Revisiting Long- and Short-Term Preference Learning for Next POI Recommendation with Hierarchical LSTM

Chen Wang, *Senior Member, IEEE*, Mengting Yuan, Yang Yang, *Member, IEEE*, Kai Peng, and Hongbo Jiang, *Senior Member, IEEE*

Abstract—Point-of-interest (POI) recommendation has drawn much attention with the widespread popularity of location-based social networks (LBSNs). Previous works define long- and short-term trajectories via long short-term memory (LSTM) to capture user's stable and current preference, and incorporate context factors to improve recommendation effectiveness. However, these factors have different impacts on POI recommendation, and meanwhile, they are mutually influenced. Existing studies either model all the factors separately, or feed them into the same LSTM model, which are less meticulous for modeling the LBSNs trajectories. To address such issues, we revisit the long- and short-term preference learning for next POI recommendation by presenting a novel framework that can model both POI level and semantic level check-in trajectories. We develop a hierarchical LSTM to learn the two-level representations and consider the interplay of the two-level features by adding factors to the gates of LSTMs for each trajectory. We further construct a semantic filter to improve the recommendation efficacy. Experimental results using two real-world check-in datasets indicate that the proposed framework outperforms four state-of-the-art baselines regarding two commonly used metrics.

Index Terms—POI recommendation, location-based social networks, hierarchical LSTM, long- and short-term preference.

1 Introduction

Nowadays, location-based social networks (LBSNs) [1]–[3] have received much attention owing to the popularity of smart mobile devices and the advancement of location acquisition technology. Millions of users have registered in LBSNs services like Facebook or Foursquare. Users can post their check-ins and share their life experience in the real-world via LBSNs. In order to improve the experience for users, next point-of-interest (POI) recommendation [4]–[6] that aims to recommend next potentially attractive POIs to users has gained considerable research interests, as it can benefit not only users but also advertising agencies with an effective way to launch advertisements.

To perform the next POI recommendation, historical check-in trajectories, which can dynamically reflect user's stable preference, are often adopted to capture personal general taste. Meanwhile, people make their next visits based on their current locations to a great extent. Therefore, previous works define short-term and long-term check-in trajectories to capture user's current and stable preference, where long short-term memory (LSTM) is usually accompanied by the long- and short-term trajectory modeling [7]–[13].

However, modeling the check-in trajectory is challenging for data with heterogeneity and sparsity. To improve the recommendation effectiveness, many efforts [9]–[13] focus on incorporating context information like spatio-temporal contextual knowledge for POI recommendation and have gained promising results. Specifically, some studies argue that context factors (e.g., category, check-in time, geographical location) are useful for next POI recommendation and they input all factors into the same LSTM model. Recent studies [14]–[18] take a more fine-grained approach to deal with these factors and they explore the impact of each specific factor, and different impacts of different factors have been demonstrated on POI recommendation [10], [17].

Nevertheless, these factors are also mutually influenced. We notice that some factors are geographically relevant, while some are semantically related. To validate this point, we randomly select a user from the real-world Foursquare NYC dataset and figure out the check-in distribution of different POIs with the same category that the user visits the most. Fig. 1 shows the check-in distribution of the same user at two coffee shops/bars. It can be observed that the check-in distribution for two coffee shops are similar, so for the bars. User prefers visiting coffee shops late at night and visiting bars at noon and night with no regard to physical positions. This indicates that the user may like visiting different POIs with the same category at a similar time slot.

- *This work was supported in part by the National Natural Science Foundation of China under Grants 61872416, 62272183, 62171189 and 62002104; by the Key R&D Program of Hubei Province under Grant 2023BAB074; by the special fund for Wuhan Artificial Intelligence Innovation under Grant 2022010702040061; by the Open Project Funding of the Key Laboratory of Intelligent Sensing System and Security, Ministry of Education under Grant KLISSS202401; and by Hunan Provincial Science and Technology Projects under Grants 2021RC4023, 2021WK2001 and 2019CB1008* (Corresponding author: Kai Peng).
- *C. Wang, M. Yuan and K. Peng are with the Hubei Key Laboratory of Smart Internet Technology, School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan 430074, China. Email: {chenwang, mtyuan, pkhust}@hust.edu.cn.*
- *Y. Yang is with the School of Artificial Intelligence, Hubei University, Wuhan 430062, China. He is also with the Key Laboratory of Intelligent Sensing System and Security, Ministry of Education, China, 430062. Email: yangyang@hubu.edu.cn.*
- *H. Jiang is with College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China. Email: hongbojiang2004@gmail.com.*

We thus call the specific POIs geographically relevant while the category and check-in time semantically relevant. They have the holistic effect on the next visit in two levels, and respectively reflect the geographically constrained characteristics and the unrestricted actual user intent.

