PG$^2$Net: Personalized and Group Preferences Guided Network for Next Place Prediction

Bin Wang, Huifeng Li, Weipeng Wang, Menghan Wang, Yaohui Jin, and Yanyan Xu

Abstract—Predicting the next destination is a key in human mobility behavior modeling, which is significant in various fields, such as epidemic control, urban planning, traffic management and recommendation. To achieve this, one typical solution is designing modules based on RNN to capture their preferences to various locations. Although these RNN-based methods can effectively learn individual’s hidden personalized preferences to her visited places, the interactions among users can only be weakly learned through the representations of locations. Targeting this, we propose an end-to-end framework named personalized and group preference guided network (PG$^2$Net), considering the users’ preferences to various places at both individual and collective levels. Specifically, PG$^2$Net concatenates Bi-LSTM and attention mechanism to capture each user’s long-term mobility tendency. To learn population’s group preferences, we utilize spatial and temporal information of the visitations to construct a spatial-temporal dependency module. We adopt a graph embedding method to map users’ trajectory into a hidden space, capturing their sequential relation. In addition, we devise an auxiliary loss to learn the vectorial representation of her next location. Experimental results on two Foursquare check-in datasets and one mobile phone dataset indicate the advantages of our model compared to the state-of-the-art baselines. Source code is available at https://github.com/urbanmobility/PG2Net.

Index Terms—Next place prediction, human mobility, trajectory data, personalized and group preferences, attention mechanism.

I. INTRODUCTION

WITh the rapid development of information and communication technologies, users can share their locations almost anytime and anywhere to acquire the location-aware services, thus generating an abundant amount of trajectory data. Location-based social networks (LBSNs), such as Foursquare and Yelp, sell a huge amount of location data from millions of individuals [1], [2]. These vast amount of trajectory data provides an unprecedented opportunity to study human mobility behavior [3], [4]. With massive mobile phone data, González et al. studied temporal and spatial regularity in human trajectories [3]. They found that their radius of gyration clearly follow a power law distribution, indicating the sample reproducible patterns of human mobility. In a following study, Song et al. assessed the potential predictability of human mobility was 93% [5]. Due to the high predictability of human mobility, researchers attempted to model the mobility behavior of the population at urban scale and utilized it to tackle various urban challenges, such as traffic congestion mitigation [6], [7], [8], air pollution exposure assessment [9], electric vehicles charging behavior planning [10], epidemic control [11], [12].

Prediction of user’s next destination is the key in human mobility modeling, and has attracted more and more researchers’ attention [13]. The task is designed to predict the destination of each user’s next trip. Assuming that a user’s next destination is highly correlated with her most recently visited locations, Rendle et al. developed a personalized transition matrix based on the Markov chain to capture the impact of users’ most recently visited locations on the current mobility decision [14]. Researchers also notice that the users’ periodic behavior observed from long-term historical trajectory plays a critical role in the decision of next travel [15], [16]. For instance, in a user’s daily routine, she or he is accustomed to going to the library every weekend. Therefore, combining the influence of users’ long-term and short-term trajectory can benefit the prediction of next location [17], [18]. Feng et al. proposed DeepMove based on an attention mechanism to evaluate the influence of the user’s historical trajectory information on the current travel choice and provide personalized location recommendations for users [17]. Manotumruksa et al. proposed a deep recurrent collaborative filtering framework (DRCF), utilizing the location of users similar in the historical trajectory to predict the next location. More recently, researchers proposed a CNN-based method and a gated network under the RNN-based framework to extract users’ long-term and short-term preferences [19], [20].

Although the prediction accuracy has been gradually improved in the above literature, it is still significantly challenging to predict the mobility behavior of a large number of users in complex urban environments. On one hand, existing studies focused on modeling the mobility patterns at individual level. In these approaches, the collective pattern of the interaction between population and space is not clearly modeled. In the early studies, statistical physicists used historical
movement data to discover the collective pattern of humans. For example, Schneider et al. [21] found that more than 90% of residents’ mobile behavior conformed to one of 17 basic network modes (Motif). Alessandretti et al. [22] found that the number of places repeatedly visited by individuals is stable, revealing people’s stable social relations. These studies indicate that statistical physical characteristics are of great value for studying the movement trend of human group level. On the other hand, existing methods for next location prediction usually cascade the embedding of user ID with the latent vector of locations in the long-term historical and recent trajectories, so as to capture the user’s personalized preferences [23], [24], [25]. Users always have various preferences for different locations. However, these methods can not directly model such heterogeneity and dynamics of the preferences. Targeting this weakness, researchers introduced attention mechanism to learn the personalized preference of each user [26]. However, the users’ mobility patterns in space and time are not fully explored from their trajectory sequences in these attention-based approaches. Specifically, they mainly focus on exploiting the sequential patterns of users’ personalised preferences without considering the associated timestamps and geographic information which reflects the group mobility patterns. Therefore, how to reasonably use trajectory data to understand users’ personalized preferences is the main content of our research.

To this end, we propose a novel personalized and group preference guided network (PG$^2$Net) to tackle the above issues. The framework is devised to predict each user’s next place to visit via considering her preferences to various locations at both individual and group level. Fig. 1 illustrates the end-to-end procedure of our learning-based model. PG$^2$Net first learns users’ personalized characteristics and group characteristics from the training data, and integrates the urban information into the prediction module. In the testing phase, the well-trained model is used to predict the users’ next location given their current trajectories. Specifically, we consider several key factors affecting group movement behavior, including geographical environment, distance attenuation and individual spatial activity characteristics. Among them, the geographical environment determines the spatial distribution of potential visiting locations that users can reach; the distance attenuation indicates the relationship between user’s visit frequency and distance; the individual spatial activity characteristics reflect people’s potential life habits. Considering the above factors, we design a module that combines prior statistical information and recurrent neural network, namely, the dynamic spatial-temporal dependency module, which can learn the influence of group preferences on users’ mobility patterns. For learning personalized preferences of each user, we propose a module based on bidirectional long short-term memory networks (Bi-LSTM) [27] and an attention mechanism to capture the user’s dynamic preference. Besides, a novel graph embedding method is proposed to represent each location and its category, which can efficiently learn the sequential relation between visited places. It should be mentioned that most previous studies did not take category information into account. In our model, we use category information to construct a statistical model to study the characteristics of users’ group activities. Finally, we propose a novel auxiliary loss function to learn the vectorial representation of the target location to improve the prediction accuracy. The framework of our proposed approach is shown in Fig. 2.

Our contributions are summarized as follows:

1. We propose a novel PG$^2$Net framework to predict the next place to visit via learning the user’s personalized preferences and group preferences. PG$^2$Net consists of two parts: (i) dynamic spatio-temporal dependency module that leverages temporal and spatial information to model the group preferences of users; (ii) personalized preference module which uses Bi-LSTM and attention mechanism to capture users’ personalized preferences.

2. We introduce an auxiliary loss function to enhance the similarity between the predicted and actual representation of the next position, so as to find a more efficient position embedding hypothesis space. The loss function is designed to increase the probability that the
actual location occurs and decrease the distance between the predicted location and the actual location in the hypothesis space.

