



Mining dynamic preferences from geographical and interactive correlations for next POI recommendation

Jieyu Ren¹ · Mingxin Gan¹

Received: 6 April 2022 / Revised: 7 June 2022 / Accepted: 22 August 2022 / Published online: 22 September 2022
© The Author(s), under exclusive licence to Springer-Verlag London Ltd., part of Springer Nature 2022

Abstract

Next point-of-interest recommendation has become an increasingly significant requirement in location-based social networks. Recently, RNN-based methods have shown promising advantages in next POI recommendation due to their superior abilities in modeling sequential transitions of user behaviors. Despite their success, however, exploring complex correlations between POIs and capturing user dynamic preferences are still challenging issues. To overcome the limitations, we propose a novel framework named MPGI (Mining Preferences from Geographical and Interactive Correlations) for next POI recommendation. Specifically, we first design a POI correlation modeling layer to capture geographical distances and interactive correlations between all of POI pairs. Then, we fuse relevant signals from highly correlated POIs into target POI for high-quality POI representations. Furthermore, for user long- and short-term preferences modeling, we propose position-aware attention unites and attention network to dynamically select the most valuable information in check-in trajectories. Experimental results on two real-world datasets demonstrate that MPGI consistently outperforms the state-of-the-art methods.

Keywords Next POI recommendation · Geographical distance · Interactive correlation · User preference · Node2Vec · Position-aware attention

1 Introduction

With the advances in wireless community technologies, location-based social networks (LBSNs) such as Foursquare and Gowalla have grown rapidly and become pervasive in our daily lives. Millions of people on LBSNs share their experience on points of interest (POIs) when they check-in at locations. The huge amount of accumulated check-in data could be utilized to determine users' preferences on POIs and further predict their mobility behaviors. Therefore, many POI recommendation methods have been proposed [1–4]. As an essential service of LBSNs, POI recommendation not only improves user experience but also increases advertising revenue of e-commerce platforms [5].

✉ Mingxin Gan
ganmx@ustb.edu.cn

¹ School of Economics and Management, University of Science and Technology Beijing, Beijing, China

Among a variety of POI recommendation tasks, next POI recommendation is absolutely the most challenging one. It aims to capture sequential dependencies of user travel behaviors and recommend exactly the next POI to visit [6]. In recent years, a large number of methods have been proposed to model such sequential transitions. Previous studies mainly adopt Markov Chains technique [7, 8], which exploit user's latest check-in in the check-in sequence. Such models would degrade recommendation performance since user's next check-in is also highly dependent on other visited POIs [9]. With the superiority of modeling long-term sequential transitions, recurrent neural networks (RNNs)-based methods are currently the state-of-the-art solutions [9–11]. Based on the assumption that the spatiotemporal factors along with nearby check-ins restrict user movements [10], these models first integrate various contextual information, such as transition time and transition distance between consecutive visited POIs, with user's check-in sequence, and further apply RNN to estimate user's temporal preferences on POIs to predict the next visit.

Although existing methods have achieved satisfactory results, there are still two major limitations.

Firstly, most of previous works only capture the transition distances between consecutive visited POIs in users' independent check-in sequences [10–15]. However, since a POI is geographically adjacent to multiple other POIs, the geographically adjacent relations between POIs are not thoroughly captured. In addition, the collaborative relations between POIs, whose importance has been verified in collaborative filtering-based models [16], are barely considered in next POI recommendation. In real scenarios, the correlations between POIs are fairly complex. First, it is common that people would consecutively visit geographically adjacent POIs. As shown at the bottom part of Fig. 1, POI l_3 is geographically close to both l_4 and l_5 . More generally, a POI is geographically adjacent to multiple other POIs in real world. Therefore, it would be better to construct a POI graph to fully capture the geographically adjacent relations for better recommendations.

In addition, collaborative filtering-based methods have demonstrated that items interacted by the same users are similar to some extent [17, 18]. As shown at the top part of Fig. 1, l_2 and l_5 are simultaneously visited by all users, *i.e.*, u_1, u_2 and u_3 . Thus, l_2 and l_5 tend to share similar user preferences. If another user has visited POI l_2 , the user may also be fond of l_5 . Inspired by this, we define the “interactive correlation” to capture the important collaborative signals from similar POIs to refine POI representations. By thoroughly exploring the geographical distances and interactive correlations using POI graph, the POI characterizations will be refined and the recommendation performance will be further improved.

Secondly, most existing models only capture user's short-term preference based on several recent check-ins [12, 19]. Nevertheless, user's long-term preference, which contains both relevant and irrelevant information, also has contributions to user's behaviors. Several recent studies tried to capture both long- and short-term preferences for user representation [20, 21]. However, they failed to adaptively differentiate the importance of POIs to find valuable check-ins of users. In fact, different POIs have distinguished meanings to a certain user according to their significances at different time-points. Therefore, weighting relevant POIs dynamically is necessary for enhancing recommendation performance.

To overcome the above limitations, we propose a novel method named MPGI (Mining Preferences from Geographical and Interactive Correlations) for next POI recommendation. Specifically, we design a POI correlation modeling layer to capture geographical distances and interactive correlations between all of POI pairs. Then, we connect highly correlated POIs through weighted edges and employ Node2Vec to train embeddings of POIs. Furthermore, we develop position-aware attention units and an attention network to adaptively select the most valuable information, so as to obtain both of user's long-term and short-term preferences

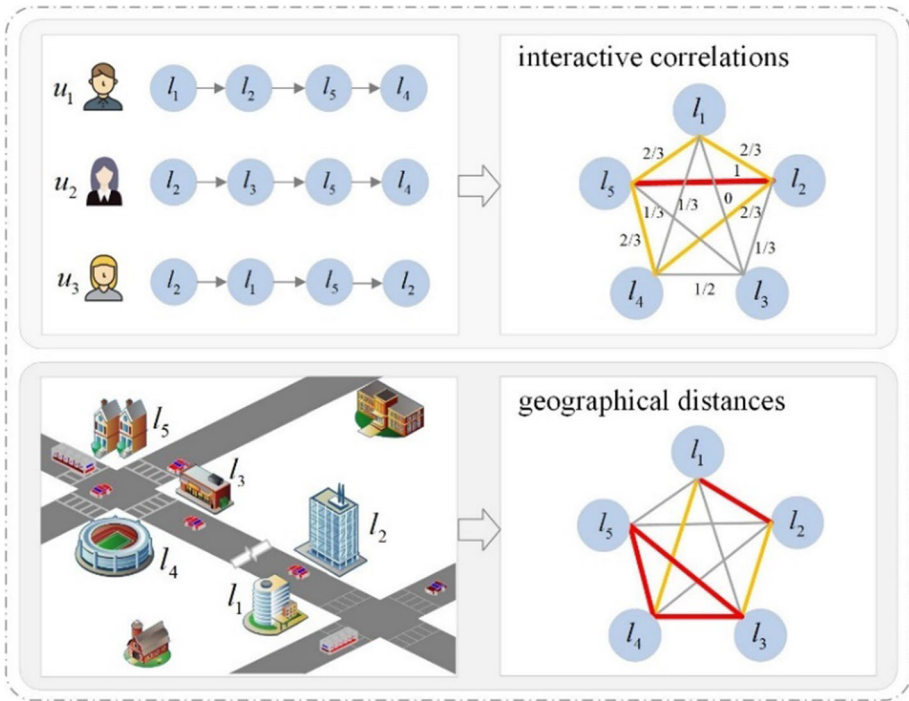


Fig. 1 A toy example of POI correlation modeling

dynamically. Finally, we compute the visiting probability distribution over all POIs using a softmax function in the prediction layer.

Our primary contributions could be summarized as follows:

- We propose a novel next POI recommendation model to learn POI embeddings by explicitly capturing the complex correlations, *i.e.*, geographical distances and interactive correlations between all of POI pairs.
- We design position-aware attention units and an attention network to adaptively select the most valuable check-ins for modeling user’s long- and short-term preferences.
- We conduct extensive experiments on two real-world datasets, which demonstrates the effectiveness of MPGI against the state-of-the-art methods in next POI recommendation.

2 Related work

In this section, we delve into relevant previous works on general POI recommendation, next POI recommendation and attention mechanism in recommendations.

2.1 General POI recommendation

General POI recommendation captures user’s general preferences for POIs without considering the sequential transitions of user behaviors. Generally, the most well-known

recommendation method is collaborative filtering (CF). Earlier studies mainly integrated CF with various contextual information to estimate user's general preferences [22–26]. For example, Ren et al. [22] presented a context-aware probabilistic matrix factorization method which fused five kinds of contextual information comprehensively. Furthermore, Davtalab et al. [27] integrated POI similarity and user similarity into the framework of probabilistic matrix factorization to capture the implicit correlations. Nowadays, deep learning has been introduced into POI recommendation to model nonlinear complex relationships between users and POIs [5, 28–32]. For example, Pang et al. [5] developed a structure of local-to-global to attentively mine more hidden information from multiple features of POIs. Considering that user preference was the key factor influencing user behaviors, Liu et al. [29] proposed a pairwise ranking-based method to enhance the feature learning of POIs and preference learning of users.

