analyze the capability of different models to handle current sessions with different lengths. Similar to H-RNN[10],



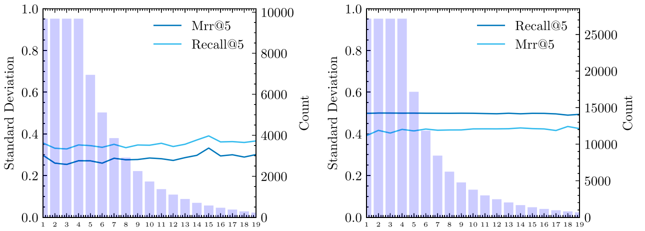
(a) Recall@5 on Xing (b) Mrr@5 on Xing (c) Recall@5 on Reddit (d) Mrr@5 on Reddit

Fig. 3. Performance comparison in terms of Recall@5 and Mrr@5 tested on current sessions with different length on Xing data(a)(b) and Reddit data (c)(d).

we limit the analysis to sessions having length ≥ 5 (8751 sessions for Xing and 26980 for Reddit). The histograms in Fig. 4 show the counts of test cases under different length of current session. For comparison, we evaluate the recommendation performance of A-PGNN with HierTCN, H-RNN, SR-GNN, GRU4Rec, and VSKNN on different length of current sessions, respectively. Fig. 3 shows the Recall@5 and Mrr@5 results. Fig. 4 line charts show the standard deviation (STD) of A-PGNN results under each length of current sessions. Because the maximum length of session is 20, the length of current session ranges from 1 to 19. From these results, some interesting conclusions can be drawn.

On the Xing data set, A-PGNN consistently outperforms other neural models on almost all length sessions. It outperforms other methods in terms of Recall@5 and Mrr@5 when the length is greater than 6 and 3, respectively. In this case, A-PGNN and HierTCN have an advantage over SR-GNN in most length sessions. H-RNN also outperforms GRU4Rec on all sessions. This shows the superiority of personalized models over "pure" session-based models. It is worth noting that the non-neural model VSKNN has achieved surprisingly good results in session length from 1 to 3. This may be because the users' interests are more concentrated in the first few clicks. Therefore, VSKNN can easily produce good results by looking for similar sessions to recommend items. However, the limitations of its model capabilities make it difficult to capture the evolution of interests, so the performance deteriorates as the session length increases.

On the Reddit data set, A-PGNN outperforms SR-GNN in recommendation accuracy by a large margin (up to 43 percent Recall@5 and 43 percent Mrr@5) on sessions with length 1. With the increase of current session length, SR-GNN also becomes more competitive. When it comes to RNN-based models H-RNN and GRU4Rec, the advantage of H-RNN in short sequences is especially apparent. Nevertheless, as the length of the current session increases, we find that its performance in terms of Recall@5 and Mrr@5 becomes worse than GRU4Rec when session length greater than 10 and 12, respectively. A possible explanation is that, for Reddit, user's clicks at the beginning of the current session depend on historical user interactions. As the session becomes longer, user interest drifts gradually. In this case, the evolution of user interest mainly depends on the current session. Overuse of historical information may bring some invalid or interference information, which makes H-RNN hard to deal with sessions with larger length. In comparison, our model maintains stability in this case. Thanks to the attention mechanism that could explicitly model the impact of historical sessions on the current session, which reduces the effects of interference sessions. The mechanism of HierTCN dynamically updating items might alleviate this drawback, but it still underperforms A-PGNN in long sessions. On the Reddit data set, VSKNN performs worse on short sessions than on long sessions. Our analysis believes that the interests of users are diverse in each session, and it needs to take a more extended session to extract their interest. Therefore, VSKNN performance continues to increase and gradually stabilizes as the length of the session increases.



(a) Xing data (b) Reddit data

Fig. 4. The histogram shows the counts of sessions with different length and the line chart shows the standard deviation of indicator tested on sessions with different length.

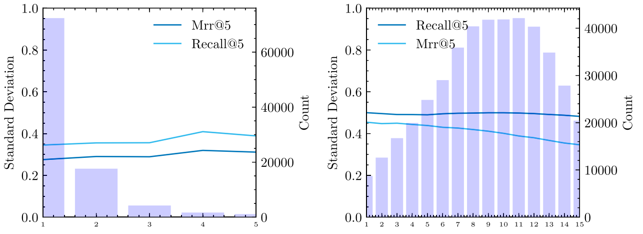
From Fig. 4, we find that for both data sets, the counts of current session with different length obey long-tailed distributions. However, the STD of the performance tested under each length has little variation, which shows that A-PGNN's performance is relatively stable.

