



Spatio-Temporal Representation Learning with Social Tie for Personalized POI Recommendation

Shaojie Dai¹ · Yanwei Yu¹ · Hao Fan¹ · Junyu Dong¹

Received: 29 May 2021 / Revised: 14 September 2021 / Accepted: 16 January 2022 / Published online: 31 January 2022
© The Author(s) 2022

Abstract

Recommending a limited number of Point-of-Interests (POIs) a user will visit next has become increasingly important to both users and POI holders for Location-Based Social Networks (LBSNs). However, POI recommendation is a challenging task since complex sequential patterns and rich contexts are contained in extremely sparse user check-in data. Recent studies show that embedding techniques effectively incorporate POI contextual information to alleviate the data sparsity issue, and Recurrent Neural Network (RNN) has been successfully employed for sequential prediction. Nevertheless, existing POI recommendation approaches are still limited in capturing user personalized preference due to separate embedding learning or network modeling. To this end, we propose a novel unified spatio-temporal neural network framework, named PPR, which leverages users' check-in records and social ties to recommend personalized POIs for querying users by joint embedding and sequential modeling. Specifically, PPR first learns user and POI representations by joint modeling User-POI relation, sequential patterns, geographical influence, and social ties in a heterogeneous graph and then models user personalized sequential patterns using the designed spatio-temporal neural network based on LSTM model for the personalized POI recommendation. Furthermore, we extend PPR to an end-to-end recommendation model by jointly learning node representations and modeling user personalized sequential preference. Extensive experiments on three real-world datasets demonstrate that our model significantly outperforms state-of-the-art baselines for successive POI recommendation in terms of Accuracy, Precision, Recall and NDCG. The source code is available at: <https://www.anonymous.4open.science/r/DSE-1BEC>.

Keywords POI recommendation · Location-based social networks · Spatio-temporal neural network · Heterogeneous graph

1 Introduction

Newly emerging LBSNs has become an important mean for people to share their experience, write comments, or even interact with friends. With the prosperity of LBSNs, many users check-in at various POIs via mobile devices in real time. Therefore, a large amount of check-in data is being generated, which is crucial to understand the users'

preferences and behaviors. Recently, [9] leverages a large-scale LBSN simulator to simulate human behaviors and generate synthetic, dense and large-scale LBSN data based on human patterns of life, which provides a good opportunity to understand people's behavior patterns. POI recommendation not only helps users explore attractive and interesting places, but also gives guidance to location-based service providers, where to launch advertisements to target customers for marketing. Due to the great significance to both users and businesses, how to use spatio-temporal information effectively and to recommend a limited number of POIs users more likely visit next have been attracting increasing attention in both industry and academia.

In particular, several studies [3, 15, 19, 26, 30] have been conducted to recommend successive POIs for users based on users' spatio-temporal check-in sequence in LBSNs. Based on Markov chain model, LORE [30] and NLPMM [3] explore users' successive check-in patterns by considering temporal and spatial information. ST-RNN [15] employs

✉ Yanwei Yu
yuyanwei@ouc.edu.cn

Shaojie Dai
daishaojie@stu.ouc.edu.cn

Hao Fan
fanhao@ouc.edu.cn

Junyu Dong
dongjunyu@ouc.edu.cn

¹ Department of Computer Science and Technology, Ocean University of China, Qingdao 266100, Shandong, China

RNN to capture the users' sequential check-in behaviors. In a follow-up work, STGN [34] carefully designs the time gates and distance gates in LSTM to model users' sequential visiting behaviors by enhancing long short-term memory. Additionally, some models [1, 7] based on Word2Vec [20] framework to capture the preference and mobility pattern of users and the relationship among POIs also achieved decent performance. GE [26] uses graph embedding to combine the sequential effect, geographical influence, temporal effect and semantic effect in a unified way for location-based recommendation. Recently, SAE-NAD [19] utilizes a self-attentive encoder to differentiate the user preference and a neighbor-aware decoder to incorporate the geographical context information for POI recommendation.

However, location-based POI recommendation still faces three major challenges. First, *data sparsity*, unlike the general e-commerce, music and movie recommendation, which can be collected and verified just online, location-based POI recommendation systems usually associate with the POI-entities. Only when a user visits a POI-entity, a check-in record is generated. Therefore, the check-in records in the POI recommendation task is much sparser. This issue has plagued many POI recommendation models based on the collaborative filtering. Furthermore, data sparsity problem in check-in data makes it difficult to capture user's sequential pattern, because the check-in sequence is very short or is not continuous in time. Second, *contextual factors*, POI recommendation may be affected by various contextual factors, including social tie influence, geographical influence, temporal context, and so on. In fact, social ties are often available in LBSNs, and recently studies show that social networks associated with users are important in POI recommendation task since users are more likely to be influenced by their close friends (Who keeps company with the wolf will learn to howl). In this work, we incorporate social ties, check-in time interval, sequential and geographical effect into user-POI interaction graph to joint learn user and POI representations. Lastly, *dynamic and personalized preferences*, users' preferences are changing dynamically over time. At different time and circumstances, users may prefer different POIs. For example, some users prefer to visit gourmet restaurants in the local area, but when they go to a new city, some prefer to visit the cultural landscapes, while some prefer the natural landscapes. Dynamically and accurately capturing this trend has been proved to be essential for personalized POI recommendation task. However, effectively modeling the personalized sequential transitions from the sparse check-in data is challenging.

To address the aforementioned challenges, in this work, we stand on advances in embedding technique and RNN network, and propose our model, named *PPR*, which is a *spatial-temporal representation learning* framework for personalized and successive POI recommendation. First,

we jointly model the user-POI relation, sequential effect, geographical influence and social ties by constructing a heterogeneous graph and then develop a densifying trick by adding second-order neighbors to nodes with low in/out-degrees to alleviate the data sparsity issue. Then, we learn user and POI representations by embedding the densified heterogeneous graph into a shared low-dimensional space. Furthermore, to better capture the user dynamic and personalized preference, we also design a spatio-temporal neural network by concatenating user embedding, POI embedding and POI category as personalized sequence input to feed the network.

This work is extended from a conference paper [6]. The differences between this work and the conference paper are summarized as follows: First, we extend PPR [6] to an end-to-end POI recommendation model, named GCN-LSTM. GCN-LSTM jointly learns user and POI representations using Graph Convolutional Network (GCN) in the constructed heterogeneous graph and captures the user personalized sequential preference using the spatio-temporal neural network. Second, we conduct additional experiments to evaluate our extended model. Experimental results demonstrate that GCN-LSTM outperforms all baselines and also performs better than our previous PPR. Third, we add an overview framework for our model, which illustrates the relationships between the key components of our model. Additionally, we design a new variant *PPR-Soc* to verify the social ties influence on model performance, and the result is shown in Fig. 3. Lastly, we verify the effect of hyperparameters ρ and α on the performance of our model, and results are depicted in Table 5 and Fig. 6, respectively.

We conduct extensive experiments on three real-world datasets to verify the effectiveness of our proposed models. Experimental results demonstrate our models significantly improve the performance on successive POI recommendation task compared to state-of-the-art baseline methods. The main contributions of this paper are summarized as follows:

- We propose a novel PPR model for personalized POI recommendation, which incorporates users' check-in records and social ties. We construct a heterogeneous graph by jointly taking user-POI relation, sequential pattern, geographical effect and social ties into consideration to learn the representations of users and POIs.
- We propose a spatio-temporal neural network to model users' dynamic and personalized preference by concatenating user, POI embedding and POI category to generate personalized behavior sequence.
- we propose an end-to-end POI recommendation models to jointly learn user and POI representations and model users' dynamic and personalized preference.
- We conduct extensive experiments to compare our method with state-of-the-art baselines, and our method

significantly outperforms state-of-the-art baselines for successive POI recommendation task.