2.2 Next POI recommendation

Compared with general POI recommendation, next POI recommendation emphasizes sequential dependencies of check-ins and models dynamic evolutions of user preferences [9]. Earlier studies mainly used the first-order Markov chains to capture the transition relationships between consecutive POIs [7, 8, 33]. For instance, Cheng et al. [8] developed FPMC-LR model which not only applied personalized Markov chains to the check-in sequence, but also considered user's movement constraints. However, these methods only exploited user's latest check-in of a sequence, thus failed to capture high-order transition relationships between previous visited POIs and the predicted one.

Recently, recurrent neural networks (RNNs) have been widely applied in sequential recommendations to model the long-term sequential patterns. Numerous studies integrated various contextual information into RNNs for next POI recommendation. For example, Liu et al. [13] considered time intervals and geographical distances between consecutive check-ins and integrated them into vanilla RNN. DSPR explored user's preferences and real-time demands simultaneously, both of which are crucial factors to determine user's behaviors [11]. To overcome the data sparsity problem, Yu et al. [34] incorporated POI category and geographical influence into RNN to reduce search space. DeepMove was an earlier attempt to combine user's long-term preference and short-term preference, which, respectively, represented user's behavior periodicity and real-time needs [14]. To pursue this further, Sun et al. [21] developed a nonlocal network to model long-term preference and a geo-dilated RNN to model short-term preference. Similarly, PLSPL [35], NATR [20] and RTPM [36] learned user's both long-term and short-term preferences to get a comprehensive user representation. However, complex relationships between POI pairs are seldomly considered in the above works. GT-HAN [9] was a beneficial attempt which developed three matrices to explicitly capture geographical interactions between POI pairs in user's check-in history.

Although RNN-based methods have demonstrated great superiority in next POI recommendation, one major issue is that the geographical distances and interactive correlations between any pair of POIs in the entire POI set are not thoroughly explored. In contrast, the proposed model constructs a POI graph to incorporate geographical distances and interactive correlations between POIs for more accurate POI representations. At the same time, it is of great importance to capture user long- and short-term preferences simultaneously. However, existing methods fail to differentiate the importance degrees of POIs in user's check-in trajectories, thus cannot capture user dynamic preferences to boost recommendation.

2.3 Attention mechanism in recommendations

The attention mechanism [37], which derives from the study of human vision, has shown great power in recommendation systems. The advantage of attention mechanism lies in its ability to adaptively capture the important parts of input data. For example, Zhou et al. [38] employed attention mechanism in the interest evolving layer to strengthen the effects of relative information during interest evolution, and further captured the latent user interests dynamically. Considering the importance of the latest interacted item that usually plays a decisive role in the next behavior, SR-GNN [39] and CaSe4SR [40] took last item's representation as query vector to attentively represent the global session preference. To recommend POIs in the target time, Shi et al. [41] assigned visited POIs into corresponding time slots and designed a temporal-level attention mechanism to adaptively give different weights to history time slots. Zheng et al. [42] analyzed the contributions of attributes and interaction behaviors using attention mechanism to better represent users and items. Similarly, Ni et al. [43] developed multi-attention-based convolutional neural networks to refine the features of users and items. Inspired by previous attempts, we will leverage attention mechanism to adaptively select valuable check-in information to improve recommendation quality.

3 Preliminaries

Formally, the set of users is defined as $U = \{u_1, u_2, \dots, u_{|U|}\}$ and the set of POIs is defined as $L = \{l_1, l_2, \dots, l_{|L|}\}$, where $|U|$ and $|L|$ are the total number of users and POIs, respectively. Since POIs are spatial entities with specific longitudes and latitudes, each POI l could be geocoded as $(\text{lon}_l, \text{lat}_l)$. For each user $u \in U$, we split his/her long check-in sequence into multiple trajectories $S = \{S_1, S_2, \dots, S_n\}$, according to time intervals between consecutive check-ins. Here n represents the total number of trajectories. Each trajectory $S_h \in S$ is denoted by $S_h = \{l_1^h, l_2^h, \dots, l_{|S_h|}^h\}$, representing a set of POIs visited by u in chronological order. Key notations and explanations are listed in Table 1.

Given history trajectories $S_{his} = \{S_1, S_2, \dots, S_{n-1}\}$ and current trajectory S_n of target user $u \in U$, the recommendation task is to generate Top- k POIs that u will probably visit next.

4 The proposed method

In this part, we detail the proposed model MPGI. The illustration of the overall framework is shown in Fig. 2. MPGI consists of three parts: POI correlation modeling layer, long- and short-term preferences modeling layer and prediction layer. Specifically, POI correlation modeling layer is included to capture the geographical distances and interactive correlations between all of POI pairs for learning POI representations. Sequentially, position-aware attention units and attention network are developed to select the most valuable information for better inferring user long- and short-term preferences. Lastly, comprehensive user representation is obtained to calculate the possibilities of POIs being visited next by the target user. Top- k POIs will be selected as the recommendation result.

Table 1 Notations and explanations

Notation	Explanations
U, L	Sets of users and POIs
S	Set of all trajectories of user u
S_{his}	Set of all history trajectories of user u
S_n	Current trajectory of user u
\mathbf{D}	The geographical distance matrix
\mathbf{I}	The interactive correlation matrix
U_l	Set of users who have visited POI l
$dist$	The threshold of geographical distances
θ	The threshold of interactive correlations
e_l	Embedding of POI l trained by Node2Vec
\mathbf{E}	POI embedding matrix
\mathbf{P}	Position embedding matrix
$\{\mathbf{W}\}, \mathbf{q}, \{\mathbf{b}\}$	Weight matrices, weight vector and bias vectors
s_{Short}	Short-term preference representation of user u
s_{Long}	Long-term preference representation of user u
n	Total number of trajectories
m	Maximum length of all trajectories
d	Embedding size of POIs and position vectors
$ \cdot $	The Cardinality of the set

4.1 POI correlation modeling layer

Most studies only capture the transition time and transition distance between consecutive check-ins, neglecting complex correlations between POIs. Consequently, the features of highly correlated POIs could hardly be fused into the target POI, which heavily constrains the expressiveness of POI embeddings. To this end, we take the complex correlations between POIs into consideration. Here, the “complex” property stems from two aspects. First, the *geographical distances* and *interactive correlations* are supposed to be considered simultaneously. Second, these two correlations of *all of POI pairs*, not a part of POI pairs, will be thoroughly captured in this paper. After capturing such correlations via Node2Vec training, similar POIs would have similar representations in specific aspects, laying a foundation for the following user preferences learning.

First, the geographical distance matrix $\mathbf{D} \in \mathbb{R}^{|L| \times |L|}$ is defined as follows:

$$\mathbf{D} = \begin{bmatrix} d_{1,1} & d_{1,2} & \cdots & d_{1,|L|} \\ d_{2,1} & d_{2,2} & \cdots & d_{2,|L|} \\ \vdots & \vdots & \ddots & \vdots \\ d_{|L|,1} & d_{|L|,2} & \cdots & d_{|L|,|L|} \end{bmatrix} \tag{1}$$

$$\begin{aligned} d_{pq} &= \text{Haversine}(\text{lon}_{l_p}, \text{lat}_{l_p}, \text{lon}_{l_q}, \text{lat}_{l_q}) \\ &= 2R \arcsin \left(\sqrt{\sin^2 \left(\frac{\text{lat}_{l_p} - \text{lat}_{l_q}}{2} \right) + \cos(\text{lat}_{l_p}) \cos(\text{lat}_{l_q}) \sin^2 \left(\frac{\text{lon}_{l_p} - \text{lon}_{l_q}}{2} \right)} \right) \end{aligned} \tag{2}$$