4.7.2 The Effect of the Number of Historical Sessions

When making predictions within the current session, is it the more historical sessions we incorporate, the better the recommendation performance will be?

To answer this question, for each test session, we group the test sessions by the number of historical sessions they own, which is denoted as H. For Xing, $H \in [1, 50]$ whereas for Reddit, $H \in [1, 150]$. To facilitate the analysis of indicator change trends, we partition the test sessions into several groups by units of ten. For xing, the testing sessions can be divided into 5 groups $[1, 10), [10, 20), \ldots, [40, 50]$. Also, Reddit can be divided into 15 groups $[1, 10)$, $[10, 20), \ldots,$ $[140, 150]$. Fig. 5 shows the counts of test cases within each group and the STD of indicator tested within each group.

The experiment results are shown in Fig. 6. For Xing, testing sessions in group 4 and 5 generally get a higher performance compared with those in group from 1 to 3. It suggests that those testing sessions with a larger amounts of



(a) Xing data (b) Reddit data

Fig. 5. The histogram shows the counts of test cases within each group and the line chart shows the standard deviation of indicator tested within each group.

historical sessions to assist with prediction generally perform better. For Reddit, the performance rises to a peak at group 3 then continues to fall for the rest of the period. A-PGNN has achieved satisfactory results in group from 1 to 8, while the performances of group from 9 to 15 are even worse than group 1. This shows that as the amount of historical session continues to increase, that is, after approximately greater than 100, the effectiveness of the model begins to deteriorate with the increase of user history length. This disproves the assumption that ”more is better.” From the aspect of reality, Xing is an employment-oriented website, which means that user interest drift is small and action in current session is strongly correlated to historical sessions. In contrast, Reddit is an entertainment, social , and news website, the purpose of the user’s browsing behavior is often unclear, and it is susceptible to drift due to the content posted on the website. So, the interest of a long period in the past may bring noise to the predication of the current test session. In summary, we could choose to retain the appropriate number of historical sessions in actual scenarios according to the characteristic of the data set.

What is more, through Fig. 5, it is found that the number of historical sessions owned by each test session is unevenly distributed, especially Xing, which follows the long-tail distribution. In Xing data set, the STD of the group with a small number of sessions is slightly bigger than that of group with a larger number of session. In Reddit data set, most test cases are mainly concentrated in the 7th to 13th groups. The STD of Recall@5 varies little for cross-groups. From the results of groups 11 to 15, we can see that, as the counts of test cases within each group continue to drop, the STD of Mrr@5 continues to get smaller.

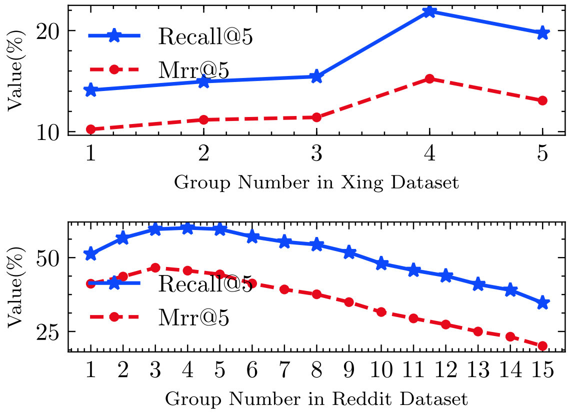
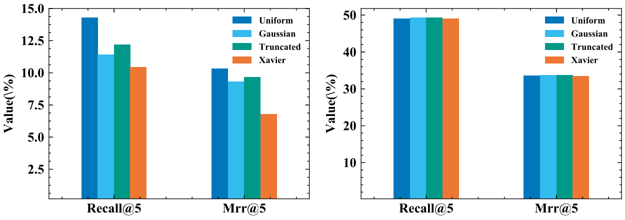


Fig. 6. Performance of A-PGNN in terms of Recall@5 and Mrr@5 with different numbers of historical sessions.



(a) Xing data (b) Reddit data

Fig. 7. Performance of A-PGNN in terms of Recall@5 and MRR@5 with different parameter initialization methods.