(3) To model the people’s collective preferences to variant locations during long-term and short-term periods, we design a group preference module in $PG^2$Net, which captures the long-term and short-term preferences via integrating the prior statistical knowledge of people’s mobility behavior with the encoding of historical and recent trajectories respectively.

(4) We propose a framework that can characterize the user’s activity preference at different time based on the observation that users also have different activity preferences at different time on the same day. To our best knowledge, it is the first time to use graphical representation to define this problem.

(5) We conduct extensive experiments on three real-world datasets in different countries, including two public check-in datasets and a mobile phone dataset, a.k.a. call detail records (CDRs). Experiments show that our model achieves significant improvements over the state-of-the-art methods.

II. RELATED WORK

The purpose of the next location prediction task is to recommend a set of ranked locations for users, where the highest-ranking is the predicted next location. At present, there are two types of methods for this task – conventional (non-deep learning) machine learning-based methods and deep learning-based methods [17].

A. Conventional Machine Learning-Based Methods

The trajectory prediction task needs to mine the users’ past trajectory information to predict where the users will go. Typically, for this sequence prediction task, we can use a method based on the Markov model and its variants to make predictions. The Markov-based method is mainly to calculate the transition matrix of the location. According to transition matrix, we can predict where the user will go next. For example, Rendle et al. [14] proposed the factorized personalized Markov chains (FPMC), which is a method applying personalized Markov and matrix factorization to learn users’ transition matrix and overall preference. Cheng et al. [28] proposed a method FPMC-LR to obtain users’ personalized preference and realize the prediction of the next location. Excepted for Markov-based model, Alhasoun et al. utilized information from similar strangers for next place Prediction [13]. In this model, they proposed several human mobility similarity metrics used to identify other users with similar mobility characteristics, and proposed dynamic Bayesian network (DBN) model that incorporates the mobility patterns of similar strangers towards better predicting next locations.

B. Deep Learning-Based Methods

In recent years, deep learning has developed rapidly. Especially, RNN-based methods have attracted increasing attention and been successfully applied to many sequential problems, such as natural language processing, Vehicle Trajectory Prediction, voice recognition, image annotation, and machine translation. So far, researchers have tried to use the RNN-based method for travel information prediction and achieve inspiring results [17], [20], [29], [30], [31], [32]. For example, Liu et al. [29] proposed the ST-RNN network, which takes users’ adjacent time and space information as the input of the RNN module to capture the spatial-temporal influence. Yao et al. [33] established a recursive model of semantic perception (SERM), which learned the embedding of multiple factors (user, location, time) and captured the time and spatial transition regularity of semantic perception.

Recently, the attention mechanism has been widely used in various fields. Feng et al. [17] proposed a model named DeepMove based on attention mechanism and recurrent neural network to capture human mobility. In DeepMove, a multi-module embedding method is adopted to convert sparse features (user, location, time) into a dense representation, and then the historical attention module is used to obtain the most relevant historical trajectory information. However, this method fails to capture the user’s dynamic personalized preferences and barely considers the temporal and spatial dependence of the actual users. Gao et al. [34] proposed a variation-attention-based next location prediction model to overcome the sparsity problem of trajectory data. Long et al. [35] designed four features to represent travel regularity and preference which is incorporated in context and proposed a deep neural network to jointly capture regularity and preference via explicitly retrieving the context. Wu et al. [26] proposed a model named PLSPL to learn the specific preference for each user. This method attempts to model category information to predict the next location, but the method does not consider the specific regularities of human mobility and the sequence interaction influences. LSTM model proposed by [20] uses context-aware non-local network and geo-dilated RNN to obtain users’ long-term and short-term preference respectively. Meanwhile, researchers use attention mechanism to capture spatio-temporal information in trajectory prediction. Chen et al. [32] propose a novel spatial-temporal dynamic attention network for vehicle trajectory prediction, which can comprehensively capture temporal and social patterns in a hierarchical manner. Luo et al. [36] proposed STAN, which is a spatio-temporal bi-layer attention model focusing on learning regularities of non-adjacent locations and non-consecutive check-ins. DGCN [37] leveraged the disentangled representations to explicitly model different aspects and corresponding influence for representing a POI more precisely. GETNext [38] incorporated the global transition patterns, user’s general preference, spatio-temporal context, and time-aware category embeddings together into a transformer model to make the next POI recommendation.

Different from the above mentioned methods, our proposed $PG^2$Net considers the users’ preferences to various places at both individual and collective levels, which concatenates Bi-LSTM and attention mechanism to capture each user’s personalized long-term mobility tendency. Meanwhile, different from Peng et al. [39] and GCN-DHSTNet [40], which use dynamic graph convolutional network to exploit the
spatio-temporal correlation, we utilize spatial and temporal information of the visitations to construct a spatial-temporal dependency module to represent users’ long-term and short-term group preferences. Compared with dynamic graph learning model, the proposed dynamic spatio-temporal dependency module is less prone to overfitting. During the construction of graph model, due to the large number of trajectory points and the correlation between the points, when the layer of graph neural network is deep, the noisy points may also be fully learned, resulting in the overfitting of the model and affecting the final prediction accuracy.

III. PROBLEM FORMULATION

We define $U = \{U_1, U_2, U_3, \ldots, U_N\}$, $L = \{l_1, l_2, l_3, \ldots, l_{|L|}\}$ and $V_c = \{C_1, C_2, C_3, \ldots, C_{|V_c|}\}$ as a set of users, locations and categories respectively. Each $l_i$ is associated with its longitude, latitude and category (lon$_i$, lat$_i$, $C_i$). Now, we define some concepts used in our paper.

Definition 1 (Record): A record is a tuple $q = (U_i, l_j, T_k)$, indicating that the user $U_i$ visits $l_j$ at timestamp $T_k$, where $U_i \in U$. $l_j \in L$.

Definition 2 (Trajectory): A person’s trajectory is defined as a record sequence $R_u = \{q_{u1}^U, q_{u2}^U, \ldots, q_{u|R_u|}^U\}$ created by a user $U$, where $q_{ui}^U$ is the user’s i-th check-in record. $R_u$ includes an individual’s all the records, which are arranged in chronological order. Note that the time interval between the two consecutive records is heterogeneous due to the irregular travel behavior.

Definition 3 (Session): A session $S$ is a subsequence of the records in a time slot (e.g. 24 hours), the length of which may be different. A trajectory can be divided into $s$ series of sessions with various strategies. We define the session where the prediction target is located as the short-term session $S_p$ (Current Trajectory) and the previous historical sessions as long-term sessions $S_{q,q} \in \{1, 2, \ldots, p - 1\}$ (Historical Trajectory).

We formulate the next location prediction problem as: given a record sequence $\{q_1, q_2, q_3, \ldots, q_{T-1}\}$ of a specific user $U_i$ and a set of historical trajectories, the goal is to predict where $U_i$ is most likely to go in her next trip at the next timestamp $T$ by learning from the user’s current trajectory and all the users’ historical trajectories.

IV. OUR MODEL

In this part, we propose our PG²Net model. We begin with an overview of our proposed framework, before zooming into the details.