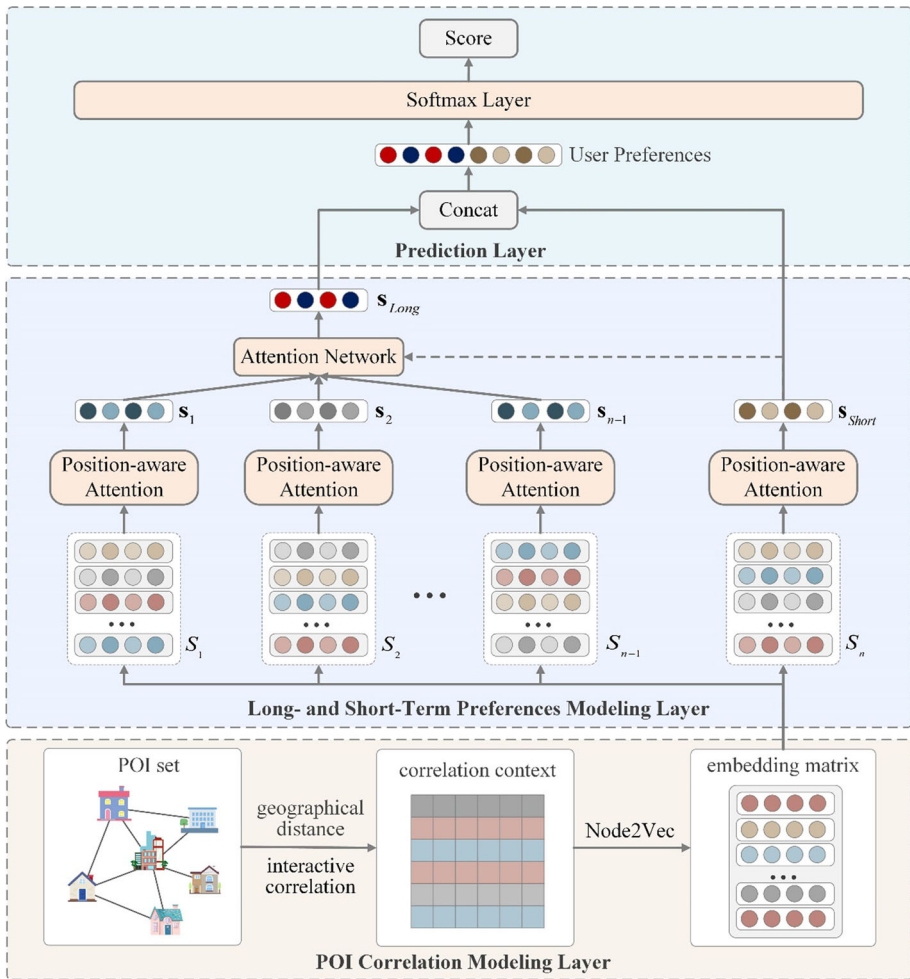


Fig. 2 The framework of the proposed model MPG1

where geographical distance d_{pq} between POI l_p and l_q is calculated by Haversine distance. R denotes the radius of the earth; lon_{l_p} and lat_{l_p} are the longitude and latitude of POI l_p ; lon_{l_q} and lat_{l_q} are the longitude and latitude of POI l_q .

Haversine distance is widely used in the existing studies [1, 21, 44], which assume that the earth is a sphere and calculate the great-circle distances between POIs along the surface of the sphere. Compared with other metrics such as Euclidean distance, the advantage of Haversine distance is that it takes the earth sphere into consideration to approximately estimate real path distances between POIs. Although users care more about the distance traveled by the actual path between the POIs, the travel distance data are not available. Therefore, we use Haversine distance as calculation method.

Next, the interactive correlation is defined as the collaborative relationship between POIs reflected in overall user check-in behaviors. For example, if two POIs have always been simultaneously visited by users, they are supposed to have high interactive correlations in

some aspects. In this way, we consider not only the geographical distances between POIs from the perspective of physical space, but also the interactive correlations observed from all users' check-in data.

The interactive correlation matrix $\mathbf{I} \in \mathbb{R}^{|L| \times |L|}$ is defined as follows:

$$\mathbf{I} = \begin{bmatrix} i_{1,1} & i_{1,2} & \cdots & i_{1,|L|} \\ i_{2,1} & i_{2,2} & \cdots & i_{2,|L|} \\ \vdots & \vdots & \ddots & \vdots \\ i_{|L|,1} & i_{|L|,2} & \cdots & i_{|L|,|L|} \end{bmatrix} \tag{3}$$

$$i_{pq} = \frac{|U_{l_p} \cap U_{l_q}|}{|U_{l_p} \cup U_{l_q}|} \tag{4}$$

where the interactive correlation i_{pq} between POI l_p and l_q is estimated by the Jaccard similarity of the visitor sets U_{l_p} and U_{l_q} .

We define two hyperparameters: the threshold of geographical distances $dist$ and the threshold of interactive correlations θ . For any pair of POIs l_p and l_q , if $d_{pq} \leq dist$ and $i_{pq} \geq \theta$ are simultaneously met, l_p and l_q are considered as highly correlated POIs, *i.e.*, l_p is one of the correlation context L_q^{Pos} for l_q , at the same time, l_q is one of the correlation context L_p^{Pos} for l_p . By traversing values in the geographical distance matrix and the interactive correlation matrix, the correlation context for each POI is obtained. Then, POIs and their corresponding correlation contexts are connected by edges to form a network. After that, we employ Node2Vec to learn representations of all POIs.

It is worth noting that the higher the interactive correlation value between l_p and l_q , the more similarities they share. Therefore, the interactive correlation value i_{pq} is assigned as the weight of the edge between them. Specifically, when we construct the weighted graph, we have already considered the geographical distances through the condition $d_{pq} \leq dist$; thus we only take the interactive correlations as the edge weights for simplicity. More complicated method fusing geographical distances and interactive correlations explicitly will also be welcomed. After constructing the weighted graph, more information from the most correlated POIs will be fused to accurately characterize the target POI.

We use negative sampling technique. The embeddings of all POIs are learned by minimizing the loss function defined as follows:

$$loss = - \sum_{l_p \in L} \sum_{l_q \in L_p^{Pos}} \log \left(\sigma(\mathbf{e}_p \mathbf{e}_q) - \frac{1}{r} \sum_{l_q^{neg} \in L_p^{Neg}} \sigma(\mathbf{e}_p \mathbf{e}_q^{neg}) \right) \tag{5}$$

where POI l_p is each POI in L , l_q is one of the correlation context L_p^{Pos} for l_p , L_p^{Neg} is a set of r negative POI samples for l_p . \mathbf{e}_p , \mathbf{e}_q and \mathbf{e}_q^{neg} are POI representations trained by Node2Vec model.

The POI correlation modeling algorithm is depicted in Algorithm 1.

Algorithm 1 POI correlation modeling algorithm

Input: the longitudes and latitudes of all POIs; the check-ins of all users in training set; the threshold of geographical distance $dist$; the threshold of interactive correlation θ .

Output: POI embedding matrix \mathbf{E} .

- 1: construct the geographical distance matrix \mathbf{D} by Eqs. (1) and (2);
- 2: construct the interactive correlation matrix \mathbf{I} by Eqs. (3) and (4);
- 3: **for** each POI l_p in POI set **do**
- 4: $L_p^{Pos} \leftarrow \emptyset$;
- 5: **for** each POI l_q in POI set **do**
- 6: find d_{pq} in \mathbf{D} and i_{pq} in \mathbf{I} ;
- 7: **if** $d_{pq} \leq dist$ and $i_{pq} \geq \theta$ **then**
- 8: add l_q into L_p^{Pos} ;
- 9: **end if**
- 10: **end for**
- 11: **end for**
- 12: construct a weighted graph using all POIs and their correlation contexts;
- 13: exploit Node2Vec to learn representations of all POIs by Eq. (5);
- 14: return POI embedding matrix \mathbf{E} .

4.2 Long- and short-term preferences modeling layer

So far, we have obtained POI embedding matrix $\mathbf{E} \in \mathbb{R}^{|L| \times d}$ which composes of embeddings of all POIs learned via Node2Vec training. Next, taking the POI embedding matrix as input, we retrieve the embedding representations of check-in records to capture user's preferences.

Previous studies have shown that it is crucial to model both long- and short-term preferences of users in the task of next POI recommendation. Nevertheless, they failed to select the most valuable check-ins dynamically to get user's temporal preferences. To this end, for history trajectories $S_{his} = \{S_1, S_2, \dots, S_{|n-1}|\}$ and current trajectory S_n , we design position-aware attention units to capture preference representation of each trajectory. Moreover, an attention network is developed to adaptively select important trajectory representations that conform to user's short-term demand for long-term preference modeling.

4.2.1 Preference representation of each trajectory

In this paper, user's long check-in sequence is divided into multiple trajectories based on the threshold of time intervals. This means that user's behaviors and preferences in each

trajectory are homogeneous and can change immediately across different trajectories. Therefore, trajectories are relatively independent from each other, and we will model different trajectories, respectively. Note that the existing literature has demonstrated that extracting preferences from multiple sessions or trajectories would model user’s sequential behaviors more effectively [45, 46]. Next, we employ position-aware attention unit to extract user preferences from each trajectory.

Taking the current trajectory S_n as example, the impact of check-ins in the trajectory increases along with time. In other words, POIs on different positions have different contributions to the trajectory representation. Therefore, we define a position embedding matrix $\mathbf{P} = [\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_m] \in \mathbb{R}^{m \times d}$ to indicate the specific position of each POI in the trajectory, where m is the maximum length of trajectories in the dataset and d is the dimensionality of position vectors. Inspired by GCE-GNN [47], the last POI can be regarded as user’s current preference, meanwhile, the distance between a POI and the last POI in the sequence contains more meaningful information. For example, given two trajectories $S' = \{l_1, l_2, l_4\}$ and $S'' = \{l_1, l_2, l_3, l_7, l_8\}$, the distance between l_2 and the last check-in POI is smaller in S' than that in S'' , which means that l_2 is more representative of user’s current preference in S' . Therefore, POI l_2 is more important for recommendation in S' than in S'' . Given that, reversed position embeddings are assigned to check-ins in the trajectory. In addition, we map the representation of the last POI to a query vector and map the representation of each POI in the trajectory to the key and value, which is an effective way to remove noise data. Similarly, the preference representations of history trajectories are also captured by the position-aware attention units. Following the design of LSTPM [21], the parameters of attention mechanism in all history trajectories are shared with each other but not shared with that in the current trajectory.