In addition, the interplay between POI level and semantic level trajectories is non-negligible, as the next POI visit is affected by the semantic level intent and the intent is constrained by the geographical position, and meanwhile the semantic level features also have an impact on POI level learning. For example, a user may prefer a bar after shopping on weekends; he may not change the preference even if he goes to another shopping mall, and another bar around is the most likely to be visited next. However, existing studies either model all the factors separately, or feed all the factors into the same LSTM model and they are obviously not that meticulous for modeling LBSNs trajectories.

Against this background, in this paper, we revisit the long- and short-term preference learning for next POI recommendation by presenting a novel **Hi**erarchical LSTM with **L**ong- and **S**hort-term preference framework (HiLS) that can model both POI level and semantic level features. The influencing factors are expressed as embeddings to transform the sparse feature into dense representations that are further input into HiLS. To consider the interplay of the POI and semantic level features, we design a hierarchical LSTM to guide the learning process. By feeding the semantic level features into the POI level in each step, the POI level learning will be affected by the semantic level features, which will be updated in turn by predicting the next POI. We make the best of semantic level preference to predict the user's next location. With the semantic level features, which reflect the user intent, we further construct a semantic filter to preliminarily filter out POIs that are in consist with user intent before recommendation. In this way, the semantic level features will be made better use of to help improve the effectiveness of the final recommendation.

Our major contributions are summarized as follows:

- We propose a novel framework which can learn the long- and short-term preference for more effective next POI recommendation.
- We design a hierarchical LSTM, coined as HiLSTM, to learn both the semantic level and POI level representations for trajectories, and model the interplay of the two levels by feeding the semantic factor into LSTM gates.
- We utilize the short-term trajectory to capture the check-in sequential correlation with the hierarchical LSTM, and learn the long-term preference by exploring the correlation between historical and current trajectories with attention mechanism.
- We conduct extensive experiments to evaluate the performance of HiLS on two real-world datasets. The results show the effectiveness and superiority of HiLS by comparing with state-of-the-art baselines. The code of HiLS has been released for reproducibility purposes[1].

1. https://www.dropbox.com/s/q0ggpirmg0kbtdp/HiLS-Code.zip?dl=0

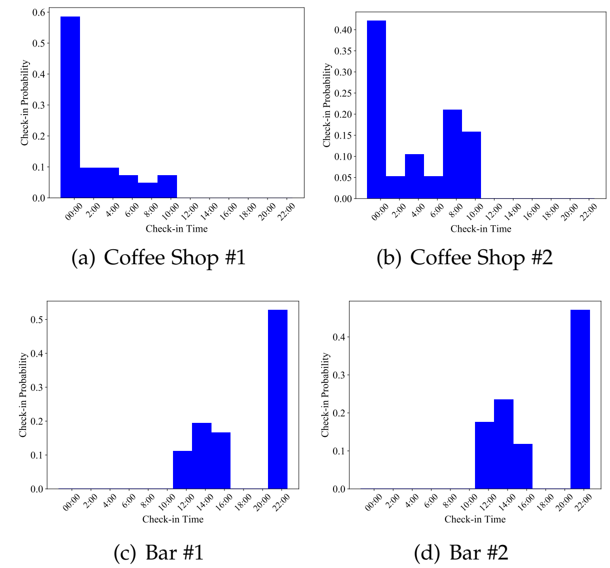


Fig. 1: Check-in time distribution on coffee shops/bars #1 and #2, which show a similar pattern, while the check-in distribution of the coffee shop is different from that of the bar, indicating that the user may like visiting different POIs with the same category at similar time slot.