2 Related Work

General POI Recommendation The most well-known approaches of personalized recommendation are collaborative filtering (CF) and Matrix Factorization (MF). The conventional CF techniques have been widely studied for POI recommendation. LARS [11] employs item-based CF to make POI recommendation with the consideration of travel penalty. FCF [28] is a friend-based CF model based on the common visited POIs among friends, which considers the social influence. UTE [29] is a collaborative recommendation model that incorporates with temporal and geographical information. However, such methods suffer the data sparsity problem, leading them difficult to identify similar users.

Recommendation models based on MF and embedding learning [12, 16, 17] have been intensively studied. Rank-GeoFM [13] fits the users' preference rankings for POIs to learn the latent embeddings. By incorporating the geographical context, it utilizes a geographical factorization method for calculating the recommendation score. TSG-MF [32] models the multi-tag influences via extracting a user-tag matrix and the social influences via social regularization, and uses a normalized function to model geographical influences.

Next POI Recommendation In the literature, next POI recommendation issues have been studied in [25, 34], in which the main objective is to exploit the user's check-in sequence between different POIs and dynamic preference.

Markov Chains (MC)-based Methods MC-based models aim to predict the next behavior according the historical sequential behaviors. FPMC-LR [4] considers first-order Markov chain for POI transitions and distance constraints. HMM [27] exploits check-in category information to capture the latent user movement pattern by using a mixed hidden Markov chain. LORE [30] incrementally mines sequential patterns and represents it as a dynamic location-location transition graph. By utilize an additive Markov chain, LORE fuses the sequential, geographical and social influence in a unified way.

Graph-based Methods Graph-based approaches are exploring in the literature of next POI recommendation. GE [26] jointly captures the latent relations among the POI, region, time slot and words related to the POIs by constructing four bipartite graphs. HME [8] projects the entities into a hyperbolic space after study multiple contextual subgraphs. Although the above approaches achieve promising performance, they cannot model the sequential patterns effectively.

RNN-based Methods Recently, RNNs such as LSTM or GRU have demonstrated groundbreaking performance

on predicting sequential problem. ST-RNN [15] utilizes RNN structure to model the temporal contexts by carefully designing the time-specific and distance-specific transition matrices. NEXT [31] encodes the sequential relations within the pre-trained POI embeddings by adopting DeepWalk [21] technique. Time-LSTM [37] employs LSTM with time gates to capture time interval among users' behaviors. CAPE [1] first uses a check-in context layer to capture the geographical influence of POIs and a text content layer to model the characteristics of POIs from text content. Then, CAPE employs RNN as recommendation component to predict successive POIs. PEU-RNN [18] proposes a LSTM-based model that combines the user and POI embeddings, which are learned from Word2Vec. ASPPA [33] proposes to identify the semantic subsequences of POIs and discover their sequential patterns. Recently, STGN [34] extends the LSTM gating mechanism with the spatial and temporal gates to capture the user's space and time preference. However, these approaches fail to capture users' personalized preferences. In addition, PLSPL [25] adopts the attention mechanism to model the long-term preference and employs two LSTM models to model the short-term preference on location-based and category-based sequence, respectively. Nevertheless, PLSPL does not utilize the geographical information and social ties, which play important roles in POI recommendation task [5, 6, 32].

3 Problem Definition

In this section, we first give the key concepts used in this paper. Then, the problem definition for personalized POI recommendation is formulated.

Definition 1 (POI) A POI is a uniquely identified venue in the form of $\langle p, \ell, cat \rangle$, where p is the POI identifier, cat denotes the category of the POI, and ℓ represents the geographical coordinates of the POI (*i.e.*, longitude and latitude).

Definition 2 (Check-in record) A check-in record is a triple $c = \langle u, v, t \rangle$ that represents user u visiting POI v at timestamp t .

The collection of all users is denoted as U , and the collection of all POIs is denoted as V .

Definition 3 (Trajectory) The trajectory of a user u is a sequence of all check-in records $(\langle u, v_1, t_1 \rangle, \langle u, v_2, t_2 \rangle, \dots, \langle u, v_n, t_n \rangle)$ made by user u in chronological order. We denote it as T_u .

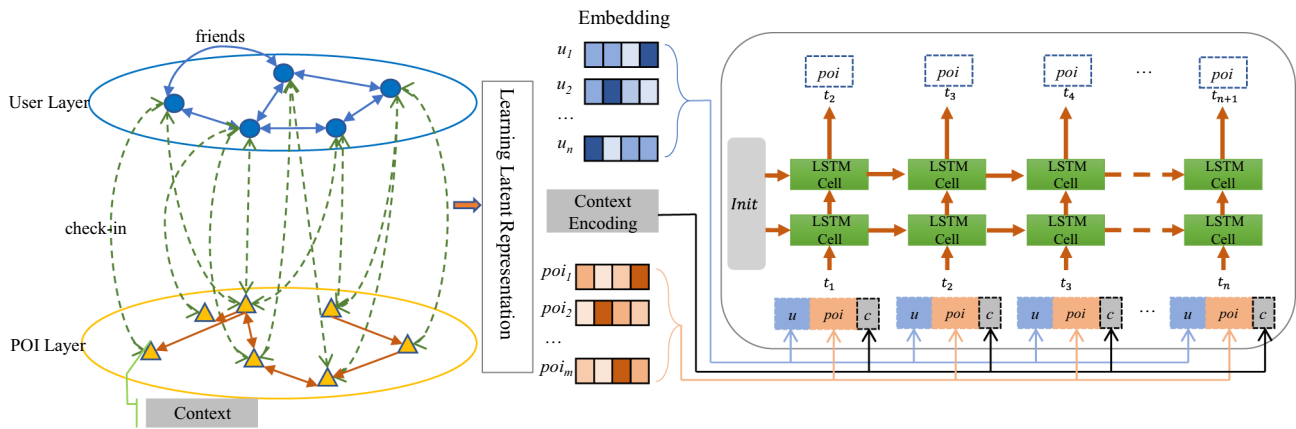


Fig. 1 The overview of the proposed model

Definition 4 (Social Ties) Social ties among users is defined as a graph $\mathcal{G}_u = (U, \mathcal{E}_u)$, where U is the set of users, and \mathcal{E}_u is the set of edges between the users. Each edge $e_{ij} \in \mathcal{E}_u$ represents users u_i and u_j being friends in LBSNs and is associated with a weight $w_{ij} > 0$, which indicates their tie strength.

Problem (Successive POI Recommendation) Given users' check-in records and their social ties, and a querying user u with his/her current check-in $\langle u, v, t \rangle$, our goal is to recommend top- k POIs that user u would be interested in in the next τ time period.

4 Methodology

In this section, we first present the details of the proposed framework PPR. Then, we introduce our model how to utilize PPR model to make personalized POI recommendation. Finally, we describe how to extend PPR to an end-to-end recommendation model.

Figure 1 illustrates the overview of the proposed model, consisting of three key components: (1) *heterogeneous graph construction*, (2) *learning latent representation*, and (3) *modeling user personalized preference*.

4.1 Heterogeneous Graph Construction

We first introduce the heterogeneous User-POI graph to model users' sequential check-ins and social relationships. Specifically, we employ a heterogeneous graph $\mathcal{G} = (V, U, \mathcal{E}, W)$ to jointly model the multiple relations between users and POIs. U and V are the user collection and POI collection, respectively, and \mathcal{E} is the set of all edges between nodes in \mathcal{G} , which are categorized into three edge types, *i.e.*, \mathcal{E}_u , \mathcal{E}_v , and $\mathcal{E}_{u,v}$. As mentioned in Definition 4,

each edge $e_{i,j} \in \mathcal{E}_u$ represents that user u_i and u_j are friends. Each edge $e_{i,j} \in \mathcal{E}_v$ denotes that there exists at least one user visits POI v_j after visiting POI v_i , and each edge $e_{i,j} \in \mathcal{E}_{u,v}$ indicates that user u_i visits POI v_j at least one time. Notice that each edge $e \in \mathcal{E}_u \cup \mathcal{E}_{u,v}$ is a bi-directed edge and each edge $e \in \mathcal{E}_v$ is a directed edge, and each edge is associated with a weight $w \in W (w > 0)$, which indicates the strength of the relation.