The attention weight of the p -th POI l_p^h in trajectory S_h is defined as follows:

$$\alpha_p^h = \mathbf{q}^T \sigma \left(\mathbf{W}_1 \mathbf{e}_p^h + \mathbf{W}_2 \mathbf{e}_{|S_h|}^h + \mathbf{W}_3 \mathbf{p}_{|S_h|-p+1} + \mathbf{b}_1 \right) \tag{6}$$

where \mathbf{e}_p^h and $\mathbf{e}_{|S_h|}^h$ are, respectively, representations of l_p^h and the last POI $l_{|S_h|}^h$, $\mathbf{p}_{|S_h|-p+1}$ is the reversed position embedding for l_p^h , $\sigma(\cdot)$ is the sigmoid activation function. $\mathbf{W}_1, \mathbf{W}_2, \mathbf{W}_3 \in \mathbb{R}^{d \times d}$ and $\mathbf{q}, \mathbf{b}_1 \in \mathbb{R}^d$ are trainable parameters.

The importance weight α_p^h of POI l_p^h is then normalized by the softmax function:

$$\beta_p^h = \frac{\exp(\alpha_p^h)}{\sum_{k=1}^{|S_h|} \exp(\alpha_k^h)} \tag{7}$$

Finally, the preference representation of S_h is obtained from the weighted sum of representations of all POIs in the trajectory:

$$\mathbf{s}_h = \sum_{p=1}^{|S_h|} \beta_p^h \mathbf{e}_p^h \tag{8}$$

4.2.2 Long- and short-term preferences representation

User’s short-term preference is extracted from the current trajectory S_n , i.e., $\mathbf{s}_{\text{Short}} = \mathbf{s}_n$. Accordingly, user’s long-term preference is extracted from set of all history trajectories $S_{\text{his}} = \{S_1, S_2, \dots, S_{n-1}\}$. After 4.2.1, preference representations of all history trajectories

have been obtained, *i.e.*, $\mathbf{s}_{\text{his}} = \{\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_{n-1}\}$. We suppose that trajectories that are highly relevant to the current trajectory play important roles in deciding user’s long-term preference [48]. It will inevitably introduce noise that all history trajectories are given the same importance. Therefore, an attention network is introduced to adaptively give relevant history trajectories more weights.

Specifically, we calculate the affinity scores between history trajectories and the current trajectory:

$$\gamma_h = \frac{\exp(\mathbf{s}_{\text{Short}}^T \mathbf{s}_h)}{\sum_{j=1}^{n-1} \exp(\mathbf{s}_{\text{Short}}^T \mathbf{s}_j)} \tag{9}$$

We transform \mathbf{s}_h through a feedforward neural network to get the final trajectory representation. Then, we multiply the final representation by the trajectory’s corresponding weight and sum all multiplication results to obtain user’s long-term preference:

$$\mathbf{s}_{\text{Long}} = \sum_{h=1}^{n-1} \gamma_h (\mathbf{W}_4 \mathbf{s}_h + \mathbf{b}_2) \tag{10}$$

where $\mathbf{W}_4 \in \mathbb{R}^{d \times d}$ and $\mathbf{b}_2 \in \mathbb{R}^d$ are trainable parameters.

4.3 Prediction layer

In this part, we obtain user’s preference representation by concatenating short-term preference $\mathbf{s}_{\text{Short}}$ and long-term preference \mathbf{s}_{Long} together. Then, we calculate user’s visiting possibility distribution $\hat{\mathbf{y}}$ over all $|L|$ POIs:

$$\hat{\mathbf{y}} = \text{softmax}(\mathbf{W}_u [\mathbf{s}_{\text{Long}}; \mathbf{s}_{\text{Short}}] + \mathbf{b}_u) \tag{11}$$

where $[\cdot; \cdot]$ is the concatenation operation, $\mathbf{W}_u \in \mathbb{R}^{|L| \times 2d}$ and $\mathbf{b}_u \in \mathbb{R}^{|L|}$ are trainable parameters. Let $\hat{\mathbf{y}}^{\text{true}} \in \hat{\mathbf{y}}$ denote the visiting possibility of the ground-truth POI, then log likelihood function is used to measure the loss of model training:

$$L = - \sum_{i=1}^N \log(\hat{\mathbf{y}}_i^{\text{true}}) \tag{12}$$

where N represents the total number of training samples and $\hat{\mathbf{y}}_i^{\text{true}}$ is the possibility of the ground-truth POI generated by our model in the i -th training sample.

To recommend next POI for user u , we take user’s all check-in trajectories as input and generate visiting possibility distribution $\hat{\mathbf{y}}$. POIs that have k largest probability values in $\hat{\mathbf{y}}$ are selected as Top- k recommendation list.

4.4 Model size analysis

Firstly, in the POI correlation modeling layer, the POI embedding matrix has the parameter of size $|L| \times d$. Secondly, in the long- and short-term preferences modeling layer, the position embedding matrix has the parameter of size $m \times d$. In each trajectory, the weight matrices $\mathbf{W}_1, \mathbf{W}_2$ and \mathbf{W}_3 for the attention mechanism are of the same size $d \times d$, and the parameters of \mathbf{q} and \mathbf{b}_1 are of size d . Since MPGI has two sets of trajectory parameters (one for history trajectories and one for the current trajectory), the total parameters of position-aware attention

units are $md + 2(3d^2 + 2d)$. Besides, as for the attention network, parameter is $d^2 + d$. Finally, in the prediction layer, \mathbf{W}_u of size $|L| \times 2d$ and \mathbf{b}_u of size $|L|$ are required for classification. Therefore, the total model size approximates $(3|L| + m + 5)d + 7d^2 + |L|$. Considering that d is small number generally not greater than 256, and m is usually less than 10, the proposed model is fairly light. As will be verified in the following experiments, 160 thousand check-ins in NYC are enough to learn the model with fairly good performance.

5 Experiments

We conduct extensive experiments on two real-world datasets, to demonstrate the effectiveness of the proposed MPGI model. In this section, we first introduce the experimental setup, then report the experimental results and corresponding analysis.

5.1 Experimental setup

5.1.1 Datasets

We evaluate our model on two real-world datasets: the Foursquare check-ins in New York and Tokyo [49] from April 12, 2012, to February 16, 2013. They are, respectively, denoted as NYC and TKY in the following contents. Each original check-in record in the two datasets consists of user ID, POI ID, POI category ID, POI category name, longitude and latitude of the POI and check-in timestamp.

For both datasets, we remove inactive users that have less than 10 check-ins and unpopular POIs visited less than 10 times. Next, for each user, we split the user's long check-in sequence into multiple trajectories according to the time intervals between consecutive check-ins. In detail, if the time interval exceeds 48 h, two consecutive POIs are divided into different trajectories. Then, we eliminate trajectories that have less than 3 POIs and remove users who have less than 5 trajectories. Thus, the statistics of both datasets after pre-processing are summarized as Table 2.

In Table 2, the sparsity of the datasets is calculated by:

$$\text{sparsity} = 1 - \frac{\#\text{check-ins}}{\#\text{users} \cdot \#\text{POIs}} \quad (13)$$

where $\#\text{check-ins}$, $\#\text{users}$ and $\#\text{POIs}$ represent the total number of check-ins, users and POIs, respectively.

Similar to most studies, the first 80% of each user's trajectories are used as the training set to train the model, and the last 20% are used as the test set to evaluate the model's performance.

Table 2 Statistics of datasets after pre-processing

Datasets	#Users	#POIs	#Check-ins	#Trajectories	Sparsity (%)
NYC	1074	15,514	159,676	22,439	99.04
TKY	2285	23,441	450,976	62,039	99.16

5.1.2 Baselines

We compare MPGI with the following state-of-the-art next POI recommendation models.

LSTM [50]: LSTM is known as a variant of RNN. It solves the problem of vanishing gradient and exploding gradient, thus gains widespread applications in sequential models. Here, we leverage LSTM as a baseline model for next POI recommendation.

TMCA [19]: TMCA embeds spatiotemporal transitions as well as POI categories. It further combines these embeddings with POI representations for next POI recommendation. Here, we remove the POI category because none of other models incorporates it. Especially, TMCA leverages only several check-ins to capture user's short-term preference for recommendation.

DeepMove [14]: DeepMove models both of user's long-term and short-term preferences for next POI recommendation. In particular, it learns short-term preference using RNN and aggregates long-term preference from history activities using the attention mechanism.

STAN [51]: STAN is one of the state-of-the-art models. To capture complex relationships between non-consecutive check-ins in the entire check-in sequence, STAN fuses spatiotemporal transition matrices into self-attention and further refines representations of visited POIs.

LSTPM [21]: LSTPM is currently one of the state-of-the-art methods, learning both of the short-term and long-term preferences. Specifically, LSTPM takes spatiotemporal relationships between history check-ins and the latest check-in into consideration and then fuses these relationships together.

5.1.3 Evaluation criterion

To evaluate recommendation performance of different models, we adopt two widely used metrics: Recall@ k and Normalized Discounted Cumulative Gain (NDCG@ k). Here, k represents the number of recommended POIs. The two metrics measure the performance of recommendation models from different perspectives. Recall@ k denotes whether the truly visited POI appears in the recommendation list. NDCG@ k is a position-based ranking metric. The value of NDCG@ k would be larger if the truly visited POI appears on a higher position of recommendation list. In this paper, we report Recall@ k and NDCG@ k with $k \in \{1, 5, 10, 20\}$.