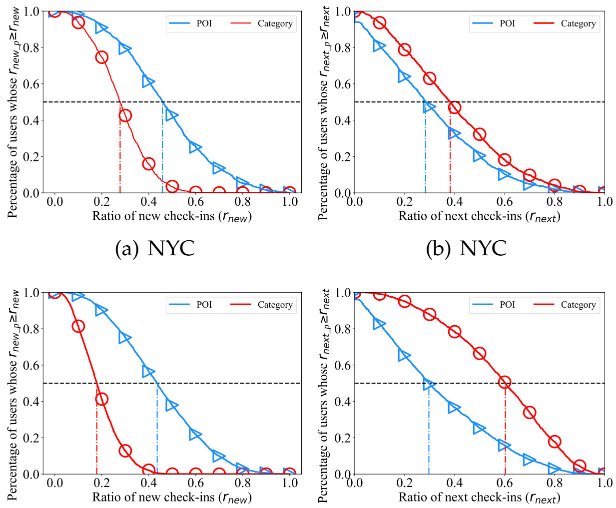
2 Preliminary

2.1 Empirical Data Analysis

We first conduct an empirical data analysis to reveal the influence of factors on user mobility, using two datasets collected by [19] from Foursquare between Apr. 2012 and Jan. 2014. The detailed statistics of the datasets are shown in Table 1. Different from recommendation on the web, POI recommendation is related to the physical location and the check-ins are determined by a variety of influencing factors. We therefore preliminarily analyze the impact of POI, category, timestamp and geographical distance on the next visit.

POI influence. We analyze the check-in pattern on POIs from two perspectives. First, we analyze the ratio of new POIs in all collected records for each user. As shown in Fig. 2(a), in NYC dataset, for 50% users, nearly half of the POIs are the first visits; similar trend can be observed in TKY dataset as in Fig. 2(c). This may be because users are enthusiastic in new POIs or they are bored with reporting recurring visits. So, it reminds us to consider the fact that users may frequently check in at unvisited POIs. Second, we analyze the probability of checking in at the same POIs. As shown in Fig. 2(b) and Fig. 2(d), users are likely to check in following the similar pattern and will check in next at the same POI previously visited.

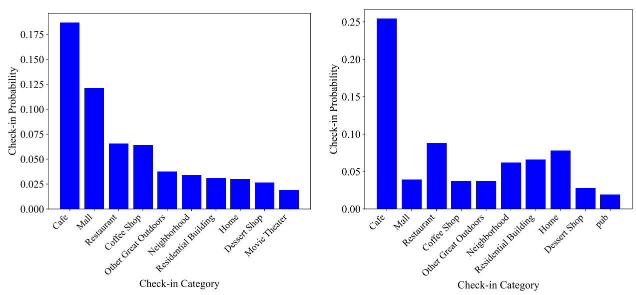
Category influence. We analyze the category influence in the same way as POI influence. We can see from Fig. 2(a), for 50% users, less than 30% categories are the first visit while nearly half for POIs. Even if recurring visits are omitted, the data shows that users visit new locations with familiar categories and they are enthusiastic enough to report them. Users may not check in at the same POI frequently. However, the ratio of new check-in categories is much less than that of POIs. We can see a more obvious tendency in



(a) NYC (b) NYC

(c) TKY (d) TKY

Fig. 2: Cumulative distribution of POI and category influence on next POI recommendation.



(a) Between 10:00-14:00 (b) Between 16:00-20:00

Fig. 3: Top-10 check-in categories distribution after checking-in on restaurant.

TKY dataset as in Fig. 2(c). Besides, as shown in Fig. 2(b) and Fig. 2(d), the ratio of checking in next at the same category previously visited is more than at the same POI. The categories reflect more intrinsic preference, and it is essential to learn the check-in intent other than certain POIs.

Timestamp influence. Fig. 3 visualizes the check-in probability on next POIs at different timestamps on NYC dataset. We can observe that users are most likely to check in at cafes, malls and restaurants after checking in at a restaurant for lunch, while they are more likely to check in at cafes, home and restaurants after dinner. Besides, the check-in categories after lunch are more dispersive, which may because people have more plentiful activities in the afternoon than in the evening, and thus it makes more sense to recommend a mall than a restaurant after lunch. Apparently, people have different next check-in preference at different time even at the same POI.

Geographical distance influence. Geographical influence is an important factor that distinguishes the recommendation in LBSNs from other scenarios, as the next check-in location is constrained by the distance from one's current position. Fig. 4 shows the distribution of the standard deviation between consecutive check-ins of sub-trajectories on

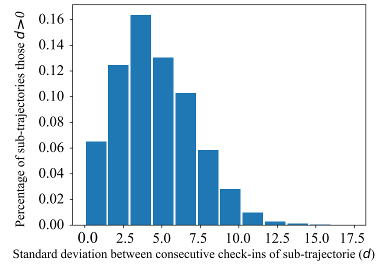


Fig. 4: The distribution of standard deviation of geographical distances for sub-trajectories.