4.1.1 Modeling User-POI Relation

Intuitively, we consider that if user u_i visit POI v_j more frequent, u_i and v_j have a stronger relation than with other POIs. Therefore, we formulate the weight between user u_i and POI v_j as:

$$w_{i,j} = freq(u_i, v_j), \tag{1}$$

where $freq(\cdot)$ denotes check-in frequency of user u_i visiting POI v_j . Since we aim to build a directed graph to accommodate the following work, we define $w_{i,j} = w_{j,i}$ for the bi-directed edge $e_{i,j} \in \mathcal{E}_{u,v}$ between user u_i and POI v_j .

4.1.2 Modeling Sequential and Geographical Effect

Compared with general POI recommendation, successive POI recommendation pays more attention to sequential pattern. The impact of user's recent check-in behaviors are greater than those of a long time ago when making POI recommendations [26]. To further model the sequential effect, we carefully design a weighting strategy for the edges in \mathcal{E}_v .

Let $\Delta t_{k,k+1}^u$ be the time interval between two consecutive check-in records in the trajectory T_u of user u . $I_{k,k+1}^u$ is the flag that indicates the status of a pair of consecutive check-in records in the trajectory T_u , which is defined as:

$$l_{k,k+1}^u = \begin{cases} 1 & \text{if } \Delta t_{k,k+1}^u < \theta \\ 0 & \text{else} \end{cases}, \quad (2)$$

where θ is a predefined time threshold.

Given an edge $e_{i,j} \in \mathcal{E}_v$ from POI v_i to POI v_j , the sequential weight $w_{i,j}^{(seq)}$ for the edge $e_{i,j}$ is defined as:

$$w_{i,j}^{(seq)} = \sum_{u \in U} \sum_{k=1}^{|T_{u_i}|-1} l_{k,k+1}^u, \quad \text{if } v_k = v_i \text{ and } v_{k+1} = v_j. \quad (3)$$

Namely, the weight $w_{i,j}^{(seq)}$ for the edge from POI v_i to POI v_j is the total number of times that all users visit v_i first and then v_j in their trajectories.

Furthermore, geographical influence indicates the impact of geographical distance to the users' spatial behaviors. According to [8, 14], the distribution of the geographical distance between two successive POIs follows the power-law distribution, which means users are more willing to visit POIs close to the current location. Therefore, we incorporate the geographical distance into our model as follows:

$$w_{i,j}^{(geo)} = \frac{d_{i,j}^\kappa}{\sum_{v_k \in N(v_i)} d_{i,k}^\kappa}, \quad (4)$$

where $N(v_i)$ represents the set of out-neighbor POIs of POI v_i in \mathcal{E}_v , $d_{i,j}$ denotes the Euclidean distance between POIs v_i and v_j , and κ is the negative exponent (*i.e.*, $\kappa < 0$). Finally, we combine the sequential and geographical influence as follows:

$$w_{i,j} = w_{i,j}^{(seq)} \cdot w_{i,j}^{(geo)}. \quad (5)$$

In such way, the sequential, time interval and geographical information are all reflected in graph \mathcal{G} .

4.1.3 Modeling Social Tie Strength

Users in an LBSN have multiple types of relations with other users, such as friends, family and colleagues. The preference of a user in social network are easily affected by his/her close friends or other users which has some kind of relations with them. Recently, these social ties are incorporated into the POI recommendation system [31] to improve the recommendation performance. In this work, we propose to assign the weight between the users based on their historical check-in interactions. Specifically, for two socially connected users u_i and u_j , we assign the edge weight $w_{i,j}$ as:

$$w_{i,j} = \frac{\varepsilon + \sum_{v \in V} \min(f_{u_i,v}, f_{u_j,v})}{|T_{u_i} \cap T_{u_j}| + 1}, \quad (6)$$

where ε is a very small float number to avoid two users have connection but no common visited POIs, $f_{u_i,v}$ denotes the

frequency of user u_i visiting at POI v , and $|T_{u_i} \cap T_{u_j}|$ represents the number of the common visited POIs for user u_i and u_j . Therefore, the common preferences between socially connected users are also taken into account in the User-POI graph \mathcal{G} .

4.1.4 Densifying Graph

Most recommendation models need to take the data sparsity into consideration, but the check-in data in POI recommendation area is much sparser. To address the data sparsity issue, we propose to construct a dense graph based on the graph \mathcal{G} . Specifically, we regard each user and POI as a node and expand the neighbors of those nodes with low in/out degrees by adding higher order neighbors. In this work, we only consider expanding second-order neighbors to every node. If the out-degree of a node in \mathcal{G} is less than a predefined threshold ρ , we create an edge from node v_i to its second-order out-neighbor node v_j and assign the weight as follows:

$$w_{i,j} = \sum_{v_k \in N(v_i)} w_{i,k} \frac{w_{k,j}}{d_k^{(o)}}, \quad (7)$$

where $N(v_i)$ is the set of out-neighbors of node v_i , and $d_k^{(o)}$ is the out-degree of the node v_k . The densifying method for nodes with a low in-degree less than ρ is same. After densifying the User-POI graph, we can get the a more dense network, denoted by \mathcal{G}_{dense} . Then, we use \mathcal{G}_{dense} instead of \mathcal{G} and exploit embedding technique to learn the nodes' representation vectors.

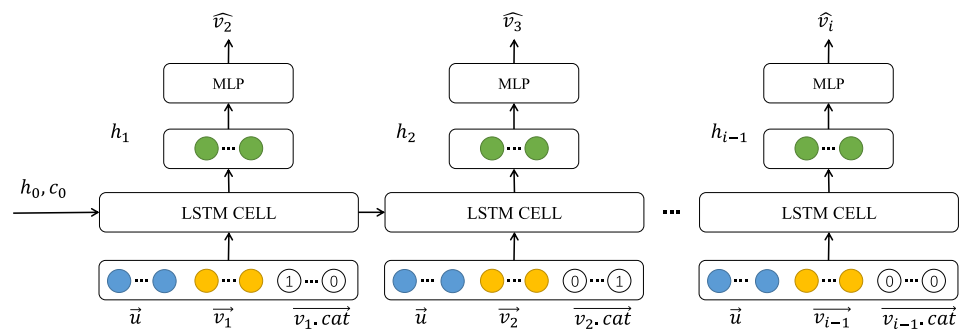
4.2 Learning Latent Representation

Inspired by LINE [22], which learns the first- and second-order relations representations of homogeneous networks. We develop it to learn heterogeneous node representations on our constructed heterogeneous graph \mathcal{G}_{dense} .

Specifically, we regard each user or POI as a node v and ignore their node type. In graph \mathcal{G}_{dense} , each node plays two roles: the node itself and a specific "context" of other nodes. We use \vec{v}_i to denote the embedding vector of node v_i when it is treated as a node, and \vec{v}_i' to denote the embedding vector of v_i when it is treated as a specific "context". In particular, we use a binary cross-entropy loss to encourage nodes and their "context" connected with an edge, to have similar embeddings. Therefore, we minimize the following objective function:

$$\mathcal{O} = - \sum_{e_{i,j} \in \mathcal{E}} \left(\log (\sigma(\vec{v}_j'^T \cdot \vec{v}_i)) \right) + w_n \sum_{v_n \in Neg(v_i)} \log (1 - \sigma(\vec{v}_n'^T \cdot \vec{v}_i)), \quad (8)$$

Fig. 2 Modeling user dynamic and personalized preference



where $\sigma()$ is the sigmoid function, \vec{v}_j^{rT} denotes vector transpose, $Neg(v_i)$ is a negative edge sampling *w.r.t.* node v_i in \mathcal{G}_{dense} , and w_n denotes the negative sampling ratio, which is a tunable hyperparameter to balance the positive and negative samples.