5.1.4 Parameter settings

The optimal parameters for all models are achieved by experimental analysis or adopting default settings in original papers. As for our model, MPGI is implemented in python with PyTorch deep learning library. On both datasets, the embedding size of POIs d is searched in the range of $\{32, 64, 128, 256\}$, and the optimal value is finally determined as $d = 128$. The optimal setting for the threshold of interactive correlations θ is searched in $\{0.01, 0.03, 0.05, 0.07, 0.09, 0.11\}$, and the best setting on both datasets is $\theta = 0.05$. The optimal threshold of geographical distances on both datasets is $dist = 5$.

The best settings for hyperparameters in Node2Vec model are as follows: the walk length l and the number of walks r are set as $l = 15, r = 20$; the return parameter p and in-out parameter q are set as $p = 0.2, q = 2$; the context size k , the number of iterations $iter$ and the number of negative samplings num are, respectively, set as $k = 2, iter = 30, num = 5$. As for the learning rate lr of our MPGI model, the initial value is $lr = 0.001$ and becomes 0.1 times after every 10 epochs. The detailed analysis of representative parameter settings will be introduced in 5.2.3.

5.2 Results and discussions

5.2.1 Methods for comparison

The results of different models for next POI recommendation on both of the NYC and TKY datasets are illustrated in Table 3 and Fig. 3.

Firstly, the proposed MPGI model significantly outperforms all baselines in terms of Recall and NDCG on both datasets. As for the NYC dataset, MPGI achieves the optimal performance on all metrics except for Recall@20. Compared with LSTPM which has the best performance among baseline models, MPGI has an improvement of 6.3%, 4.3% and 4.3% on Recall@1, Recall@5 and Recall@10, respectively. At the same time, MPGI has an improvement of 6.3%, 5.1%, 5.0% and 5.9% on NDCG@1, NDCG@5, NDCG@10 and NDCG@20, respectively. As for the TKY dataset, MPGI consistently outperforms all the baselines in terms of all metrics, where the overall improvement over LSTPM achieves 6.8% on average. Moreover, although STAN shows advantage in terms of Recall@20 on the NYC dataset, it gains poor performance in terms of other metrics. In addition, the metrics we normally focused on for recommendation are Recall@ k and NDCG@ k with $k \in \{1, 5, 10\}$. Therefore, the experimental results consistently demonstrate the superiority of the proposed MPGI model.

Secondly, we observe that TMCA consistently outperforms LSTM, which implies that spatiotemporal transitions are important for next POI recommendation. Furthermore, DeepMove and LSTPM, both of which simultaneously capture long-term and short-term preferences, achieve higher prediction accuracy than TMCA that only captures user's short-term preference. This phenomenon highlights the necessity of modeling user's long-term preference from history check-ins. Moreover, although both of LSTPM and DeepMove try to simultaneously capture long- and short-term preferences, LSTPM outperforms DeepMove. The reason is that DeepMove fails to make full use of important spatiotemporal context. On the contrary, LSTPM not only captures relationships between history check-ins and user's current status, but also applies a geo-dilated RNN to capture geographical connections between POIs in the current trajectory.

Besides, the performance improvement of MPGI over LSTPM is attributed to the POI correlation modeling layer. LSTPM captures the spatial impacts by incorporating the geographical distances into attention weights of history check-ins. However, apart from geographical distances, MPGI captures interactive correlations to thoroughly model the complex correlations between POIs, and further learns high-quality POI embeddings, thus achieves higher recommendation performance.

5.2.2 Ablation analysis

To verify the effects of key components in MPGI, we implement six degraded versions of MPGI variants by eliminating one component each time in the following ablation study. Here, we divide these ablation versions into two groups for discussions.

(1) Ablation analysis of three components of MPGI

In this part, the following variants of the proposed MPGI are constructed to validate the effects of both the long- and short-term preferences of users as well as the POI correlation modeling:

MPGI-WL: the first degraded version by removing the long-term preference modeling part.

Table 3 Performance comparison on NYC and TKY datasets

Dataset	Methods	Recall@1	Recall@5	Recall@10	Recall@20	NDCG@1	NDCG@5	NDCG@10	NDCG@20
NYC	LSTM	0.1271	0.2954	0.3702	0.4284	0.1271	0.2152	0.2394	0.2542
	TMCA	0.1284	0.3062	0.3835	0.4498	0.1284	0.2211	0.2462	0.2630
	DeepMove	0.1829	0.3866	0.4672	0.5341	0.1829	0.2903	0.3165	0.3335
	STAN	0.0938	0.2969	0.5078	0.6406	0.0938	0.1990	0.2684	0.3020
	LSTPM	0.1832	0.4151	0.5125	0.5877	0.1832	0.3041	0.3357	0.3548
	MPGI	0.1948	0.4331	0.5347	0.6269	0.1948	0.3195	0.3525	0.3758
TKY	LSTM	0.1435	0.3227	0.3985	0.4679	0.1435	0.2377	0.2623	0.2798
	TMCA	0.1551	0.3343	0.4070	0.4748	0.1551	0.2494	0.2730	0.2901
	DeepMove	0.1868	0.3671	0.4374	0.4957	0.1868	0.2818	0.3046	0.3194
	STAN	0.0825	0.2887	0.4021	0.5567	0.0825	0.1847	0.2205	0.2590
	LSTPM	0.1876	0.4103	0.4977	0.5718	0.1876	0.3049	0.3333	0.3521
	MPGI	0.2019	0.4367	0.5275	0.6098	0.2019	0.3260	0.3554	0.3763

Best results for each dataset are presented in bold

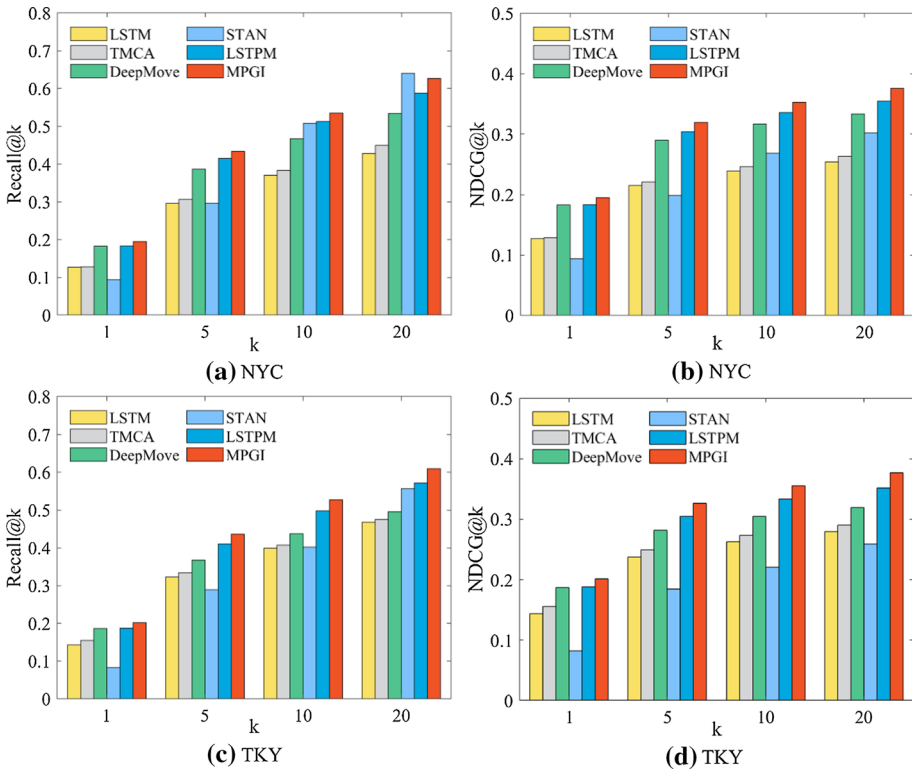


Fig. 3 Performance comparisons on NYC and TKY datasets

MPGI-WS: the second version by removing the short-term preference modeling part.

MPGI-WC: the third version by removing the POI correlation modeling layer. Instead, POI embeddings are randomly initialized similar to most studies.

The experimental results are shown in Fig. 4. We observe that all the variants have poorer performance than the complete MPGI model on both datasets. Among three ablation versions, MPGI-WL performs worst among all models on both datasets. For instance, on the NYC dataset, MPGI outperforms MPGI-WL by 27.6% and 26.7% on Recall@5 and NDCG@5. On the TKY dataset, MPGI outperforms MPGI-WL by 40.6% and 45.2% on Recall@5 and NDCG@5. This observation indicates that user’s long-term preference plays a significant role in recommendations. Therefore, it is important to select the valuable information in user’s long-term check-in trajectories for user preferences modeling. Moreover, except MPGI-WL, MPGI-WS also does not perform well on both datasets, indicating that capturing user’s short-term preference is a necessity in next POI recommendation. In fact, many models such as TMCA only capture user’s short-term preference and could achieve decent performance. Furthermore, the performance of MPGI-WC also suffers a significant decrease, especially on the NYC dataset. It is worth noting that the POI embeddings in MPGI-WC are randomly initialized. So, explicit POI correlation modeling is indispensable for obtaining accurate POI representations and further boost recommendations.