A. Overview

The overall framework of PG²Net is depicted in Fig. 2. Our PG²Net characterizes the user’s preference to various places at both personalized and group level, and fuses them into a unified framework. Personalized preference modeling is to mine information from the historical trajectories of a user, calculating the importance of each record to her, and integrating the sequence information of the record to reflect her personalized preferences. Group preference modeling incorporates the knowledge of statistical physics to excavate the common statistical characteristics which reflect group behavior at a population level. To this end, We construct a statistical physics model which uses temporal and spatial information to model the group preferences of users. Specially, since human’s mobility behavior is affected by her periodic lifestyle and recent travel behavior, we use the historical trajectory to capture her periodic mobility pattern (long-term group preference), and use the recent trajectory to capture the user’s immediate travel decision (short-term group preference).

Specifically, we learn the personalized preference of a specific user $U$ from her historical trajectory $\{S_1^U, S_2^U, S_3^U, \ldots, S_{p-1}^U\}$, which contains longer trajectory data reflecting the general preference of $U$. And different users have different preferences for the same location. Thus we use user-location attention to learn the latent vectors of user $U$ and location $l_i$. Firstly, we learn the latent vectors for user $U$ and record $q_i$ (containing location $l_i$, category $C_i$, and timestamp $T_i$) in the multi-modal trajectory embedding module. Then we use Bi-LSTM to learn the historical trajectory’s transition relationship. Also, we compute the importance of each record $q_i$ to the user, which is denoted as $a_i$. Finally, we integrate the sequence information of the records to present the user’s personalized preference.

With regards to group preference modeling, as human’s decision of moving is impacted by her periodic lifestyle and recent travel behavior, we use historical trajectory to explore her lifestyle while recent trajectory to capture instant decision. Specifically, we firstly learn the latent vectors for $S_p = \{q_1, q_2, q_3, \ldots, q_{T-1}\}$ in the embedding layer, where $q_i$ contains the location $l_i$, category $C_i$, and timestamp $T_i$. To better understand user’s check-in behaviors, we feed the concatenated embedding of $(l_i, C_i, T_i)$ into LSTM. Then we use a statistical spatial-temporal physical module to model a user’s historical trajectory and current trajectory to learn her group preference. Finally, concate layer is used to combine the outputs of the long and short-term group preference and personalized preference, which are then fed into the output layer to generate the final probabilities of candidate locations. Notably, we propose a new graph embedding method to learn the latent vectors of locations and categories. At the same time, in the output layer, we propose an auxiliary loss function to supervise the vectorial representation of the next location and improve the prediction accuracy.

B. Multi-Modal Trajectory Embedding Module

Trajectory sequence usually contains a large amount of human mobility information. Due to the mobile device or the user itself, the trajectory sequence has strong sparseness. Targeting this weakness, we use sequence embedding for this kind of data. For example, in check-in sequence, it contains four different types of attributes, namely user ID, timestamp, location, and location category. We will perform different embedding methods for these different attributes in trajectory sequences.

1) User ID and Timestamp: The original user ID and timestamp cannot be directly inputted into the model. We refer
to the embedding method mentioned in [17] and [41] for these two attributes. As each timestamp \( T_i \) is continuous, which is difficult to embed, we map it into discrete hours. Firstly, we divide one week into 48 slots, where 0-23 slots represent weekdays, and 24-47 slots represent weekends. Each hour is represented as a one-hot 48-dimensional vector, where the non-zero entry denotes the index of the hour. Because one-hot encoding can’t reflect the correlation between sequences, we transform them into \( D_t \) dimensional dense vectors and represent them as \( V^t \in \mathbb{R}^{48 \times D_t} \). For a user ID sequence, we utilize the same embedding method to map it into a dense vector, the dimension of which is \( D_u \). The embedding vector is represented as \( V^u \in \mathbb{R}^{N \times D_u} \), where \( N \) is the number of users.

2) Location and Location Category: In recent years, graph embedding (also known as network embedding) has been applied to many graph related research areas, such as text classification [42], detecting an anomaly in financial networks or social networks [43], [44], etc. The task of our paper is to predict the user’s next location, and all potential locations that the user could reach can construct a graph. Therefore, we attempt to adopt a graph embedding method to learn the location representation. Firstly, we use the training dataset to construct a directed weighted graph, where the nodes are the locations in the training trajectories, the direction is given in the order in which the locations in the trajectories appear, and the weight refers to how often two locations in the trajectory are accessed consecutively. Then we use the graph embedding method node2vec [45] to map each location into a low dimensional vector, the dimension of which is \( D_l \). The embedding vector is represented as \( V^l \in \mathbb{R}^{|L| \times D_l} \), where \(|L|\) is the number of locations. We adopt the same embedding method for the location category sequence. The embedding vector is represented as \( V^c \in \mathbb{R}^{|V_c| \times D_c} \), the dimension of which
is $D_c$, where $|V_c|$ is the number of location categories. Through this method, we can capture the characteristics of group mobility patterns and location interaction. In the following network training, location and location category embedding will no longer be trained.

The embedding of each record $q_i$ containing location, location category, and the timestamp can be represented as:

$$ E_i = [V_i^l \oplus V_i^t \oplus V_i^c] $$

where $V_i^l$, $V_i^t$ and $V_i^c$ are the location, timestamp and category embedding vectors for each record $q_i$ respectively, $\oplus$ denotes concatenation, $E_i$ represents the latent vector for $q_i$. Different from [17] and [20] which only learn the latent vector of location, we further consider the context information, such as the category of location and the check-in time.

### C. Personalized Preference Modeling

When modeling personalized preferences, an intuitive idea is to learn each user’s location preference. Motivated by this, we propose a user-location attention module, and use Bi-LSTM to learn the latent representation $P_u$ of the target user’s personalized preference. As shown in Fig. 3, we firstly embed all the records in each historical trajectory $S_q \in \{S_1, S_2, S_3, \ldots, S_{p-1}\}$, and user embedding into a low-dimensional vector for a user $U$. Then a Bi-LSTM layer is used to learn each record’s high-level representation and sequential dependency. Finally, we compute the importance $a_i$ of each record $q_i$ to the user and integrate the sequence information of all the records to present the user’s personalized preference.

To capture the high-level representation of the user and sequential dependencies of different locations, it is beneficial to mine the context information of the user’s historical trajectory. Different from [17] which used LSTM to process sequences sequentially and ignored context information, we use the Bi-LSTM network to calculate the user’s personalized preference, fully exploiting the context information of historical trajectories. This is important when learning user’s personalized preference.

The calculation of the user’s personalized preference can be summarized as follows:

$$ \overrightarrow{h_i} = LSTM(E_i \overrightarrow{h_{i-1}}) $$

$$ \overleftarrow{h_i} = LSTM(E_i \overleftarrow{h_{i-1}}) $$

$$ h_i = [\overrightarrow{h_i} \oplus \overleftarrow{h_i}] $$

$$ a_i = \frac{\exp(\mathbf{h}^T \mathbf{V}^u)}{\sum_{i=1}^{m} \exp(\mathbf{h}^T \mathbf{V}^u)} $$

$$ P_u = \sum_{i=1}^{m} a_i h_i $$

where $h_i$ represents the hidden information of the record $q_i$ in the historical trajectory, $\oplus$ denotes concatenation representing the combination of the forward and backward outputs, $\mathbf{V}^u$ represents the user’s latent vector, $a_i$ denotes the importance of each record $q_i$ to the user, $m$ is the number of the locations appeared in the historical trajectories, $P_u$ is the final representation of personalized preferences of the user $U$.