NYC dataset. We can observe that most of sub-trajectories have standard deviation less than 7.5km. The observation supports that geographical distance between consecutive visits in a sub-trajectory is restricted.

2.2 Motivation of Hierarchical LSTM

LSTM [20] is measurable as the basic unit in modeling the specific continuous order of the POIs resulting from a certain chronological order between user activities. It is an optimized variant of the famous recurrent neural network (RNN) [21], and is able to avoid the vanishing gradient problem by introducing the gate mechanism. The basic LSTM consists of one cell state and three gates to control the output and update of LSTM cell. Based on the previous cell state and the input, LSTM first updates cell states with part to keep and part to forget, and then generates the output from the current cell for the next cell.

The basic update equations of LSTM as follows [20]:

$$\boldsymbol{i}_t = \sigma(\boldsymbol{W}_{ix}\boldsymbol{x}_t + \boldsymbol{W}_{ih}\boldsymbol{h}_{t-1} + \boldsymbol{b}_i) \tag{1}$$

$$\boldsymbol{f}_t = \sigma(\boldsymbol{W}_{fx}\boldsymbol{x}_t + \boldsymbol{W}_{fh}\boldsymbol{h}_{t-1} + \boldsymbol{b}_f) \tag{2}$$

$$\boldsymbol{o}_t = \sigma(\boldsymbol{W}_{ox}\boldsymbol{x}_t + \boldsymbol{W}_{oh}\boldsymbol{h}_{t-1} + \boldsymbol{b}_o) \tag{3}$$

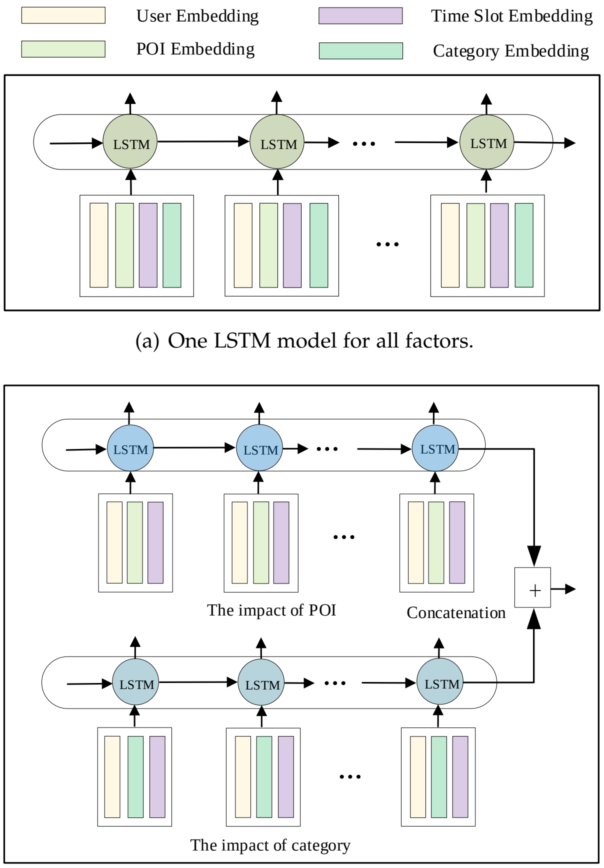
$$\tilde{\boldsymbol{c}}_t = \tanh(\boldsymbol{W}_{cx}\boldsymbol{x}_t + \boldsymbol{W}_{ch}\boldsymbol{h}_{t-1} + \boldsymbol{b}_c) \tag{4}$$

$$\boldsymbol{c}_t = \boldsymbol{f}_t \odot \boldsymbol{c}_{t-1} + \boldsymbol{i}_t \odot \tilde{\boldsymbol{c}}_t \tag{5}$$

$$\boldsymbol{h}_t = \boldsymbol{o}_t \odot \tanh(\boldsymbol{c}_t) \tag{6}$$

where $\boldsymbol{i}_t$, $\boldsymbol{f}_t$, $\boldsymbol{o}_t$ represent the input, forget and output gates, respectively, $\boldsymbol{c}_t$ is the cell activation vector representing cell state, $\boldsymbol{x}_t$, $\boldsymbol{h}_t$ and $\boldsymbol{h}_{t-1}$ represent the input feature, hidden output vector and the last output of cell unit, respectively, and $\odot$ is the operation of element-wise multiplication. Each LSTM cell computes $\boldsymbol{h}_t$ which incorporates the current information and the information before time t. There is also a learnable weight $\boldsymbol{W}$ to control the update.