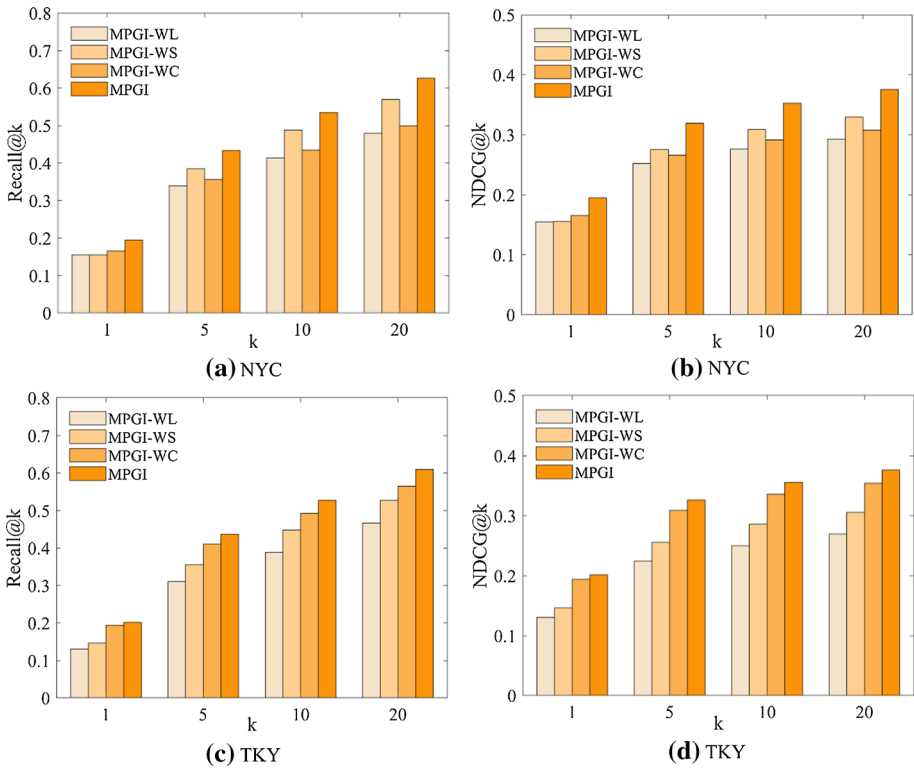


Fig. 4 Performance comparison among four versions of MPGI variants

(2) Ablation analysis of three key technologies

So far, the effects of subtle components of MPGI, such as the position-aware attention unit, have not been fully explored. To this end, we further remove three key technologies to prove their effectiveness. The ablation versions are described as follows:

MPGI-WP: MPGI that removes position-aware attention units. Instead, the mean pooling is used to capture the preference representation of each trajectory.

MPGI-WA: MPGI that removes the attention network. Instead, mean pooling is applied to all history trajectories to get user long-term preference.

MPGI-WW: In POI correlation modeling layer, we construct an unweighted graph where the different weights of edges are removed.

As shown in Table 4, MPGI has the best performance, demonstrating that all the key components are effective in boosting recommendation. On the NYC dataset, the attention network contributes most to the performance improvement. This verifies that it is inappropriate to assign the same weights to different history trajectories. Therefore, an attention network is needed to adaptively aggregate history trajectories' representations for long-term preference modeling. On the TKY dataset, however, MPGI-WP gains the most performance drop. This clearly demonstrates that position-aware attention units are able to adaptively select valuable information and avoid noise for effective trajectory representations. Lastly, MPGI-WW also performs worse than MPGI. By assigning different weights to edges, important signals from more correlated POIs are incorporated into the central POI to refine its representation.

Table 4 Validation of three key technologies in MPGI

Methods	NYC		TKY	
	Recall@5	NDCG@5	Recall@5	NDCG@5
MPGI-WP	0.4214	0.3026	0.4132	0.2975
MPGI-WA	0.4051	0.2898	0.4208	0.3125
MPGI-WW	0.4285	0.3166	0.4359	0.3256
MPGI	0.4331	0.3195	0.4367	0.3260

Bold values indicate experimental results in best score

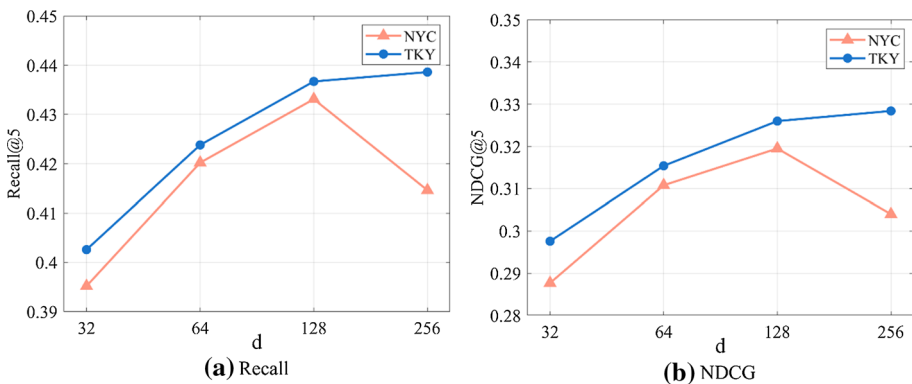
5.2.3 Sensitivity analysis of parameters

Now we take Recall@5 and NDCG@5 as example to investigate how representative hyperparameters affect the performance of MPGI, including the embedding size of POIs d , the threshold of interactive correlations θ , and the threshold of geographical distances $dist$. In the following experiments, a hyperparameter is tested while the settings for the rest parameters are optimal as introduced in 5.1.4.

(1) Effect of embedding size d

We evaluate how the embedding size of POIs d impacts the recommendation performance by varying the value of d in $\{32, 64, 128, 256\}$. As shown in Fig. 5, on NYC dataset, the performance of MPGI first climbs up and yields the best performance with $d = 128$, while significantly drops with further increase in d ($d = 256$). On TKY dataset, the performance first grows dramatically while then improves very slowly as the increase in d ($d = 256$). The results demonstrate that a larger value of d is able to encode more valuable information to sufficiently capture the characteristics of POIs. However, oversized embeddings are too complicated to describe the POIs, and even over-represent POIs. In this work, we choose the embedding size $d = 128$ for both datasets.

(2) Effect of the threshold of interactive correlations θ

**Fig. 5** Parameter sensitivity of d on Recall@5 and NDCG@5

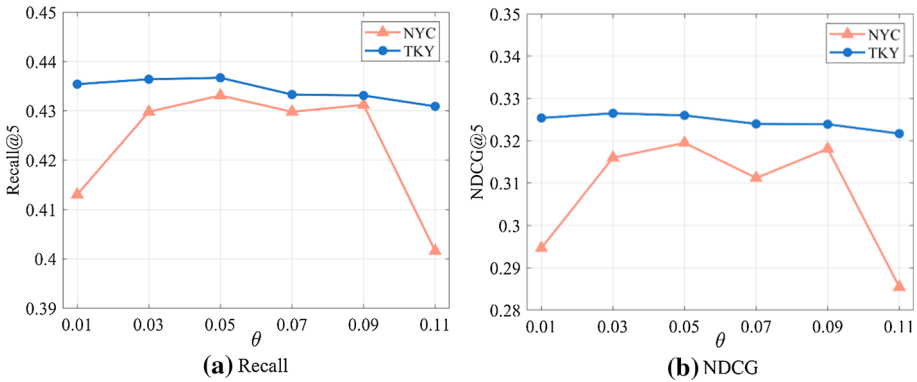


Fig. 6 Parameter sensitivity of θ on Recall@5 and NDCG@5

In POI correlation modeling layer, the threshold of interactive correlations θ plays an important role in selecting the correlation context for the target POI. Therefore, we vary values of θ in the range of $\{0.01, 0.03, 0.05, 0.07, 0.09, 0.11\}$ to study its impact on recommendation performance. As shown in Fig. 6, on both datasets, the performance improves with the increase in θ from 0.01 to 0.05, since a larger value of θ introduces more useful information of highly correlated POIs to enrich the representation of the target POI. The optimal performance of MPGI is obtained when $\theta = 0.05$ on both datasets. However, the performance drops with further increase in θ , as too large value of θ will inevitably propagate the information from irrelevant POIs into the target POI, thus introducing noise and misleading the inference of POI features.

(3) Effect of the threshold of geographical distances $dist$

The threshold of geographical distances $dist$ is another factor affecting the selection of highly correlated POIs. To this end, we examine the influence of $dist$ by varying $dist$ in the range of $\{1, 5, 9, 13, 17\}$. From the results shown in Fig. 7, we note that both Recall@5 and NDCG@5 increase with larger $dist$, while significantly drop on NYC and slightly drop on TKY after $dist$ reaches the optimal setting $dist = 5$. Similar to θ , the value of $dist$ is supposed to be cautiously set to facilitate the effective construction of POI correlation contexts.