### D. Group Preference Modeling

When predicting the user’s next location, her personalized preference can reflect the user’s general preference. However, the user’s interested future trajectory not only follows her personalized preference but also is influenced by the group behavior pattern. At the same time, the user’s preference for the next location changes dynamically with time and space. As a result, we use trajectory data to construct a statistical physical model named dynamic spatial-temporal dependency module, which uses time and spatial information to model user’s group preference and realize next location prediction. See Fig. 4 for the illustration of the modeling of group preference. As shown in Fig. 4, we divide the user’s trajectory into $p$ sessions. Session $p$ is the current trajectory of the user, and the historical trajectory is comprised of $p-1$ sessions which starts from session 1 to session $p-1$. For a specific user, we use the historical trajectory to capture long-term group preference, while use current trajectory to capture the short-term preference respectively. The dynamic spatial-temporal dependency module is comprised of the dynamic spatial dependency module, the dynamic time dependency module, and the dynamic activity preference module, which illustrates the dynamic change of user’s preference from multiple dimensions.

1) Dynamic Spatial Dependency Module: Generally, the distance between geographic locations has a great impact on the user’s next location prediction [20]. The statistical analysis of the distance between adjacent locations is shown in Fig. 5. We can observe that the user’s travel patterns completely follow the distance attenuation rule of group mobility. And the higher the cost of long-distance travel, the lower the probability people choose it. That is to say, users tend to visit nearby locations. Based on this observation, this paper proposes a dynamic spatial dependency module to characterize the changing spatial preferences of users when they move. The module can understand the user’s dynamic interest in geographic locations [46]. When we model the distance preference of users, the key issue is to select the trajectory that has the greatest impact on the recent situation from the historical trajectories based on the distance between different locations. Specifically, we firstly generate a geo-distance matrix based on the real-world geographic locations and the historical trajectories, the value of which represents the distance between any locations. Then we generate the weight vector between the current trajectory and the historical trajectory based on the distance matrix as follows:

$$ \alpha_c(l_{cur}, l_j) = \frac{\exp(1/d(l_{cur}, l_j))}{\sum_{k=1}^{N} \exp(1/d(l_{cur}, l_k))} $$

$$ \alpha_h = (\alpha_h(l_{cur}, l_1), \alpha_h(l_{cur}, l_2), \ldots, \alpha_h(l_{cur}, l_{N_h})) $$

$$ \alpha_e = (\alpha_e(l_{cur}, l_{N_h+1}), \alpha_e(l_{cur}, l_{N_h+2}), \ldots, \alpha_e(l_{cur}, l_{N_h+N_e})) $$

where $l_{cur}$ is the current location, $l_j$ represents the past location, $d(l_{cur}, l_j)$ is the distance between $l_{cur}$ and $l_j$, $\alpha_c(l_{cur}, l_j)$ measures the impact of location $l_j$ on the current location $l_{cur}$, $N_h$ is the number of locations in the historical trajectories, $N_e$ is the number of locations in the current trajectory.
\{l_1, l_2, \ldots, l_{N_h}\} \text{ are locations appeared in the historical trajectory,} \{l_{N_h+1}, l_{N_h+2}, \ldots, l_{N_h+N_c}\} \text{ are locations appeared in the current trajectory, and } \alpha_s^h \text{ is the weight vector between the current location and historical trajectory, } \alpha_s^c \text{ is the weight vector between the current location and the current trajectory. Obviously, the dimensions of } \alpha_s^h \text{ and } \alpha_s^c \text{ are } 1 \times N_h \text{ and } 1 \times N_c \text{ respectively. Finally, we utilize the above generated weight vector to integrate the sequence information of POIs. Specifically, for the current records of a user, we first learn its latent embedding vector before modeling spatial preference. Considering that the historical trajectory and current trajectory have different influences on the current situation of the user, we utilize geo-distance to model long and short-term spatial group preference,  

\begin{align*}
\mathbf{P}_L &= \alpha_s^h \mathbf{H}_h \\
\mathbf{P}_S &= \alpha_s^c \mathbf{H}_c
\end{align*}

where } \mathbf{H}_h \text{ and } \mathbf{H}_c \text{ are the outputs of Bi-LSTM and LSTM, representing the hidden state of historical trajectory and current trajectory respectively, } \mathbf{P}_L \text{ and } \mathbf{P}_S \text{ represent long and short-term spatial group preference of the user respectively. The dimensions of } \mathbf{H}_h \text{ and } \mathbf{H}_c \text{ are } N_h \times K \text{ and } N_c \times K \text{ respectively, where } K \text{ is the size of the network output vector.}

2) Dynamic Time Dependency Module: Traditional methods always consider the influence of time on the next location prediction [17], [26]. However, these methods simply learn the semantic relationship of the timestamp sequences and ignore the interaction between the time sequences. For example,
most users are accustomed to eating at 12:00 and 18:00 in the cafeteria, and drinking coffee at 15:00 and 20:00 in a coffee shop. The location of the users at 12:00 is more related to that at 18:00 rather than 15:00, because both 12:00 and 18:00 are the user’s mealtime. This reflects some group regularities in human mobility. Moreover, as the user moves, her location preferences at different timestamps are also dynamically changing. Therefore, we propose a dynamic time dependency module to capture the influence of the user’s historical trajectory information on the current situation in the temporal dimension.

We first divide one week into 48 slots, where 0-23 slots represent weekdays, and 24-47 slots represent weekends. We construct a location set to represent the location preference of weekdays, and 24-47 slots represent weekends. The location of the users at 12:00 is more in the cafeteria, and drinking coffee at 15:00 and 20:00 in a coffee shop. The location of the users at 12:00 is more in the cafeteria, and drinking coffee at 15:00 and 20:00 in a coffee shop.

Fig. 7. The statistical analysis of location categories in the NYC dataset, where the vertical axis is the number of trajectories.

As shown in Fig. 7, we find that users more tend to work and commute on workdays, and are more inclined to relax on weekends. Users also have different activity preferences at different times on the same day. Based on this observation, we propose a framework that can characterize the user’s activity preference at different time. It is worth to mention that, it is the first time to use graphical representation to define this problem. We attempt to construct a bipartite graph, in which location category and time are the two end nodes, and the correlation between them is the edges of the nodes. See details in Fig. 8.

We define a bipartite graph as $G = (V, E)$. $V = V_c \cup V_t$ is a set containing two types of vertices, where $V_c = \{C_1, C_2, C_3, \ldots, C_{|V_c|}\}$ is a location category set, and $V_t = \{t_1, t_2, t_3, \ldots, t_{|V_t|}\}$ is a timeslot set. $E = \{(C_i, T_j, W_{C_i, T_j})\}$ is a collection of edges with $C_i$ and $T_j$ as its nodes, where $W_{C_i, T_j}$ is the weight of the edge, representing the correlation between the location category $C_i$ and timeslot $T_j$. The task is to learn the user’s activity preference at each timestamp. However, it would be expensive to directly iterate over the bipartite graph. To alleviate this problem, we generate a candidate list of all activities at each moment, and get the correlation between each timestamp and all location categories. Then we generate the weight vector between the recent situation and the historical/current trajectory based on the bipartite graph respectively.