4.8 Hyper-Parameter Study (RQ4)

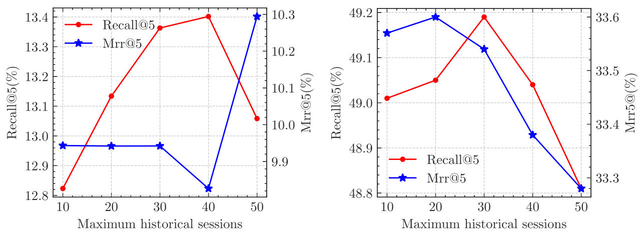
We first conduct a sensitivity analysis of the model parameter initialization. Then, we perform experiments to explore how the hyperparameters like maximum historical sessions and PGNN propagation steps influence the performance.

4.8.1 *Parameter Sensitivity Analysis*

We evaluate the performances of A-PGNN with a variety of choices of initialization mechanisms, including Gaussian $\mathcal{N}(0, 0.1)$, Uniform $\mathcal{U}(-1/\sqrt{d}, 1/\sqrt{d})$, Truncated Gaussian $\mathcal{N}(0, 0.1)$, and Xavier initialization[48]. Fig. 7 shows the experiment results obtained with both data sets. It can be observed that the performance of A-PGNN on Xing with different initialization methods varies greatly and achieves the best results under Uniform initialization, while the results on Reddit data set are more stable. So A-PGNN is more sensitive to parameter initialization on Xing than on Reddit. The reason might be that the small scale of Xing data causes the model to converge quickly, which makes the model more sensitive to the initialization method on this data set, whereas it is more stable on a large scale data set, Reddit.

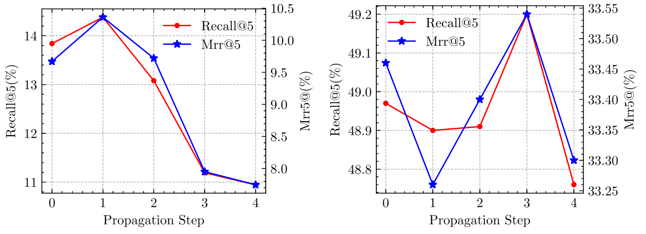
4.8.2 *Effect of Maximum Historical Session*

In this subsection, we investigate how the performance change with hyperparameter maximum historical session M, which indicates the upper limits of historical information that the network can utilize to make predictions for current session. Fig. 8 shows the evaluation values with maximum historical sessions M. Recall@5 reaches its highest level when M is 40 for Xing and M is 3 for Reddit. And it then continues to fall. Although the same thing does not happen on Xing’s performance of Mrr@5, we can make a general conclusion



(a) Xing data (b) Reddit data

Fig. 8. Performance of A-PGNN in terms of Recall@5 and MRR@5 with different maximum historical sessions.



(a) Xing data (b) Reddit data

Fig. 9. Performance of A-PGNN in terms of Recall@5 and Mrr@5 with different propagation steps.

that higher M does not lead to better results. This shows that the increase in user historical information does not necessarily lead to an increase in model performance.

4.8.3 Effect of PGNN Propagation Steps

As the PGNN is a pivotal component of our model, we investigate the effect of propagation step T mentioned in Section 3.4. What is worth mentioning is that the propagation steps mentioned in GGNN[38] is consistent with the Graph Neural Network layers.

The experimental results are shown in Fig. 9. We can see that, for Xing and Reddit, the performance reaches the highest value when T is 1 and 3, respectively, and then gradually deteriorates with the increase of T. The value of the highest point is great than the value of T=0, which also verifies the effectiveness of PGNN. From the perspective of data type, for users in Xing, their interactive items tend to be of similar categories or topics. For example, they tend to look through the same kind of job postings for a specific career, which may suggest that node embeddings in the same user behavior graph are more likely to be closer in the embedding space compared with that of Reddit. The use of far propagation steps in Xing may lead to over-smoothing. In summary, it is more reasonable to choose a smaller T for Xing, and a larger T for Reddit.

5 Conclusion

In this paper, we propose PGNN for session-aware recommendation scenario. A-PGNN captures the complex transition relationships between items in each user behavior graph by the PGNN. At the same time, it uses the Dot-Attention mechanism to explicitly model the effect of historical sessions on the current session, which makes it easy to capture the user's long-term performance. Comprehensive experiments on two public data sets verify the effectiveness of different components in our model and confirm that A-PGNN can outperform other state-of-the-art models in most cases.

For future work, we will improve the flexibility and scalability of PGNN by incorporating the dynamic graph neural networks. Besides, we are also interested in exploring more effective attention mechanisms to integrate the users' long- and short-term interests.

Acknowledgments

This work was supported by National Key Research and Development Program (2018YFB1402605, 2018YFB1402600), National Natural Science Foundation of China (U19B2038, 61772528), Beijing National Natural Science Foundation (4182066).