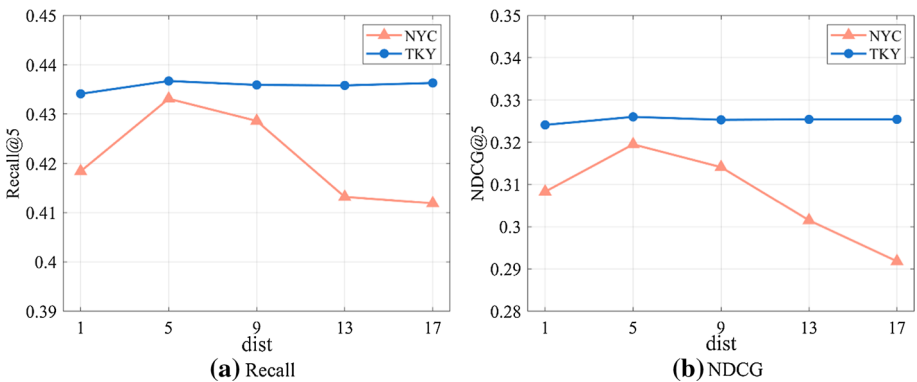


Fig. 7 Parameter sensitivity of $dist$ on Recall@5 and NDCG@5

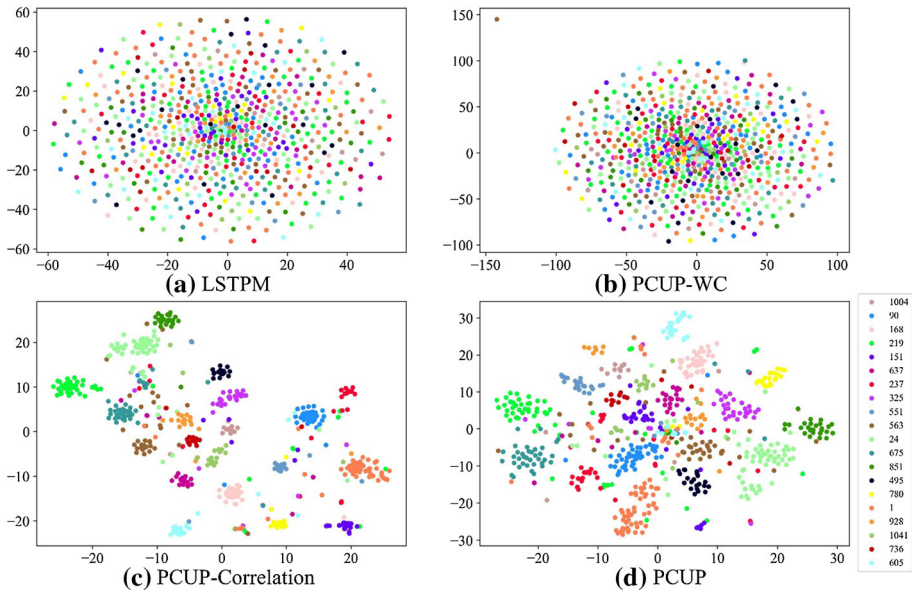


Fig. 8 Visualization of POI embeddings trained by different models

5.2.4 Visualization of POI embeddings

Next, we examine whether the POI correlation modeling layer is able to learn effective POI representations. Taking NYC dataset as example, we randomly select 20 users' visited POIs. Note that each POI is visited by one of 20 users; therefore, each POI is labeled by the corresponding user's ID. For these selected POIs, we use t-SNE method [52] to visualize their embeddings learned from four models, namely LSTPM, MPGI-WC, MPGI-Correlation and MPGI. Here, MPGI-Correlation represents our degraded model that only contains POI correlation modeling layer. Specifically, we map these POI embeddings as two-dimensional vectors, and visualize them as points with different colors. The visualization results are presented in Fig. 8.

As shown in Fig. 8a, for the LSTPM method, POIs visited by the same users are scattered randomly, which indicates that LSTPM fails to explicitly capture the complex correlations between POIs, thus achieving sub-optimal performance. Similarly, as for the MPGI-WC method, the distribution of POIs in Fig. 8b is irregular.

Instead, Fig. 8c verifies the effectiveness of POI correlation modeling: POIs visited by same users are close to each other, tending to form multiple clusters. The clustering phenomenon verifies that POI correlation modeling layer of the proposed model is able to capture correlations between POIs visited by same users. This is reasonable because users tend to visit POIs that have short geographical distances and high interactive correlations in real scenarios. Furthermore, POIs in Fig. 8d also exhibit discernible clustering, while the clustering situation is different from that in Fig. 8c. This is because that POI embeddings are further adjusted in the training process. In fact, the POI correlation modeling layer only captures the general features of POIs, which is irrelevant to the specific next POI recommendation task. Only in the following training process can the proposed model capture user's temporal preferences and characterize POIs more precisely.

6 Conclusions

In this paper, we propose a novel next POI recommendation framework named MPGI to capture the complex correlations between POIs and further model user dynamic preferences. Specifically, we explore the complex correlations, *i.e.*, geographical distances and interactive correlations between all of POI pairs, to obtain the correlation context for each POI. Then, we construct a weighted graph and apply Node2Vec to learn high-quality POI representations. Corresponding POI representations in user's trajectories are retrieved from the learned POI embeddings. Furthermore, position-aware attention units and attention network are designed to adaptively select the most valuable information for user's long- and short-term preferences modeling. Experimental results on two real-world datasets demonstrate that MPGI consistently outperforms all the baselines. Further, six ablation experiments and a visualization study demonstrate the effectiveness of the key components in our model.

For future work, we will enhance MPGI from the following aspects. First, we will employ graph neural networks to implement our idea in an end-to-end framework. Second, we will assign more effective edge weights on the constructed weighted graph. Finally, we consider as a promising direction the exploration of descriptions associated with POIs and users' comments.

Acknowledgements This work was supported by the National Natural Science Foundation of China under Grants Nos. 72271024, 71871019, 71471016.

Author contributions J.R. contributed to data curation, methodology, validation, writing—original draft. M.G. contributed to conceptualization, resources, investigation, writing—review and editing, supervision, and funding acquisition.

Declarations

Conflict of interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Halder S, Lim KH, Chan J, Zhang X (2022) Efficient itinerary recommendation via personalized POI selection and pruning. *Knowl Inf Syst* 64:963–993. <https://doi.org/10.1007/s10115-021-01648-3>
2. Zhang H, Gan M, Sun X (2021) Incorporating memory-based preferences and point-of-interest stickiness into recommendations in location-based social networks. *ISPRS Int J Geo-Inf* 10:36. <https://doi.org/10.3390/ijgi10010036>
3. Bahari Sojahrood Z, Taleai M (2021) A POI group recommendation method in location-based social networks based on user influence. *Expert Syst Appl* 171:114593. <https://doi.org/10.1016/j.eswa.2021.114593>
4. Christoforidis G, Kefalas P, Papadopoulos AN, Manolopoulos Y (2021) RELINE: point-of-interest recommendations using multiple network embeddings. *Knowl Inf Syst* 63:791–817. <https://doi.org/10.1007/s10115-020-01541-5>
5. Pang G, Wang X, Hao F et al (2020) Efficient point-of-interest recommendation with hierarchical attention mechanism. *Appl Soft Comput J* 96:106536. <https://doi.org/10.1016/j.asoc.2020.106536>
6. Zheng C, Tao D, Wang J, Cui L (2021) Memory augmented hierarchical attention network for next point-of-interest recommendation. *IEEE Trans Comput Soc Syst* 8:489–499. <https://doi.org/10.1109/TCSS.2020.3036661>
7. Chen M, Liu Y, Yu X (2014) NLPMM: a next location predictor with Markov modeling. *Springer Int Publ* 8444:186–197. https://doi.org/10.1007/978-3-319-06605-9_16