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
where \( f(C_i \land T_j) \) calculates the frequency with which activity \( C_i \) occurs at timestamp \( T_j \). \( W_{C_i,T_j} \) is the weight of activity \( C_i \) at timestamp \( T_j \) in activity preference bipartite graph, \( \alpha^f(C_i, T_{cur}) \) measures the influence of \( C_i \) on \( T_{cur} \). \( T_{cur} \) is the time slot where the current location is in. \( \alpha^c_i \) is the weight vector between current location and the historical trajectory calculated based on the activity preference matrix. \( \{C_i, i = 1, \ldots, N_h\} \) is the activity category of location \( \{l_i, i = 1, \ldots, N_h\} \) in the historical trajectory. \( \alpha^c_j \) is the weight vector between current location and the current trajectory calculated based on the activity preference matrix. \( \{C_j, j = N_h + 1, \ldots, N_h + N_c\} \) is the activity category of location \( \{l_j, j = N_h + 1, \ldots, N_h + N_c\} \) in the current trajectory. The dimensions of \( \alpha^c_i \) and \( \alpha^c_j \) are \( 1 \times N_h \) and \( 1 \times N_c \) respectively.

Finally, we utilize the activity preference bipartite graph to model the long and short-term activity group preference of the user.

\[
P_L^c = \alpha^c_i H_h \quad \text{(22)}
\]

\[
P_S^c = \alpha^c_j H_c \quad \text{(23)}
\]

where \( H_h \) and \( H_c \) are the outputs of Bi-LSTM and LSTM, representing the hidden state of historical trajectory and current trajectory respectively, \( P_L^c \) and \( P_S^c \) represent long and short-term activity group preference of a user respectively. The dimensions of \( H_h \) and \( H_c \) are \( N_h \times K \) and \( N_c \times K \) respectively, where \( K \) is the size of the network output vector.

Ultimately, we model the group preference of a user by integrating the outputs of the three modules mentioned above,

\[
P_L = P_L^c + P_L^r + P_L^e \quad \text{(24)}
\]

where \( P_L \) is the weight matrix of the fully connected layer, \( \oplus \) represents the concatenation of personalized preferences with long and short-term group preference. \( W_f \) is a trainable parameter matrix. Consequently, the index with the largest probability is used as the predicted value of the next location. When training the model, we use negative log likelihood as the loss function. However, in the sequential structure model, the output hidden state can more effectively represent the user’s potential interest [47]. As a result, in order to improve the network’s prediction accuracy, we propose an auxiliary loss function to supervise the hidden state of the user’s target location. Our proposed loss function shown in Fig. 9 is defined as follows,

\[
P = \text{softmax}(W_p (P_u \oplus P_L \oplus P_S \oplus V)) \quad \text{(26)}
\]

where \( P_L \) and \( P_S \) are the long and short-term group preference of a user respectively.

\[
P = \text{softmax}(W_p (P_u \oplus P_L \oplus P_S \oplus V)) \quad \text{(26)}
\]

\[
P_L = P_L^c + P_L^r + P_L^e \quad \text{(24)}
\]

\[
P_S = P_S^c + P_S^r + P_S^e
\]

where \( \oplus \) represents the concatenation of personalized preferences with long and short-term group preference. We choose the L2 loss as our auxiliary loss function. \( \varepsilon \) is used to balance the weight of the prediction and auxiliary loss function. With the help of the auxiliary loss function, the generated hidden vector can better express the user’s interest and increase the accuracy of network prediction.
TABLE I
STATISTICS OF THE THREE DATASETS

| City     | # users | # locations | Timespan |
|----------|---------|-------------|----------|
| New York | 1083    | 227420      | 10 months|
| Tokyo    | 2293    | 573703      | 10 months|
| Shanghai | 1000    | 44476       | 1 months |

V. EXPERIMENTS

In this section, we proceed to evaluate the PG$^2$Net model on three real-world data (two check-in data and one CDRs data). We compare our proposed approach with state-of-the-art next location prediction models, and discuss the experimental results.

A. Datasets

Two Foursquare check-in datasets and one mobile phone dataset used in our paper is introduced in this part.

1) NYC and TKY: We evaluate our model on the publicly available Foursquare check-in data collected from New York City (NYC) and Tokyo (TKY) [48], which are widely used in related studies. The Check-in datasets collected approximately 10 months of check-ins in New York City and TKY via Foursquare from April 12, 2012 to February 16, 2013, which contain the anonymized user ID, location id and its coordinate, location category and timestamp.

2) CDRs: We leverage CDRs data collected from Shanghai to evaluate our model. The CDRs dataset was collected in March 2014 from 1000 anonymous users. It contains the anonymized user ID, the base station ID and its coordinate, and the timestamp. During the lack of category information, we remove the embedding of category information when testing our model on CDRs data.

Table I presents the details of the three datasets. Note that the temporal visitors are removed in check-in datasets via eliminating users those with less than ten records, following previous work [17]. Then, taking three days as an interval, the trajectory of each user is divided into multiple sub-trajectories, and the time interval between each sub-trajectory is limited to more than ten minutes. That is to say, when multiple check-ins occur consecutively within 10 minutes, only the first check-in location is reserved. Next, we limit the number of records in each sub-trajectory for each user to between 5 and 10. Sub-trajectories with less than 5 records are filtered out. When the number of records of a sub-trajectory exceeds 10, we will continue to divide it. The first ten records in this sub-trajectory are regarded as a new sub-trajectory, and the remaining ones comprise of another sub-trajectory. Finally, we use 80% of each users’ trajectories as the training set and the rest as testing set. In Table I, the number of users and Locations in NYC and TKY datasets is the number of users and locations in the original data. In our experiment, the number of users and locations in NYC dataset is 937 and 14001 respectively. And the number of that in TKY dataset is 927 and 16104 respectively.

More information about the three datasets is shown in Fig. 10, where Fig. 10(a) represents the distribution of the number of trajectories for each user, and Fig. 10(b) represents the distribution of the proportion of the number of trajectories in each hour. We can observe from Fig. 10(a) that the NYC and TKY datasets are more sparse than the CDRs dataset. And Fig. 10(b) shows the distributions of TKY and CDRs datasets behave similarly, reflecting similar living habits of the inhabitants in Shanghai and Tokyo.