References

[1] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk, "Session-based recommendations with recurrent neural networks," in *Proc. Int. Conf. Learn. Representations*, 2016.

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

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

[4] 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. Discov. Data Mining*, 2018, pp. 1831–1839.

[5] H. Ying *et al.*, "Sequential recommender system based on hierarchical attention networks," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, 2018, pp. 3926–3932.

[6] 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.*, 2019, vol. 33, pp. 346–353.

[7] B. Twardowski, "Modelling contextual information in session-aware recommender systems with neural networks," in *Proc. 10th ACM Conf. Recommender Syst.*, 2016, pp. 273–276.

[8] C. Wu and M. Yan, "Session-aware information embedding for e-commerce product recommendation," in *Proc. ACM Conf. Inf. Knowl. Manage.*, 2017, pp. 2379–2382.

[9] V. W. Anelli, V. Bellini, T. Di Noia, W. La Bruna, P. Tomeo, and E. Di Sciascio, "An analysis on time-and session-aware diversification in recommender systems," in *Proc. 25th Conf. User Model. Adaptation Personalization*, 2017, pp. 270–274.

[10] M. Quadrana, A. Karatzoglou, B. Hidasi, and P. Cremonesi, "Personalizing session-based recommendations with hierarchical recurrent neural networks," in *Proc. 11th ACM Conf. Recommender Syst.*, 2017, pp. 130–137.

[11] M. Ruocco, O. S. L. Skrede, and H. Langseth, "Inter-session modeling for session-based recommendation," in *Proc. 2nd Workshop Deep Learn. Recommender Syst.*, 2017, pp. 24–31.

[12] J. You, Y. Wang, A. Pal, P. Eksombatchai, C. Rosenburg, and J. Leskovec, "Hierarchical temporal convolutional networks for dynamic recommender systems," in *Proc. World Wide Web Conf.*, 2019, pp. 2236–2246.

[13] A. Vaswani *et al.*, "Attention is all you need," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.

[14] A. Mnih and R. Salakhutdinov, "Probabilistic matrix factorization," in *Proc. 20th Int. Conf. Neural Inf. Process. Syst.*, 2007, pp. 1257–1264.

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

[16] Y. Koren and R. Bell, "Advances in collaborative filtering," in *Recommender Systems Handbook*. Berlin, Germany: Springer, 2011, pp. 145–186.

[17] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proc. 10th Int. Conf. World Wide Web*, 2001, pp. 285–295.

[18] G. Shani, R. I. Brafman, and D. Heckerman, "An MDP-based recommender system," in *Proc. 18th Conf. Uncertainty Artif. Intell.*, 2002, pp. 453–460.

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

[20] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. 26th Annu. Conf. Neural Inf. Process. Syst.*, 2013, pp. 3111–3119.

[21] T. Mikolov, M. Karafiát, L. Burget, J. Cernockỳ, and S. Khudanpur, "Recurrent neural network based language model," in *Proc. 11th Annu. Conf. Int. Speech Commun. Assoc.*, 2010, vol. 2, Art. no. 3.

[22] K. Cho *et al.*, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *Proc. Conf. Empir. Methods Natural Lang. Process.*, 2014, pp. 1724–1734.

[23] I. V. Serban, A. Sordoni, Y. Bengio, A. Courville, and J. Pineau, "Building end-to-end dialogue systems using generative hierarchical neural network models," in *Proc. 30th AAAI Conf. Artif. Intell.*, 2016, pp. 3776–3784.

[24] 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.*, 2016, pp. 241–248.

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

[26] T. X. Tuan and T. M. Phuong, "3D convolutional networks for session-based recommendation with content features," in *Proc. 11th ACM Conf. Recommender Syst.*, 2017, pp. 138–146.

[27] J. Li, P. Ren, Z. Chen, Z. Ren, T. Lian, and J. Ma, "Neural attentive session-based recommendation," in *Proc. ACM Conf. Inf. Knowl. Manage.*, 2017, pp. 1419–1428.

[28] C. Wu and M. Yan, "Session-aware information embedding for e-commerce product recommendation," in *Proc. ACM Conf. Inf. Knowl. Manage.*, 2017, pp. 2379–2382.

[29] T. M. Phuong, T. C. Thanh, and N. X. Bach, "Neural session-aware recommendation," *IEEE Access*, vol. 7, pp. 86 884–86 896, 2019.

[30] T. Liang, Y. Li, R. Li, X. Gu, and Y. Hu, "Personalizing session-based recommendation with dual attentive neural network," in *Proc. Int. Joint Conf. Neural Netw.*, 2019, pp. 1–8.