By minimizing the objective function \mathcal{O} with ASGD (asynchronous stochastic gradient) optimization and edge sampling technique, we can learn a d -dimensional embedding vector for each user and POI in \mathcal{G}_{dense} . Additionally, the representation learning is highly efficient and is able to scale to very large graphs because of the use of edge sampling technique.

4.3 Modeling User Dynamic and Personalized Preference

After representation learning, all users and POIs are mapped into a low dimensional space. However, the latent representations only capture the users' preferences or POIs' characteristics in a general way. Although it can model sequence transition patterns and geographical influence, some personalized preference may not be preserved in the node representations.

Furthermore, the categories of POIs are very useful to make a better representation of venues and improve the recommendation performance. In order to model user dynamic and personalized preference, we propose to concatenate user embedding, POI embedding and POI category to generate a new and more personalized embedding to represent a check-in record. More concretely, we use one-hot encoding to represent the POI category information.

Additionally, to better model user dynamic preference and sequential behavior patterns, we utilize LSTM model to construct a spatio-temporal neural network.

As illustrated in Fig. 2, h_t and c_t denote the hidden state and cell state of LSTM at time t , respectively. Given a user u and his/her trajectory sequence T_u , first, we concatenate the user embedding, POI embeddings with POI categories that he/she visited, and we can get a new embedding sequence. Second, we feed LSTM network with these new embedding sequences of all users. Specifically, we utilize the first $i - 1$

POIs as input to train the network and predict the $(i + 1)$ th POI as the recommended POI based on the current i th POI. At the output layer, we also connect a multi-layer perceptron (MLP). Therefore, we use the following objective function to train the model:

$$\mathcal{O}_{seq} = \sum_{u \in U} \sum_{t=1}^{i-1} MSE(MLP(h_t), \vec{v}_{t+1}), \tag{9}$$

where h_t is hidden representation at time step t , $MSE(\cdot, \cdot)$ is a criterion that measures the mean squared error (*e.g.*, squared L2 norm) between each element.

4.4 Personalized POI Recommendation

As described in Sect. 4.3, the user embedding and the first i POI embedding sequence are used to train the spatio-temporal neural network. For the querying user u , the embedding vector of the $(i + 1)$ th POI can be predicted by the current POI v_i as:

$$\widehat{v}_{i+1} = MLP(h_i). \tag{10}$$

Therefore, for each POI v , we calculate its recommendation score as follows:

$$Score(v|\widehat{v}_{i+1}, u, T_u) = 1 - MSE(\widehat{v}_{i+1}, \vec{v}). \tag{11}$$

Finally, we rank all POIs by their recommendation scores and select top- k POIs as the candidate that user u is more likely to visit in the next τ time period.

4.5 End-to-End GCN-LSTM Recommendation Model

Recently, GCNs have been widely used in network embedding to capture graph structure by aggregating neighbor node features and have achieved great success [2, 24, 36]. In this paper, the User-POI heterogeneous network is a weighted graph in nature, and the features of each user or POI can be regarded as the signals on the graph. Therefore, in order to make full use of topological properties of the heterogeneous User-POI graph, we perform graph convolutions

based on the spectral graph theory [10] to directly process the features.

In spectral graph analysis, User-POI graph can be represented by its corresponding Laplacian matrix. The properties of the User-POI graph structure can be obtained by analyzing Laplacian matrix and its eigenvalues. Laplacian matrix of a graph is defined as $\mathbf{L} = \mathbf{D} - \mathbf{A}$, and we adopt its normalized form $\mathbf{L} = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}} \in \mathbb{R}^{N \times N}$, where \mathbf{A} is the adjacent matrix, \mathbf{I} is the identity matrix, and the degree matrix \mathbf{D} is diagonal matrix.

Following [10], multi-layer GCN performs the following layer-wise propagation rule:

$$\mathbf{H}^{(l+1)} = \text{Relu}(\tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}\mathbf{H}^{(l)}\mathbf{W}^{(l)}), \quad (12)$$

where $\mathbf{W}^{(l)}$ is a specific layer trainable weight matrix, $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ and $\tilde{\mathbf{D}}_{ii} = \sum_j \tilde{\mathbf{A}}_{ij}$. $\mathbf{H}^{(0)} = \mathbf{X} \in \mathbb{R}^{N \times D_0}$, where \mathbf{X} denotes the feature matrix and D_0 is the number of features. $\mathbf{H}^{(l)}$ is the output of the l th layer.

4.5.1 Graph Convolution on User-POI Graph

As described in Sect. 4.1, our User-POI graph is a weighted, directed and heterogeneous graph, *i.e.*, $\mathcal{E}_{v,u}$ are directional. To accommodate to GCN, we first symmetrize the adjacency matrix \mathbf{A} as follows:

$$\mathbf{A}_{i,j} = \max(\mathbf{A}_{i,j}, \mathbf{A}_{j,i}). \quad (13)$$

Then, we utilize the layer-wise propagation rule in Eq. (12) to model the user-POI relation, sequential pattern, geographical effect and the common preferences between socially connected users. For POIs, we adopt the contextual factors as their features, *e.g.*, the POI category and textual comments. For users, we aggregate the POIs' feature where the user visited. Afterward, the max-min normalization operation is performed. We take the symmetrical adjacency weight matrix \mathbf{A} and the feature matrix \mathbf{X} as the input of graph convolution network. The forward-propagation output of graph convolution network is the embeddings \mathbf{H} of all nodes.

4.5.2 Jointly Learning with LSTM

Different from Sect. 4.2, which learns the latent representation and user dynamic and personalized preference separately, in this section, we formally define the jointly learning objective function to obtain the recommendation result of global optimization. Specifically, we adopt an unsupervised objective function \mathcal{O}_{gcn} to maximize the similarity of the node representations appearing in the same random walks:

$$\mathcal{O}_{gcn} = \sum_{v_i \in V} \left(\sum_{e_{ij} \in Walks} -\log(\sigma(s_{i,j})) - \sum_{v_n \in Neg(v_i)} \log(1 - \sigma(s_{i,n})) \right), \quad (14)$$

where $s_{i,j}$ denotes the similarity score (*e.g.*, inner product operation) between the representation vectors of node v_i and node v_j , $Walks$ is the set of random walks sampled in User-POI graph, and $Neg(v_i)$ is a negative sampling *w.r.t.* node v_i in User-POI graph.

Finally, we combine \mathcal{O}_{gcn} and \mathcal{O}_{seq} via a hyperparameter α (we can tune α automatically by optuna,¹ a Bayesian hyperparameter optimization tools), which is used to balance the importance of \mathcal{O}_{gcn} and \mathcal{O}_{seq} . Namely, we minimize the following objective \mathcal{O}_{joint} to train our end-to-end recommendation model:

$$\mathcal{O}_{joint} = \alpha \mathcal{O}_{gcn} + (1 - \alpha) \mathcal{O}_{seq} \quad (15)$$

Notice that POI recommendation process for the given user u is the same as in Sect. 4.4.

5 Experiments

5.1 Datasets

We conduct extensive experiments on three public real-world large-scale datasets: Foursquare,² Gowalla³ and Brightkite.⁴ The basic statistics of these three datasets are summarized in Table 1.

- *Foursquare* This dataset contains 483,813 check-in records generated by 4163 users who live in California from December 2009 to July 2013.
- *Gowalla* Gowalla is a location-based social networking website where users share their locations by checking-in. We choose data from Asian area for our experiments. It includes 251,378 check-in records generated by 6846 users over the period of February 2009 to October 2010.
- *Brightkite* Brightkite is also a location-based social networking service provider. We use the same selection strategy to obtain the check-in records generated by Asian users, which contains 572,739 records of 5677 users.