B. Metrics

For comparing our model with the baselines, we utilize two evaluation indicators: Recall@K and normalized discounted cumulative gain (NDCG@K). Recall@K measures whether there is a correct location among the top K recommended locations. NDCG@K measures the quality of top-K recommended location list. In this paper, we choose K={1,5,10} for comprehensive evaluation. The definitions of Recall@K and NDCG@K are given as follows,

\[
Recall@k = \frac{1}{N} \sum_{u=1}^{N} \frac{|S_u^k \cap S_u^{visited}|}{|S_u^{visited}|} \tag{29}
\]

\[
NDCG@k = \frac{1}{N} \sum_{u=1}^{N} \frac{1}{\log_2(k+1)} \sum_{j=1}^{k} \frac{2^{I(S_u^j \cap S_u^{visited})} - 1}{S_u^j - z_u} \tag{30}
\]

where $S_u^k$ denotes the top-k locations recommended for user u, N is the number of users, $S_u^{visited}$ represents the list of locations visited in the test set, $I(\cdot)$ is an index function, $S_u^j$ represents the j-th location recommended in $S_u^k$, and $z_u$ is the maximum value in $DCG@k$, which is a normalized constant representing the number of records for which each user makes a prediction.

C. Baselines and Settings

To verify the effectiveness of our proposed method, we compare PG$^2$Net with a classic traditional method and some mainstream deep learning methods:

1) Markov Chain (MC): It is widely used to predict human trajectories. It builds a transition matrix based on past trajectories to generate the probability of future locations. In our paper, we use the first-order MC method.

2) LSTM: A neural network-based model, which is a variant model of the recurrent neural network, and can efficiently process sequence data.
layers along the trajectory. temporal information of all the check-ins with self-attention. D locations are model to make the prediction of user's future moves. time-aware category embeddings together into a transformer patterns, user's general preference, spatio-temporal context, and short-term preferences respectively. Our model also performs the best than other baselines with respect to all the metrics in the TKY dataset. The performance of PG\(^2\)Net is the best among all the compared methods in terms of Rec@5, Rec@10, NDCG@5 and NDCG@10, and is comparable with DeepMove with respect to the other two metrics in the CDRs dataset. The quantitative evaluation demonstrates the superior effectiveness of our method.

(2) STAN and GETNext are two recently proposed models. Obviously, our model unequivocally outperforms STAN with regards to all the metrics. For example, our proposed PG\(^2\)Net is 4.42%-13.13% and 1.42%-7.9% higher than STAN with regards to Rec@k(k=1,5,10) and NDCG@k(k=1,5,10) respectively on the NYC dataset. The performance of GETNext is worse than PG\(^2\)Net in terms of all the indicators except for Rec@10 on the NYC dataset. Although both STAN and GETNext exploit spatial-temporal information when predicting the next location, they did not simultaneously consider the effect of both personalized and group preference on the prediction.

(3) Among all the methods, the Markov model shows the worst performance compared with other deep learning methods on the three datasets. This demonstrates that the neural network-based methods have great advantages compared with the traditional method. PLSPL shows better performance than LSTM with respect to all the metrics on the NYC and TKY datasets. That is because PLSPL takes the context information such as category into consideration to learn the specific preference for each user. However, the performance of PLSPL is slightly poor than DeepMove in the NYC and TKY datasets. This phenomenon can be explained that PLSPL cannot derive useful information from historical trajectory based on the current situation.

(4) Among all the baseline methods, the performance of LSTPM and GETNext is comparable, both of which perform better than other baselines. Except for LSTPM and GETNext, DeepMove performs the best. Compared with the DeepMove model, both GETNext and LSTPM model take spatio-temporal factors into consideration, which strongly illustrates the importance of considering the user’s time and spatial factors when predicting the user’s next location. Besides, none of the baselines simultaneously considered the impact of the individual and group preference in predicting the next place. Moreover, PG\(^2\)Net considers rich contextual information of user’s check-in sequence by incorporating the statistical physical characteristics and neural networks to capture the fine-grained

3) **Deepmove**: [17] A neural network model based on the attention mechanism, which uses each user’s historical and current trajectories to learn her preferences. The attention mechanism is used to capture the correlation between long-term and short-term trajectories.

4) **PLSPL**: [26] A neural network model to learn the specific preference for each user, which takes category information into consideration when constructing the network for the first time. Due to the lack of category information for CDRs, our model cannot be compared with PLSPL method in CDRs.

5) **LSTPM**: [20] It is the state-of-the-art model for next location prediction, which uses context-aware non-local network structure and geo-dilated RNN to capture users’ long and short-term preferences respectively.

6) **STAN**: [36] It explicitly exploits relative spatio and temporal information of all the check-ins with self-attention layers along the trajectory.

7) **GETNext**: [38] It incorporates the global transition patterns, user’s general preference, spatio-temporal context, and time-aware category embeddings together into a transformer model to make the prediction of user’s future moves.

For our method, the embedding dimension of users and locations are \(D^u = 40\) and \(D^l = 500\) respectively. We set the embedding dimension of categories and timestamps to be \(D^c = 50\) and \(D^t = 10\) respectively. The dimension of the hidden state is 500. We use Adam which is a gradient descent optimization algorithm to learn all the parameters in our model. We set the initial learning rate and regularization weight to be 0.0001 and 1e-5 respectively. In the training process, we adopt the gradient cutting method and adjust the learning rate to ensure that the model has the best performance. We take TKY dataset as an example to show the training process of the proposed model. See details in Fig. 11. For other baseline models, we set their parameters to be the default values that come with the original paper.

### D. Result and Analysis

The experimental results are reported in Table II. The best results in each column are highlighted in boldface. It shows that,

![Fig. 11. Loss in the training and test process of the TKY dataset.](image-url)
preferences of users. This is the reason why our model achieves the best performance in terms of most metrics in the three datasets.

(5) Although the deep learning methods perform well on the check-in datasets, they do not improve much in terms of Rec@1 and NDCG@1 when compared with the Markov method on the CDRs dataset. We argue this probably is related to the sparsity of the trajectory data. The check-in datasets are much sparser than CDRs. With respect to sparse data, deep learning methods can capture high-level semantic information while the traditional methods cannot do well in this.

(6) To validate the stability of the model’s performance, we reported the standard deviation for all the compared methods. Although Markov method shows poor performance when predicting next location, it is stable on the three datasets. As one of the best baselines, the performance of GETNext is the most stable on both NYC and TKY datasets. However, on these two datasets, our PG2Net performs better than GETNext in most cases. On the CDRs dataset, PG2Net shows the best stability and performance than all the baselines. DeepMove also has good stability on this dataset, second only to our approach.

(7) We also take prediction speed as an evaluation metric. The experimental result is shown in Fig. 12, which shows the time to predict 7780 trajectories of different models. As shown in Fig. 12, Markov model shows the fastest prediction speed, although its prediction performance is very poor. DeepMove shows brilliant performance in prediction speed. Compared with STAN and GETnext, our method is superior to them both in prediction performance and prediction speed.

E. Model Ablation Study

In this section, we analyze four variants of PG2Net to further evaluate the effectiveness of our model. The four variants are shown as follows.