8. Cheng C, Yang H, Lyu MR, King I (2013) Where you like to go next: successive point-of-interest recommendation. In: Proceedings of the twenty-third international joint conference on artificial intelligence. pp 2605–2611
9. Liu T, Liao J, Wu Z et al (2020) Exploiting geographical-temporal awareness attention for next point-of-interest recommendation. *Neurocomputing* 400:227–237. <https://doi.org/10.1016/j.neucom.2019.12.122>
10. Zhou F, Dai Y, Gao Q et al (2021) Self-supervised human mobility learning for next location prediction and trajectory classification. *Knowl Based Syst* 228:107214. <https://doi.org/10.1016/j.knsys.2021.107214>
11. Wang H, Li P, Liu Y, Shao J (2021) Towards real-time demand-aware sequential POI recommendation. *Inf Sci (Ny)* 547:482–497. <https://doi.org/10.1016/j.ins.2020.08.088>
12. Huang L, Ma Y, Wang S, Liu Y (2021) An attention-based spatiotemporal LSTM network for Next POI recommendation. *IEEE Trans Serv Comput* 14:1585–1597. <https://doi.org/10.1109/TSC.2019.2918310>
13. Liu Q, Wu S, Wang L, Tan T (2016) Predicting the next location: a recurrent model with spatial and temporal contexts. *Proc Thirtieth AAAI Conf Artif Intell* 2016:194–200
14. Feng J, Li Y, Zhang C et al (2018) DeepMove: predicting human mobility with attentional recurrent networks. *Proc World Wide Web Conf* 2018:1459–1468
15. Zhang J, Liu X, Zhou X, Chu X (2021) Leveraging graph neural networks for point-of-interest recommendations. *Neurocomputing* 462:1–13. <https://doi.org/10.1016/j.neucom.2021.07.063>
16. Zhong T, Zhang S, Zhou F et al (2020) Hybrid graph convolutional networks with multi-head attention for location recommendation. *World Wide Web* 23:3125–3151. <https://doi.org/10.1007/s11280-020-00824-9>
17. Wang X, He X, Wang M, et al (2019) Neural graph collaborative filtering. In: Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval. pp 165–174
18. Sarwar B, Karypis G, Konstan J, Riedl J (2001) Item-based collaborative filtering recommendation Algorithms. In: Proceedings of the 10th international conference on world wide web (WWW'01). pp 285–295
19. Li R, Shen Y, Zhu Y (2018) Next Point-of-interest recommendation with temporal and multi-level context attention. In: 2018 IEEE international conference on data mining. IEEE, pp 1110–1115
20. Zhu G, Wang Y, Cao J et al (2021) Neural attentive travel package recommendation via exploiting long-term and short-term behaviors. *Knowl Based Syst* 211:106511. <https://doi.org/10.1016/j.knsys.2020.106511>
21. Sun K, Qian T, Chen T, et al (2020) Where to go next: modeling long- and short-term user preferences for point-of-interest recommendation. In: Proceedings of the thirty-fourth AAAI conference on artificial intelligence. pp 214–221
22. Ren X, Song M, Haihong E, Song J (2017) Context-aware probabilistic matrix factorization modeling for point-of-interest recommendation. *Neurocomputing* 241:38–55. <https://doi.org/10.1016/j.neucom.2017.02.005>
23. Yuan Q, Cong G, Ma Z, et al (2013) Time-aware point-of-interest recommendation. In: Proceedings of the 36th international ACM SIGIR conference on research and development in information retrieval. pp 363–372
24. Yao L, Sheng QZ, Qin Y, et al (2015) Context-aware point-of-interest recommendation using Tensor Factorization with social regularization. In: Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval. pp 1007–1010
25. Li X, Cong G, Li XL, et al (2015) Rank-geoFM: a ranking based geographical factorization method for point of interest recommendation. In: Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval. pp 433–442
26. Gan M, Gao L (2019) Discovering memory-based preferences for POI recommendation in location-based social networks. *ISPRS Int J Geo-Inf* 8:279. <https://doi.org/10.3390/ijgi8060279>
27. Davtalab M, Alesheikh AA (2021) A POI recommendation approach integrating social spatio-temporal information into probabilistic matrix factorization. *Knowl Inf Syst* 63:65–85. <https://doi.org/10.1007/s10115-020-01509-5>
28. Ma Y, Gan M (2020) Exploring multiple spatio-temporal information for point-of-interest recommendation. *Soft Comput* 24:18733–18747. <https://doi.org/10.1007/s00500-020-05107-z>
29. Liu Q, Mu L, Sugumaran V et al (2021) Pair-wise ranking based preference learning for points-of-interest recommendation. *Knowl Based Syst* 225:107069. <https://doi.org/10.1016/j.knsys.2021.107069>
30. Yin H, Wang W, Wang H et al (2017) Spatial-aware hierarchical collaborative deep learning for POI recommendation. *IEEE Trans Knowl Data Eng* 29:2537–2551. <https://doi.org/10.1109/TKDE.2017.2741484>
31. Liu Y, Yang Z, Li T, Wu D (2021) A novel POI recommendation model based on joint spatiotemporal effects and four-way interaction. *Appl Intell* 52:5310–5324. <https://doi.org/10.1007/s10489-021-02677-9>
32. Chen L, Cao J, Chen H et al (2021) Attentive multi-task learning for group itinerary recommendation. *Knowl Inf Syst* 63:1687–1716. <https://doi.org/10.1007/s10115-021-01567-3>

33. Feng S, Li X, Zeng Y, et al (2015) Personalized ranking metric embedding for next new POI recommendation. In: Proceedings of the twenty-fourth international joint conference on artificial intelligence. pp 2069–2075
34. Yu F, Cui L, Guo W et al (2020) A category-aware deep model for successive POI recommendation on sparse check-in data. *Proc World Wide Web Conf 2020*:1264–1274
35. Wu Y, Li K, Zhao G, Qian x (2022) Personalized long- and short-term preference learning for next POI recommendation. *IEEE Trans Knowl Data Eng* 34:1944–1957. <https://doi.org/10.1109/TKDE.2020.3002531>
36. Liu X, Yang Y, Xu Y et al (2022) Real-time POI recommendation via modeling long- and short-term user preferences. *Neurocomputing* 467:454–464. <https://doi.org/10.1016/j.neucom.2021.09.056>
37. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser IP (2017) Attention is all you need. In: Proceedings of the advances in neural information processing systems. pp 5998–6008
38. Zhou G, Mou N, Fan Y, et al (2019) Deep interest evolution network for click-through rate prediction. In: Proceedings of the thirty-third AAAI conference on artificial intelligence. pp 5941–5948
39. Wu S, Tang Y, Zhu Y, et al (2019) Session-based recommendation with graph neural networks. In: Proceedings of the thirty-third AAAI conference on artificial intelligence. pp 346–353
40. Liu L, Wang L, Lian T (2021) CaSe4SR: using category sequence graph to augment session-based recommendation. *Knowl Based Syst* 212:106558. <https://doi.org/10.1016/j.knosys.2020.106558>
41. Shi M, Shen D, Kou Y et al (2021) Attentional memory network with correlation-based embedding for time-aware POI recommendation. *Knowl Based Syst* 214:106747. <https://doi.org/10.1016/j.knosys.2021.106747>
42. Zheng J, Li Q, Liao J (2021) Heterogeneous type-specific entity representation learning for recommendations in e-commerce network. *Inf Process Manag* 58:102629. <https://doi.org/10.1016/j.ipm.2021.102629>
43. Ni J, Huang Z, Yu C et al (2021) Comparative convolutional dynamic multi-attention recommendation model. *IEEE Trans Neural Netw Learn Syst*. <https://doi.org/10.1109/TNNLS.2021.3053245>
44. Ma Y, Mao J, Ba Z, Li G (2020) Location recommendation by combining geographical, categorical, and social preferences with location popularity. *Inf Process Manag* 57:102251. <https://doi.org/10.1016/j.ipm.2020.102251>
45. Feng Y, Lv F, Shen W et al (2019) Deep session interest network for click-through rate prediction. *IJCAI Int J Conf Artif Intell*. <https://doi.org/10.24963/ijcai.2019/319>
46. Thaipisutikul T, Shih TK (2021) A novel context-aware recommender system based on a deep sequential learning approach (CRoS). *Neural Comput Appl* 33:11067–11090. <https://doi.org/10.1007/s00521-020-05640-w>
47. Xu H, Huang C, Xu Y, et al (2020) Global context enhanced social recommendation with hierarchical graph neural networks. In: Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval pp 701–710
48. Zhong J, Ma C, Zhou J, Wang W (2020) Pdpnn: modeling user personal dynamic preference for next point-of-interest recommendation. *Int Conf Comput Sci* 2020:45–57
49. Yang D, Zhang D, Zheng VW, Yu Z (2015) Modeling user activity preference by leveraging user spatial temporal characteristics in LBSNs. *IEEE Trans Syst Man Cybern Syst* 45:129–142. <https://doi.org/10.1109/TSMC.2014.2327053>
50. Schuster M, Paliwal KK (1997) Bidirectional recurrent neural networks. *IEEE Trans Signal Process* 45:2673–2681. <https://doi.org/10.1109/78.650093>
51. Luo Y, Liu Q, Liu Z (2021) STAN: spatio-temporal attention network for next location recommendation. *Proc of the World Wide Web Conf 2021*:2177–2185
52. Van Der Maaten L, Hinton G (2008) Visualizing data using t-SNE. *J Mach Learn Res* 9:2579–2605

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.



Jieyu Ren received the bachelor's degree from University of Science and Technology Beijing, China, in 2021. She is currently pursuing the Ph.D. degree with the Department of Management Science and Engineering, School of Economics and Management, University of Science and Technology Beijing, China. Her research interests include recommender systems, social network analysis, and deep learning.



Mingxin Gan received the B.A. degree in automation from Tsinghua University, China, in 2001, and the Ph.D. degree in management science and engineering from the Beijing Institute of Technology, China, in 2006. She is currently a Professor with the Department of Management Science and Engineering, School of Economics and Management, University of Science and Technology Beijing, China. She has published more than 50 papers. Her papers have appeared in Knowledge-based Systems, Decision Support Systems, Information Systems Frontiers, Expert Systems with Applications, Bioinformatics, BMC Systems Biology, Future Generation Computer Systems, PLOS ONE, World Wide Web Journal, and so on. Her current research interests include intelligent recommender systems, social network analysis, deep learning, urban computing, and wise medical.