¹ <https://github.com/pfnet/optuna>.

² <https://sites.google.com/site/dbhonzhi/>.

³ <http://snap.stanford.edu/data/loc-Gowalla.html>.

⁴ <http://snap.stanford.edu/data/loc-Brightkite.html>.

Table 1 Basic statistics of three datasets

Dataset	Foursquare	Gowalla	Brightkite
# of users	4163	6846	5677
# of POIs	121,142	74,856	128,799
# of check-ins	483,813	251,378	572,739
# of categories	35	/	/
Time span	Dec. 2009–Jul. 2013	Feb. 2009–Oct. 2010	Apr. 2008–Oct. 2010

Notice that there are 35 POI categories in Foursquare, and no category information is attached to Gowalla and Brightkite datasets.

5.2 Evaluation Metrics

To evaluate the recommendation model performance, we use four widely used metrics, *i.e.*, Accuracy ($Acc@k$), Precision ($Pre@k$), Recall ($Rec@k$) and Normalized Discounted Cumulative Gain ($NDCG@k$), which are also used to evaluate top- k POI recommendation in [1, 23, 33, 34].

Let $\#hit@k$ denote the number of hits in the test set, and $|D_{Test}|$ is the number of all test records. $Acc@k$ is defined as:

$$Acc@k = \frac{\#hit@k}{|D_{Test}|}. \quad (16)$$

Let R_k denote the top- k POIs with the highest recommendation score, and T_k be the ground truth of the corresponding record, respectively. $Pre@k$ and $Rec@k$ are defined as:

$$Pre@k = \frac{1}{|D_{Test}|} \sum \frac{|R_k \cap T_k|}{|R_k|}, \quad (17)$$

$$Rec@k = \frac{1}{|D_{Test}|} \sum \frac{|R_k \cap T_k|}{|T_k|}. \quad (18)$$

To better measure the ranking quality, we further utilize $NDCG@k$, which assigns higher scores to POIs at top position ranks, to evaluate the model. $NDCG@k$ for each test case is defined as:

$$NDCG@k = \frac{DCG@k}{IDCG@k}, \quad (19)$$

where $DCG@k = \sum_{i=1}^k \frac{2^{rel_i} - 1}{\log_2(i+1)}$, $IDCG@k = \sum_{i=1}^k \frac{1}{\log_2(i+1)}$ and $rel_i = 1$ refers to the graded relevance of result ranked at position i . We use the binary relevance in our experiments, *i.e.*, $rel_i = 1$ if the recommended POI is in the ground truth, otherwise, $rel_i = 0$.

5.3 Baselines

We compare our model against the following baselines for successive POI recommendation:

- *Rank-GeoFM* [13] It is a ranking based geographical factorization model, which earns the embeddings of users and POIs by combining geographical and temporal influence in a weighting scheme.
- *ST-RNN* [15] ST-RNN is a RNN-based model with spatial and temporal contexts for next POI recommendation.
- *GE* [26] GE jointly learns the embedding of POIs, regions, time slots and word into a shared low dimensional space by constructing four bipartite graphs.
- *PEU-RNN* [18] It is a LSTM-based model that combines the user and POI embeddings, which are learned from Word2Vec, for modeling the dynamic user preference and successive transition influence.
- *SAE-NAD* [19] SAE-NAD exploits the self-attentive encoder to differentiate the user preference and the neighbor-aware decoder to incorporate the geographical context information for POI recommendation.

Notice that STGN [34] and ASPPA [33] are not compared in our experiment due to no publicly available source code. However, our PPR consistently outperforms ASPPA and STGN in terms of $Acc@k$ on both Foursquare and Gowalla datasets according to the experimental results reported in [33] (*e.g.*, PPR vs. STGN vs. ASPPA: 0.3008: 0.2: 0.2796 in $Acc@5$, 0.3935: 0.2592: 0.3371 in $Acc@10$ on Foursquare; PPR vs. STGN vs. ASPPA: 0.3835: 0.1947: 0.2363 in $Acc@5$, 0.4905: 0.2367: 0.2947 in $Acc@10$ on Gowalla).

To further validate the effectiveness of each component in our model, we design five variations of PPR:

- *PPR-RL* This is a simplified version of PPR, which does not use LSTM network for personalized preference modeling. After representation learning on \mathcal{G}_{dense} , we use $Score(v|v_c, u) = \vec{u} \cdot \vec{v} + \vec{v}_c \cdot \vec{v}$ to calculate the recommendation score, where v_c is the current location of the querying user u .
- *PPR-Seq* This variation does not model the sequential and geographical effect (*i.e.*, ignore POI-POI edges) in graph \mathcal{G}_{dense} , and the other components remain the same.
- *PPR-Soc* This variation does not model the social ties (*i.e.*, ignores User-User edges) in graph \mathcal{G}_{dense} , and the other components remain the same.
- *PPR-Den* This variation directly learns representations for users and POIs on graph \mathcal{G} , which does not densify the graph. And the other components remain the same.
- *PPR-GRU* In this variation, we use GRU to replace LSTM in user personalized preference modeling, and the other components remain the same.

Table 2 Performance comparison on Foursquare dataset

Methods	<i>Acc@5</i>	<i>Acc@10</i>	<i>Pre@5</i>	<i>Pre@10</i>	<i>Rec@5</i>	<i>Rec@10</i>	<i>NDCG@5</i>	<i>NDCG@10</i>
Rank-GeoFM	0.2456	0.2983	0.0618	0.0413	0.0509	0.0669	0.0683	0.0468
ST-RNN	0.1642	0.2150	0.0167	0.0118	0.1207	0.1790	0.0175	0.0152
GE	0.1357	0.3100	0.0378	0.0342	0.1579	0.1919	0.0431	0.0362
PEU-RNN	0.2021	0.2775	0.0495	0.0276	0.1888	0.2848	0.0494	0.0375
SAD-NAE	0.2429	0.3221	0.0588	0.0442	0.0333	0.0505	0.0672	0.0542
PPR	0.3008	0.3935	0.0698	0.0501	0.2471	0.3387	0.0802	0.0628
GCN-LSTM	0.3165	0.3971	0.0726	0.0521	0.2503	0.3434	0.0815	0.0637

Best two for each column are shown in bold

Table 3 Performance comparison on Gowalla dataset

Methods	<i>Acc@5</i>	<i>Acc@10</i>	<i>Pre@5</i>	<i>Pre@10</i>	<i>Rec@5</i>	<i>Rec@10</i>	<i>NDCG@5</i>	<i>NDCG@10</i>
Rank-GeoFM	0.2162	0.2643	0.0647	0.0453	0.0887	0.1180	0.0696	0.0499
ST-RNN	0.1865	0.2246	0.0278	0.0217	0.0817	0.1075	0.0606	0.0574
GE	0.1763	0.4060	0.0391	0.0203	0.1363	0.3135	0.0813	0.0157
PEU-RNN	0.3329	0.3766	0.0663	0.0390	0.2504	0.3613	0.0919	0.0627
SAD-NAE	0.3273	0.4300	0.0849	0.0645	0.1102	0.1600	0.0956	0.0777
PPR	0.3835	0.4905	0.0936	0.0687	0.2573	0.3430	0.1055	0.0840
GCN-LSTM	0.3895	0.4986	0.0974	0.0694	0.2648	0.3433	0.1116	0.0862

Best two for each column are shown in bold

5.4 Parameter Setting

In order to make our model satisfactory to the scenario of POI recommendation in real world, we first sort the check-in records of each user in chronological order. Afterward, we filter the POIs visited by less than five users and the users with less than ten check-in records according to [35]. Following [14, 34], we choose the first 80% of each user's check-ins in chronological order as train data, the remaining 20% as test data.