---

| Datasets | Methods | Rec@1 | Rec@5 | Rec@10 | NDCG@1 | NDCG@5 | NDCG@10 |
|----------|---------|-------|-------|--------|---------|---------|---------|
|          | Markov  | 0.1328(0.00251) | 0.2679(0.00256) | 0.3558(0.00232) | 0.3328(0.00251) | 0.2038(0.00251) | 0.2256(0.00248) |
|          | LSTM    | 0.1433(0.00180) | 0.2937(0.00163) | 0.3316(0.00212) | 0.1433(0.00080) | 0.2250(0.00125) | 0.2573(0.00129) |
|          | DeepMove | 0.1762(0.00167) | 0.3946(0.00217) | 0.4700(0.00189) | 0.1762(0.00167) | 0.2919(0.001478) | 0.3158(0.001603) |
|          | PLSP [6] | 0.1533(0.00239) | 0.3239(0.00974) | 0.3984(0.001574) | 0.1533(0.00239) | 0.2484(0.001238) | 0.2699(0.001128) |
|          | LSTM [20] | 0.1823(0.00711) | 0.4303(0.00585) | 0.5228(0.002186) | 0.1823(0.00711) | 0.3126(0.00319) | 0.3428(0.00345) |
|          | STAN [36] | 0.1585(0.00545) | 0.3484(0.002439) | 0.3813(0.002439) | 0.1865(0.005454) | 0.2644(0.001515) | 0.2745(0.001359) |
|          | GETNext [38] | 0.1874(0.00033) | 0.4282(0.000009) | 0.5459(0.000071) | 0.1874(0.00033) | 0.3127(0.000134) | 0.3070(0.000026) |
|          | PG2Net | 0.2027(0.000840) | 0.4410(0.003950) | 0.5126(0.006228) | 0.2027(0.000840) | 0.3308(0.001775) | 0.3538(0.002393) |

---

Table II: Performance Comparison With Seven Baselines on Three Datasets. Values in parentheses are standard deviations of the model’s performance. The best performance is highlighted in bold.

---

Fig. 12. The prediction speed of different models.

1) GNet: a variant model which only explores the group preference of users, containing long-term group preference and short-term group preference.

2) PNet: a variant model which only considers the personalized preference of users, removing dynamic spatio-temporal dependency module.

3) L-PG2Net: a variant model which takes the personalized preference and long-term group preference into consideration.

4) S-PG2Net: a variant model which mines the personalized preference and short-term group preference of users.
TABLE III

| Datasets | Methods  | Rec@1 | Rec@5 | Rec@10 | NDCG@1 | NDCG@5 | NDCG@10 |
|----------|----------|-------|-------|--------|--------|--------|---------|
| NYC      | PG\textsuperscript{2}Net | 0.2120 | 0.4585 | 0.5326 | 0.2120 | 0.3437 | 0.3679 |
|          | GNet     | 0.1795 | 0.4304 | 0.5176 | 0.1795 | 0.3116 | 0.3400 |
|          | PNet     | 0.1851 | 0.4107 | 0.4932 | 0.1851 | 0.3098 | 0.3364 |
|          | L-PG\textsuperscript{2}Net | 0.1918 | 0.4245 | 0.5003 | 0.1918 | 0.3159 | 0.3407 |
|          | S-PG\textsuperscript{2}Net | 0.2019 | 0.4462 | 0.5227 | 0.2019 | 0.3299 | 0.3550 |
| TKY      | PG\textsuperscript{2}Net | 0.1994 | 0.4336 | 0.5105 | 0.1994 | 0.3240 | 0.3490 |
|          | GNet     | 0.1645 | 0.4103 | 0.4852 | 0.1645 | 0.3065 | 0.3223 |
|          | PNet     | 0.1782 | 0.3951 | 0.4762 | 0.1782 | 0.3027 | 0.3203 |
|          | L-PG\textsuperscript{2}Net | 0.1848 | 0.3997 | 0.4801 | 0.1848 | 0.3099 | 0.3247 |
|          | S-PG\textsuperscript{2}Net | 0.1890 | 0.4224 | 0.4981 | 0.1890 | 0.3134 | 0.3379 |
| CDRs     | PG\textsuperscript{2}Net | 0.2346 | 0.5981 | 0.7021 | 0.2346 | 0.4262 | 0.4604 |
|          | GNet     | 0.2302 | 0.5956 | 0.6995 | 0.2302 | 0.4232 | 0.4573 |
|          | PNet     | 0.2320 | 0.5926 | 0.6954 | 0.2320 | 0.4207 | 0.4545 |
|          | L-PG\textsuperscript{2}Net | 0.2325 | 0.5936 | 0.6975 | 0.2325 | 0.4214 | 0.4595 |
|          | S-PG\textsuperscript{2}Net | 0.2337 | 0.5970 | 0.6993 | 0.2337 | 0.4229 | 0.4565 |

Fig. 13. The proportion of personalized preference dominated and group preference dominated in next place prediction on three datasets.

The proportion that GNet can accurately predict users’ trajectories, while PNet can achieve rough prediction of users’ trajectories. To further demonstrate the effectiveness of personalized preference and group preference, we assess whether the personalized preference dominates or group preference dominates in predicting the next location on the three datasets. Taking the NYC dataset for example, We conduct the assessment in the following manner. When only the individual preference module is used, the Rec@1 of the model is 0.1851, and when only the group preference model is used, the value of this model at Rec@1 is 0.1795. It can be seen that the performance of the individual preference module is better than that of the group preference module, indicating that the individual preference module is more effective. In this paper, 0.1851/(0.1851+0.1795) and 0.1795/(0.1851+0.1795) are respectively used to represent the proportion of group preference dominated and individual preference dominated in the NYC dataset. The assessment on the other two dataset can be derived in the same way. The experimental results are summarized in Fig. 13. As shown in Fig. 13, the user’s personalized preference has a greater influence on her trajectory prediction than the group preference. (2) S-PG\textsuperscript{2}Net always performs better than L-PG\textsuperscript{2}Net. This is mainly because that S-PG\textsuperscript{2}Net can better capture user’s group preferences based on her recent state. It also demonstrate that the recent trajectory has a greater impact on the current situation. (3) The proposed model PG\textsuperscript{2}Net, which is the combination of PNet and GNet, achieves the best performance on all test datasets. It shows that both personalized preference and group preference have positive impact on the user’s choice of the next location.

F. Importance Evaluation of Key Components in PG\textsuperscript{2}Net

To better understand the influence of the node2vec embedding method and auxiliary loss function on network training, we use the NYC dataset to evaluate the importance of each module in PG\textsuperscript{2}Net. As shown in Fig. 14, PG\textsuperscript{2}Net is our proposed complete model, PG\textsuperscript{2}Net-Node2vec denotes that there is no graph embedding (node2vec) to embed user locations and location categories in the model, and PG\textsuperscript{2}Net-Auxiliary Loss represents that the influence of the hidden state of the target location is not considered in the prediction module. Fig. 14 shows that our complete model PG\textsuperscript{2}Net performs the best, and other variant models have decreased performance. Among them, the performance of the PG\textsuperscript{2}Net-Auxiliary Loss model drops the most, with a drop of 2.68% in Rec@5, indicating that the hidden vector of the target position has a great influence on the prediction accuracy of the next position. In addition,
the performance of the PG$^2$Net-Node2vec model is worse than PG$^2$Net, which demonstrates that graph embedding training on locations and location categories can improve model performance.