[31] B. Perozzi, R. Al-Rfou, and S. Skiena, "DeepWalk: Online learning of social representations," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2014, pp. 701–710.

[32] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei, "Line: Large-scale information network embedding," in *Proc. 24th Int. Conf. World Wide Web*, 2015, pp. 1067–1077.

[33] A. Grover and J. Leskovec, "Node2vec: Scalable feature learning for networks," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2016, pp. 855–864.

[34] D. Duvenaud *et al.*, "Convolutional networks on graphs for learning molecular fingerprints," in *Proc. 28th Int. Conf. Neural Inf. Process. Syst.*, 2015, pp. 2224–2232.

[35] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. Int. Conf. Learn. Representations*, 2016.

[36] M. Gori, G. Monfardini, and F. Scarselli, "A new model for learning in graph domains," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, 2005, vol. 2, pp. 729–734.

[37] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE Trans. Neural Netw.*, vol. 20, no. 1, pp. 61–80, Jan. 2009.

[38] Y. Li, D. Tarlow, M. Brockschmidt, and R. S. Zemel, "Gated graph sequence neural networks," in *Proc. Int. Conf. Learn. Representations*, 2015.

[39] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *Proc. 6th Int. Conf. Learn. Representations*, 2018.

[40] Z. Li, X. Ding, and T. Liu, "Constructing narrative event evolutionary graph for script event prediction," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, 2018, pp. 4201–4207.

[41] R. Li, M. Tapaswi, R. Liao, J. Jia, R. Urtasun, and S. Fidler, "Situation recognition with graph neural networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 4183–4192.

[42] K. Marino, R. Salakhutdinov, and A. Gupta, "The more you know: Using knowledge graphs for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 20–28.

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

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

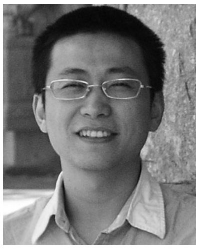
[45] M. Ludewig and D. Jannach, "Evaluation of session-based recommendation algorithms," *User Model. User-Adapted Interaction*, vol. 28, no. 4/5, pp. 331–390, 2018.

[46] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Representations*, 2015.

[47] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 448–456.

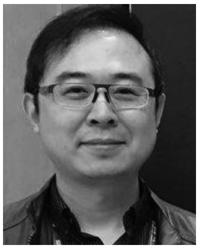
[48] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. 13th Int. Conf. Artif. Intell. Statist.*, 2010, pp. 249–256.

Mengqi Zhang received the BS degree from Xiangtan University, China, in 2016, the MS degree from Beihang University, China, in 2019. He is currently pursuing the PhD degree in computer application technology with the University of Chinese Academy of Sciences, Beijing, China. His research interests include data mining, machine learning, and recommender systems.



Shu Wu is an associate professor in the Center for Research on Intelligent Perception and Computing (CRIPAC). He has published more than 50 papers in the areas of data mining and information retrieval at international journals and conferences, such as *IEEE Transactions on Knowledge and Data Engineering*, WWW, AAAI, SIGIR, and ICDM.

Meng Gao is currently working toward the undergraduate degree from the School of Computer and Communication Engineering, University of Science and Technology Beijing, China. She is working as an intern in the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, China.



Xin Jiang is currently an associate professor in LMIB & Beijing Advanced Innovation Center for Big Data and Brain Computing, Beihang University, China. He has published more than 40 papers in the areas of data science and complexity at international journals and conferences, such *Physical Review Letters*, EPL, *Physical Review E*, and *New Journal of Physics*.

Ke Xu received the BE, ME, and PhD degrees from Beihang University, China, in 1993, 1996, and 2000, respectively. He is a professor of computer science at Beihang University, China. His research interests include algorithms and complexity, data mining, and networks.

Liang Wang (Fellow, IEEE) received both the BEng and MEng degrees from Anhui University, China, in 1997 and 2000, respectively, and the PhD degree from the Institute of Automation, Chinese Academy of Sciences (CASIA), China, in 2004. Currently, he is a full professor of Hundred Talents Program at the National Lab of Pattern Recognition, CASIA. His major research interests include machine learning, pattern recognition, and computer vision. He has widely published in highly ranked international journals such as *IEEE Transactions on Pattern Analysis and Machine Intelligence* and *IEEE Transactions on Image Processing* and leading international conferences such as CVPR, ICCV, and ICDM. He is an IAPR fellow.