We use the source code released by their authors for baselines. We set learning rate to 0.0025 in graph embedding, embedding dimension d to 128, the number of negative samples to 5, threshold θ to 24 hours, κ to -2, ε to 0.5 and in/out-degree threshold ρ to 400. For the hyperparameter α , we perform optuna, a Bayesian hyperparameter optimization tools, and the search range for weight is set to [0, 1]. Following [8], we uniformly set the next time period as $\tau = 6$ hours for all methods unless stated otherwise, and other parameters of all baselines are tuned to be optimal. In the experiment, we use a two-layer stacked LSTM, the hidden state size is 128. The learning rate of LSTM is set to 0.001 with epoch decay, which makes the learning rate becomes 1/10 of the original value when the number of training rounds reaches 75%.

5.5 Performance Comparison

First, we evaluate the overall performance of our model PPR and GCN-LSTM compared with five baselines on three real-world datasets. We repeat 10 runs for all

methods on each dataset and report average $Acc@k$, $Pre@k$, $Rec@k$ and $NDCG@k$ in Tables 2, 3 and 4, respectively. The best two are shown in bold.

From Table 2, we observe that PPR is significantly better than all baselines in terms of four evaluation metrics on Foursquare dataset. Specifically, PPR achieves 0.3008 in $Acc@5$ and 0.3935 in $Acc@10$, improving 22.5% and 22.2% over second-best baseline Rank-GeoFM and SAD-NAE, respectively. Additionally, our PPR slightly outperforms the strong baselines (e.g., SAD-NAE) in $Pre@k$, but it is significantly better than the strong baselines in $Rec@k$.

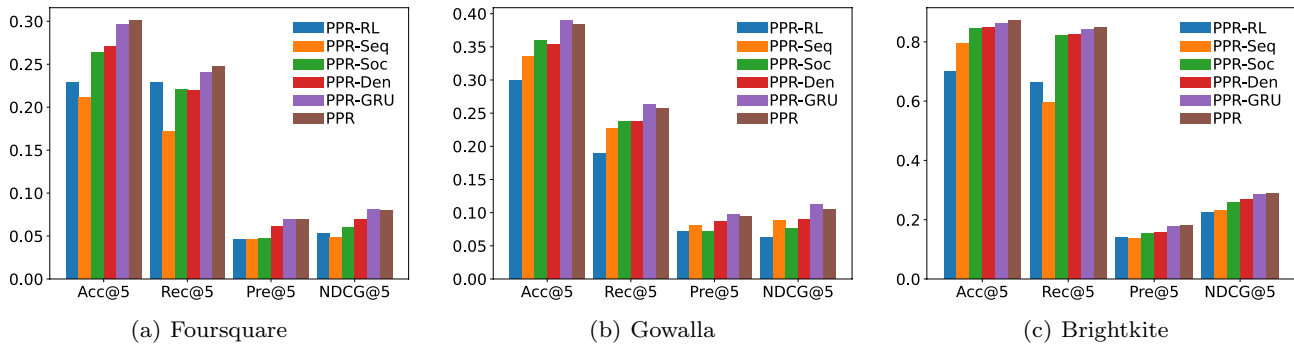
As depicted in Table 3, our PPR also significantly outperforms all baselines in terms of $Acc@k$, $Pre@k$, $Rec@k$ and $NDCG@k$ on Gowalla dataset. In particular, PPR performs better than the second-best baseline by 14.6% in $Acc@k$ and 9.2% in $NDCG@k$ on average. PPR shows slightly poor performance compared to PEU-RNN in terms of $Rec@10$. This phenomenon can be explained that PEU-RNN uses a distance constraint, which may significantly reduce the potential POIs as k increases.

As we can see in Table 4, PPR consistently significantly outperforms all baselines in terms of all evaluation metrics on Brightkite dataset. PPR achieves the state-of-the-art performance, e.g., 0.8717 in $Acc@5$ and 0.8485 in $Rec@5$. More specifically, our PPR achieves about 21.3%, 24.4%, 22.2% and 22.4% improvement compared to state-of-the-art RNN-based method PEU-RNN in terms of $Acc@5$, $Pre@5$, $Rec@5$ and $NDCG@5$, respectively. Furthermore, all methods achieve better performance on Brightkite than

Table 4 Performance comparison on Brightkite dataset

Methods	Acc@5	Acc@10	Pre@5	Pre@10	Rec@5	Rec@10	NDCG@5	NDCG@10
Rank-GeoFM	0.3681	0.4270	0.0968	0.0618	0.2497	0.2983	0.1058	0.0700
ST-RNN	0.2396	0.3540	0.0389	0.0394	0.2279	0.3400	0.1166	0.1074
GE	0.1903	0.4259	0.0869	0.0483	0.1303	0.4119	0.1313	0.1217
PEU-RNN	0.7187	0.7383	0.1437	0.0720	0.6944	0.7204	0.2348	0.1538
SAD-NAE	0.2578	0.3383	0.0645	0.0499	0.0703	0.1047	0.0708	0.0584
PPR	0.8717	0.8966	0.1788	0.0927	0.8485	0.8741	0.2875	0.1889
GCN-LSTM	0.8809	0.8974	0.1789	0.0949	0.8403	0.8890	0.2944	0.1973

Best two for each column are shown in bold

**Fig. 3** Performance comparison of variations

the other datasets. This is because users in Brightkite have more check-in records than users in Foursquare and Gowalla on average, which may enable all methods to model users' behavior and preference more accurately.

From Tables 2, 3 and 4, compared with PPR, our extended end-to-end model GCN-LSTM further improves the recommendation performance *w.r.t.* all metrics.

5.6 Ablation Study

To explore the benefits of incorporating the sequential and geographical effect, densifying technique and modeling personalized preference into PPR, respectively, we compare our model with four carefully designed variations, *i.e.*, PPR-RL, PPR-Seq, PPR-Den and PPR-GRU. We show the results in terms of *Acc@5*, *Pre@5*, *Rec@5*, and *NDCG@5* on three datasets in Fig. 3.

Based on the results, we have the following observations: First, PPR achieves the best performance in most cases on three datasets, indicating that PPR benefits from simultaneously considering the various contextual factors and personalized preference in a joint way. Second, the contributions of different components to recommendation performance are different. Sequential and geographical effect and modeling personalized preference have comparable importance, specifically, the later contributes more on Gowalla, and the former contributes more on Foursquare. And both of them are necessary for improving performance. Furthermore,

through the comparison of PPR and PPR-Den, it is obvious that the densifying trick works for alleviating the data sparse issue. Third, removing social relationships would degrade our model performance. However, the performance degradation is not significant, which means our model does not rely on social ties heavily. There may be two reasons: On the one hand, it may be related to our proposed graph modeling, which effectively captures spatial correlations between the related users. On the other hand, there is some noise (*e.g.*, two users who have social relations differ greatly in their check-in preferences) in social relationships. When social ties are removed, some noise is also removed simultaneously. This is why *PPR-Soc* and *PPR-Den* have the similar performance. Fourth, PPR and PPR-GRU exhibit a decent performance compared to other variations, which indicates that sequential pattern and users' dynamic and personalized preference play an important role in location-based recommendation.