### G. Analysis of Spatial Distribution of Predicted Locations

To show the ability of our model to predict next locations at different distances, we examine the distance distribution between the current and the next predicted locations on the three datasets. For each dataset, we compare the actual distances with the distances predicted by the LSTM, DeepMove, and PG$^2$Net respectively. Fig. 15 shows that when the prediction distance is short, PG$^2$Net has a similar performance with LSTM and DeepMove. While when performing long-distance prediction, PG$^2$Net outperforms the LSTM and DeepMove model, and can effectively predict long-distance locations. In addition, DeepMove always outperforms LSTM on the three datasets which is more obvious on the CDRs dataset. As shown in Fig. 15, when predicting locations at distance less than 20 km, the performance of LSTM and DeepMove is comparable to that of our model. When the next places are far away, e.g., over 40 km, the performance of LSTM and DeepMove gradually deteriorate. For places locating over 80 km, the distance predicted by the LSTM model has completely deviated from the real distance distribution, and the performance is the worst, followed by DeepMove. In this scenario, the distance distribution of the predicted locations by PG$^2$Net matches well with the empirical data. The reason could be that, the LSTM and DeepMove models only learn the sequence relationship of the trajectory and fail to take into account the user's personalized characteristics and the spatial-temporal information reflecting the regularity of group behavior. While the influences of two locations at a long distance on the current trajectory are similar, this could cause the LSTM and DeepMove model to be unable to distinguish the two locations, resulting in long-distance jumping of the predicted locations.

VI. CONCLUSION AND FUTURE WORK

In this paper, we propose a novel end-to-end deep neural network PG$^2$Net to predict the next place to visit via considering users’ preferences to various locations at both individual and collective level. In the personalized preference module, we use Bi-LSTM and the attention mechanism to capture the users’ personalized long-term mobility tendency. In the group preference module, we use spatial-temporal and categorical information of the visited places to represent users’ long-term and short-term group preferences. In addition, we utilize a graph embedding method, node2vec, to capture the sequential relation of users’ visited locations and propose an auxiliary loss function to learn the vectorial representation of the target location. The extensive experimental results on three real-world datasets demonstrate the effectiveness of our proposed model. In future work, we will model more heterogeneous information and use graph neural networks to learn the interaction between them to further improve the performance of next POI recommendation.

REFERENCES

[1] Z. Zhang, C. Li, Z. Wu, A. Sun, D. Ye, and X. Luo, “NEXT: A neural network framework for next POI recommendation,” Frontiers Comput. Sci., vol. 14, no. 2, pp. 314–333, Apr. 2020.
[2] Y. Xu, R. Di Clemente, and M. C. González, “Understanding vehicular routing behavior with location-based service data,” EPJ Data Sci., vol. 10, no. 1, pp. 1–17, 2021.
[3] M. C. Gonzalez, C. A. Hidalgo, and A.-L. Barabasi, “Understanding individual human mobility patterns,” Nature, vol. 453, no. 7196, pp. 779–782, 2008.
[4] S. Jiang, Y. Yang, S. Gupta, D. Veneziano, S. Athavale, and M. C. González, “The TimeGeo modeling framework for urban mobility without travel surveys,” Proc. Nat. Acad. Sci. USA, vol. 113, no. 37, pp. E5370–E5378, Sep. 2016.
[5] C. Song, Z. Qu, N. Blumm, and A.-L. Barabási, “Limits of predictability in human mobility,” Science, vol. 327, no. 5968, pp. 1018–1021, Feb. 2010.
[6] S. Çolak, A. Lima, and M. C. González, “Understanding congested travel in urban areas,” Nature Commun., vol. 7, no. 1, Mar. 2016.
[7] Y. Xu and M. C. González, “Collective benefits in traffic during mega events via the use of information technologies,” J. Roy. Soc. Interface, vol. 14, no. 129, Apr. 2017, Art. no. 20161041.
[8] L. E. Olmos, S. Çolak, S. Shafei, M. Saberi, and M. C. González, “Macroscopic dynamics and the collapse of urban traffic,” Proc. Nat. Acad. Sci. USA, vol. 115, no. 50, pp. 12654–12661, Dec. 2018.
[9] X. Xu et al., “Unraveling environmental justice in ambient PM2.5 exposure in Beijing: A big data approach,” Comput. Environ. Urban Syst., vol. 75, pp. 12–21, May 2019.
[10] X. Xu, S. Çolak, E. C. Kara, S. J. Moura, and M. C. González, “Planning for electric vehicle needs by coupling charging profiles with urban mobility,” Nature Energy, vol. 3, no. 6, pp. 484–493, Apr. 2018.
[11] M. Tizzoni et al., “On the use of human mobility proxies for modeling epidemics,” PLoS Comput. Biol., vol. 10, no. 7, Jul. 2014, Art. no. e1003716.
[12] S. Chang et al., “Mobility network models of COVID-19 explain inequities and inform reopening,” Nature, vol. 589, no. 7840, pp. 82–87, Jan. 2021.
[13] F. Alhasoun, M. Alhazzani, F. Aleissa, R. Alnasser, and M. González, “City scale next place prediction from sparse data through similar strangers,” in Proc. ACM KDD Workshop, 2017, pp. 191–196.
[14] S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme, “Factorizing personalized Markov chains for next-basket recommendation,” in Proc. 19th Int. Conf. World Wide Web, Apr. 2010, pp. 811–820.
Weipeng Wang received the B.S. degree in software engineering from Shandong University, China, in 2022. He is currently pursuing the Ph.D. degree with the Department of Computer Science and Engineering, Shanghai Jiao Tong University. His research interests include urban computing and spatio-temporal data mining.

Menghan Wang received the Ph.D. degree in computer science from Zhejiang University in 2019. He is currently an Applied Researcher with eBay Inc., working on developing and deploying scalable and extensible recommendation algorithms. His research interests include statistical machine learning and recommender systems.

Yaohui Jin was a Technical Staff Member of Bell Labs Research China. After that, he joined Shanghai Jiao Tong University in 2002, where he is currently a Professor with the State Key Laboratory of Advanced Optical Communication Systems and Networks and the Deputy Director of the Network and Information Center. His research interests include civic engagement and open innovation, cloud computing network architecture, and streaming data analysis.

Yanyan Xu received B.E. and M.S. degrees from Shandong University in 2007 and 2010, respectively, and the Ph.D. degree from the Department of Automation, Shanghai Jiao Tong University, in 2015. He is currently an Associate Professor with the AI Institute, Shanghai Jiao Tong University. Prior to joining SJTU, he was a Post-Doctoral Associate with the Human Mobility and Networks (HuMNet) Laboratory, Department of Urban and Regional Planning, UC Berkeley, and the Department of Civil and Environmental Engineering, MIT, from 2015 to 2020. He was also a Guest Post-Doctoral Fellow with the Energy Analysis and Environmental Impacts Division, Lawrence Berkeley National Laboratory, from 2017 to 2018. His work has been published in Nature Energy, Science Advances, Journal of the Royal Society Interface, IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, TRC, and IJCAI. His research interests include human mobility and urban computing, with a particular emphasis placed on the use of massive trajectory data in smart cities, transportation, energy, and the environment from an interdisciplinary perspective.