5.7 Sensitivity of Hyperparameters

We now investigate the sensitivity of our models (*i.e.*, PPR and GCN-LSTM) compared against three strong baselines (*i.e.*, Rank-GeoFM, PEU-RNN, and SAE-NAD) with respect to the important parameters, including embedding dimension d , the number of recommended POIs k , and next time period τ . To clearly show the influence of these parameters, we report *Acc@5* with different parameter settings on

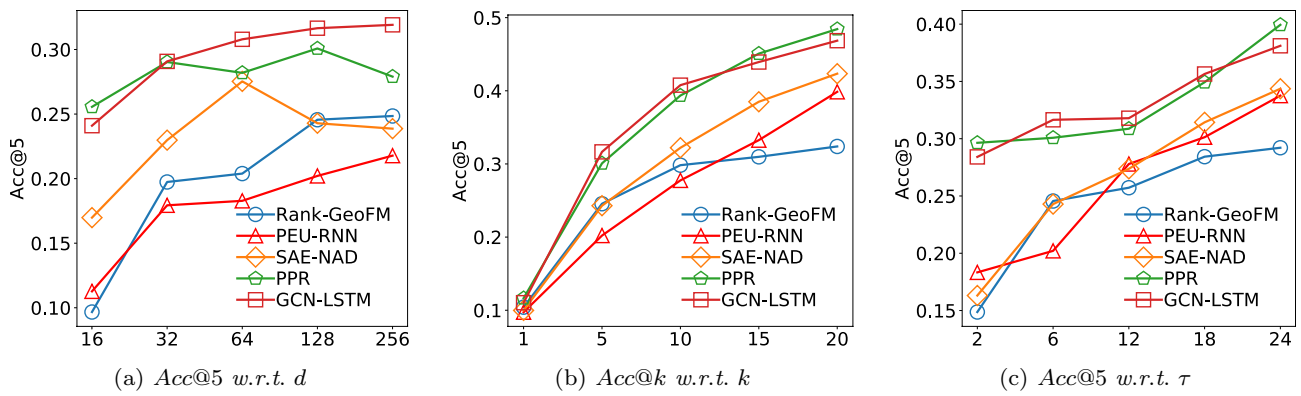


Fig. 4 Parameter sensitivity w.r.t. parameter d , k and τ on Foursquare

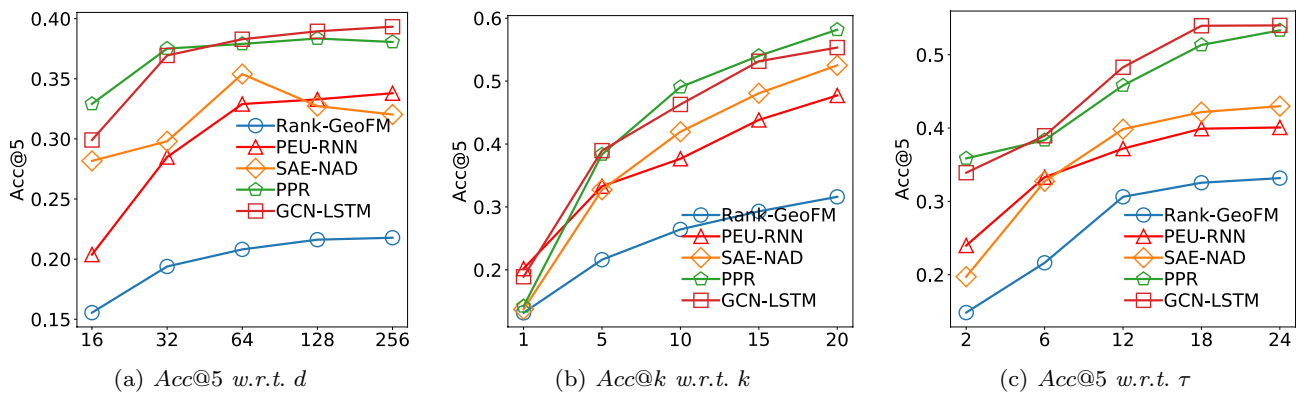


Fig. 5 Parameter sensitivity w.r.t. parameter d , k and τ on Gowalla

Foursquare and Gowalla datasets. Figs. 4 and 5 show the experimental results.

As shown in Figs. 4a and 5a, PPR and GCN-LSTM achieve better performance compared to the three strong baselines with the increasing number of dimension d . GCN-LSTM remains basically stable when d reaches 128 or more. Meanwhile, PPR achieves the best result when $d = 128$ and then begins to decline as d further increases.

From the results in Figs. 4b and 5b, we can see that the recommendation accuracy of all methods increases as k increases. This is expected, because the more results are recommended, the easier they are to fall into the ground truth. However, we also observe that our PPR and GCN-LSTM exhibit an increasing performance improvement compared to all baselines, as k increases. In Figs. 4c and 5c, as τ increases, our models are also consistently better than the strong baselines. More specifically, PPR improves the recommendation accuracy more significantly for near future prediction (*e.g.*, $\tau = 2$ vs. $\tau = 12$), indicating that our PPR can effectively capture users' personalized preferences, especially short-term preferences.

Next, we evaluate the impact of ρ on our PPR by varying ρ from 0 to 600. The results are reported in Table 5.

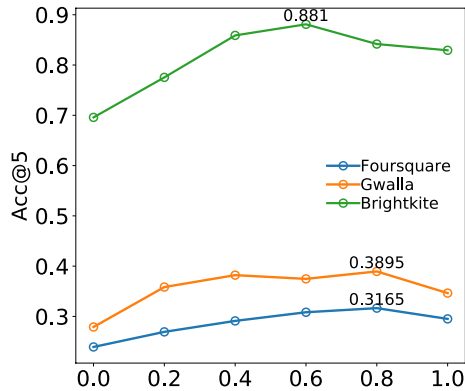
On Foursquare, Gowalla and Brightkite datasets, PPR achieves the best performance when $\rho = 300$, $\rho = 400$ and $\rho = 100$, respectively. For Brightkite, PPR achieves the best performance when ρ is small compared to Foursquare and Gowalla. The main reason may lay that users in Brightkite have denser check-in records than users in Foursquare and Gowalla on average, which leads to a small ρ . Additionally, with the increase of ρ , the overall performance is increasing gradually and then falls. This may be because as ρ increases, some noise edges are generated, resulting in the overall performance degradation.

Finally, we evaluate the impact of hyperparameter α on recommendation performance of our GCN-LSTM. From Fig. 6, we observe that both \mathcal{O}_{gcn} (representation learning) and \mathcal{O}_{seq} (sequential modeling) have their own role in POI recommendation. $Acc@5$ of GCN-LSTM first increases to the maximum value and then decreases as α increases. This is intuitive because both representation learning and sequential modeling are essential for a precise recommendation.

Table 5 Parameter sensitivity *w.r.t.* parameter ρ on $Acc@5$

ρ	0	100	200	300	400	500	600
Foursquare	0.2701	0.2919	0.2996	0.3036	0.3008	0.2903	0.2984
Gowalla	0.3535	0.3638	0.3652	0.3691	0.3835	0.3673	0.3833
Brightkite	0.8469	0.8764	0.8663	0.8675	0.8717	0.8669	0.8675

The best value for each row is shown in bold

**Fig. 6** Parameter sensitivity *w.r.t.* parameter α

6 Conclusion

In this work, we propose a novel spatio-temporal representation learning model for personalized POI recommendation. By incorporating the user-POI relation, sequential effect, geographical effect and social ties, we construct a heterogeneous network. Afterward, we exploit the embedding technique to learn the latent representation of users and POIs. In light of recent success of RNN on sequential prediction problem, we feed the spatio-temporal network with concatenated user and POI embedding sequences for capturing the users' dynamic and personalized preference. The results on three real-world datasets demonstrate the superiority of our proposal over state-of-the-art baselines. Furthermore, we explore the importance of each factor in improving recommendation performance. We observe that sequential effect, geographical effect, and users' dynamic and personalized preference play a vital role in POI recommendation task.

Acknowledgements This work is partially supported by the National Natural Science Foundation of China under Grant Nos. 62176243, 61773331 and 41927805, the National Key Research and Development Program of China under Grant No. 2018AAA0100602, and the Natural Science Foundation of Shandong Province under Grant No. ZR2020QF030. Corresponding author: Yanwei Yu.

Author Contributions SD contributed to methodology, software, data curation, visualization, and writing—original draft. YY contributed to conceptualization, supervision, writing—review and editing, and funding acquisition. HF contributed to data curation, validation, investigation, and funding acquisition. JD contributed to writing—review and editing, and funding acquisition.

Funding Yanwei Yu is supported by the National Natural Science Foundation of China under Grant Nos. 62176243 and 61773331. Junyu Dong is supported by the National Natural Science Foundation of China under Grant Nos. U1706218 and 41927805, the National Key Research and Development Program of China under Grant No. 2018AAA0100602. Hao Fan is supported by the Natural Science Foundation of Shandong Province under Grant No. ZR2020QF030.

Data Availability The used datasets in experiments are publicly available: Foursquare is available at: <https://sites.google.com/site/dbhongzhi/>, Gowalla is available at: <http://snap.stanford.edu/data/loc-Gowalla.html> and Brightkite is available at: <http://snap.stanford.edu/data/loc-Brightkite.html>.

Code Availability The source code of our method is available at: <https://www.anonymous.4open.science/r/DSE-1BEC>.

Declarations

Conflict of Interest The authors declare that there is no conflict of interests regarding the publication of this article.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Chang B, Park Y, Park D, Kim S, Kang J (2018) Content-aware hierarchical point-of-interest embedding model for successive poi recommendation. In: IJCAI, pp 3301–3307
- Chen J, Ma T, Xiao C (2018) Fastgcn: fast learning with graph convolutional networks via importance sampling. arXiv preprint [arXiv:1801.10247](https://arxiv.org/abs/1801.10247)
- Chen M, Liu Y, Yu X (2014) Nlpmm: a next location predictor with Markov modeling. In: PAKDD. Springer, pp 186–197
- Cheng C, Yang H, Lyu MR, King I (2013) Where you like to go next: successive point-of-interest recommendation. In: IJCAI
- Cho E, Myers SA, Leskovec J (2011) Friendship and mobility: user movement in location-based social networks. In: KDD, pp 1082–1090

6. Dai S, Yu Y, Fan H, Dong J (2021) Personalized poi recommendation: spatio-temporal representation learning with social tie. In: DASFAA (1), pp 558–574
7. Feng S, Cong G, An B, Chee YM (2017) Poi2vec: geographical latent representation for predicting future visitors. In: AAAI, pp 102–108
8. Feng S, Tran LV, Cong G, Chen L, Li J, Li F (2020) Hme: a hyperbolic metric embedding approach for next-poi recommendation. In: SIGIR, pp 1429–1438
9. Kim JS, Jin H, Kavak H, Rouly OC, Crooks A, Pfoser D, Wenk C, Züfle A (2020) Location-based social network data generation based on patterns of life. In: 2020 21st IEEE international conference on mobile data management (MDM). IEEE, pp 158–167
10. Kipf TN, Welling M (2016) Semi-supervised classification with graph convolutional networks. arXiv preprint [arXiv:1609.02907](https://arxiv.org/abs/1609.02907)
11. Levandoski JJ, Sarwat M, Eldawy A, Mokbel MF (2012) Lars: a location-aware recommender system. In: ICDE. IEEE, pp 450–461
12. Li K, Lu G, Luo G, Cai Z (2020) Seed-free graph de-anonymization with adversarial learning. In: CIKM, pp 745–754
13. Li X, Cong G, Li XL, Pham TAN, Krishnaswamy S (2015) Rankgeofm: A ranking based geographical factorization method for point of interest recommendation. In: SIGIR, pp 433–442
14. Li X, Han D, He J, Liao L, Wang M (2019) Next and next new poi recommendation via latent behavior pattern inference. *ACM Trans Inf Syst (TOIS)* 37(4):1–28
15. Liu Q, Wu S, Wang L, Tan T (2016) Predicting the next location: a recurrent model with spatial and temporal contexts. In: AAAI
16. Liu Z, Huang C, Yu Y, Fan B, Dong J (2020) Fast attributed multiplex heterogeneous network embedding. In: CIKM, pp 995–1004
17. Liu Z, Huang C, Yu Y, Song P, Fan B, Dong J (2020) Dynamic representation learning for large-scale attributed networks. In: CIKM, pp 1005–1014
18. Lu YS, Shih WY, Gau HY, Chung KC, Huang JL (2019) On successive point-of-interest recommendation. *World Wide Web* 22(3):1151–1173
19. Ma C, Zhang Y, Wang Q, Liu X (2018) Point-of-interest recommendation: exploiting self-attentive autoencoders with neighbor-aware influence. In: CIKM, pp 697–706
20. Mikolov T, Chen K, Corrado G, Dean J (2013) Efficient estimation of word representations in vector space. arXiv preprint [arXiv:1301.3781](https://arxiv.org/abs/1301.3781)
21. Perozzi B, Al-Rfou R, Skiena S (2014) Deepwalk: online learning of social representations. In: KDD, pp 701–710
22. Tang J, Qu M, Wang M, Zhang M, Yan J, Mei Q (2015) Line: Large-scale information network embedding. In: WWW, pp 1067–1077
23. Wang Q, Yin H, Chen T, Huang Z, Wang H, Zhao Y, Viet Hung NQ (2020) Next point-of-interest recommendation on resource-constrained mobile devices. In: WWW, pp 906–916
24. Wu F, Souza A, Zhang T, Fifty C, Yu T, Weinberger K (2019) Simplifying graph convolutional networks. In: International conference on machine learning, pp 6861–6871. PMLR
25. Wu Y, Li K, Zhao G, Xueming Q (2020) Personalized long-and short-term preference learning for next poi recommendation. *IEEE Trans Knowl Data Eng*
26. Xie M, Yin H, Wang H, Xu F, Chen W, Wang S (2016) Learning graph-based poi embedding for location-based recommendation. In: CIKM, pp 15–24
27. Ye J, Zhu Z, Cheng H (2013) What's your next move: User activity prediction in location-based social networks. In: SDM. SIAM, pp 171–179
28. Ye M, Yin P, Lee WC (2010) Location recommendation for location-based social networks. In: SIGSPATIAL, pp 458–461
29. Yuan Q, Cong G, Ma Z, Sun A, Thalmann NM (2013) Time-aware point-of-interest recommendation. In: SIGIR, pp 363–372
30. Zhang JD, Chow CY, Li Y (2014) Lore: exploiting sequential influence for location recommendations. In: SIGSPATIAL, pp 103–112
31. Zhang Z, Li C, Wu Z, Sun A, Ye D, Luo X (2020) Next: a neural network framework for next poi recommendation. *Front Comp Sci* 14(2):314–333
32. Zhang Z, Liu Y, Zhang Z, Shen B (2019) Fused matrix factorization with multi-tag, social and geographical influences for poi recommendation. *World Wide Web* 22(3):1135–1150
33. Zhao K, Zhang Y, Yin H, Wang J, Zheng K, Zhou X, Xing C (2020) Discovering subsequence patterns for next poi recommendation. In: IJCAI, pp 3216–3222
34. Zhao P, Luo A, Liu Y, Zhuang F, Xu J, Li Z, Sheng VS, Zhou X (2020) Where to go next: a spatio-temporal gated network for next poi recommendation. *IEEE Trans Knowl Data Eng*
35. Zhao S, Zhao T, King I, Lyu MR (2017) Geo-teaser: geo-temporal sequential embedding rank for point-of-interest recommendation. In: WWW, pp 153–162
36. Zhou J, Cui G, Hu S, Zhang Z, Yang C, Liu Z, Wang L, Li C, Sun M (2020) Graph neural networks: a review of methods and applications. *AI Open* 1:57–81
37. Zhu Y, Li H, Liao Y, Wang B, Guan Z, Liu H, Cai D (2017) What to do next: modeling user behaviors by time-lstm. In: IJCAI, pp 3602–3608