LSTM has achieved remarkable success in sequential prediction [23], and has been recently introduced for next POI recommendation [7], [17], [22]. A usual practice is to input all the influential factors into one LSTM model [7], [22] (c.f. Fig. 5(a)), while ignoring different impacts of different factors. The recent study [17] takes a more fine-grained approach and explores the impact of different factors separately before concatenating their results (c.f. Fig. 5(b)). We notice that different factors are also correlated with each other and it is not appropriate to use separated models



(a) One LSTM model for all factors.

(b) Each LSTM model for one specific factor.

Fig. 5: (a) All factors are fed into one LSTM model [7], [22]; and (b) different factors are fed into respective LSTM models [17], for next POI recommendation.

to learn the inherent mutual influence. We thus propose a hierarchical LSTM structure to learn the user preference on semantic level and POI level while considering their interplay by feeding the semantic features in each step to the gates of POI level LSTM to guide the learning process.

3 Problem Formulation

Before we formulate the problem, we first present some notations and definitions. Formally, let the quadri-tuple $v = (t, d, c, p)$ denotes a check-in record of a user, which indicates that the user visits POI p at timestamp t, with the category of p as c, and d is the geographical distance between the last and current POIs. Since user's movements are periodic, directly using all the historical check-ins will result in an undesirably long trajectory. Therefore, enlightened by the existing work [24]–[26], given the raw check-ins $Tr = \{v_1, v_2, ...\}$ of the user, we split it into multiple sub-trajectories $Tr = \{\boldsymbol{Tr}_1, \boldsymbol{Tr}_2 ..., \boldsymbol{Tr}_I\}$ by the time interval and check-in consecutiveness, as in recent works [8], [11], [27], [28], where I is the number of sub-trajectories.

As is known that the user activity for visiting POIs is influenced by both the general taste and the status at present, so long- and short-term trajectories are adopted to respectively capture user's stable and current preference [7], [8], [16], [17], [22], [29]. Their formal definitions are described as follows:

Definition 1 (Long/Short-Term Trajectory [8]). The long-term trajectory is a sequence of historical sub-trajectories, which is denoted as $\boldsymbol{Tr}_{long} = \{\boldsymbol{Tr}_1, \boldsymbol{Tr}_2 ..., \boldsymbol{Tr}_{I-1}\}$, while the short-term trajectory is the user's most recently visited sub-trajectory, i.e., $\boldsymbol{Tr}_{short} = \{\boldsymbol{Tr}_I\}$.

The long-term trajectory is regarded as the general check-in pattern of the user and can be used to mine the stable tastes, and the short-term trajectory indicates the consecutiveness influenced by the recently visited POIs.

Besides the long- and short-term preference of the user, different influential factors (e.g., category, check-in time, geographical location) also have impacts on the next POI recommendation [10], [14], [16], [17], as discussed in the previous section. While these factors are normally treated as unitary features, we are keenly aware of the disparity on their influence. In particular, POIs are geographically relevant, which mainly describe the user's geographical constrained preference, while others are semantically relevant and can reflect more on the user's intentional preference free from geographical constrains. We thus define the POI level and semantic level trajectories separately, on the basis of the sub-trajectory, to capture such preference.

Definition 2 (POI Trajectory). The POI trajectory of a user is built on time-ordered sequence of L POIs in check-ins, i.e., $\boldsymbol{Tr}_{poi} = \{p_{v_1}, p_{v_2}, ..., p_{v_L}\}$, where L is the length of the sub-trajectory.

Definition 3 (Semantic Trajectory). A user's semantic trajectory is composed of a semantic sequence beyond the geographical positions in check-ins, which can be described as $\boldsymbol{Tr}_{sem} = \{(t_{v_1}, d_{v_1}, c_{v_1}), ..., (t_{v_L}, d_{v_L}, c_{v_L})\}$.

Formally, given a user u at time $t_{v_{l-1}}$ with her current position $p_{v_{l-1}}$ and her historical check-in trajectory Tr, our aim is to recommend top-k POIs $P_{rec} = \{p^1_{rec}, p^2_{rec}, ..., p^k_{rec} \in \mathcal{P}\}$ ($\mathcal{P}$ is the set of all POIs in the LBSNs), such that P_{rec} is most likely to be visited by the user at the next timestamp t_{v_l}. In order to achieve this goal, we first learn user's geographically constrained preference and intentional preference as well as the interplay between them from POI trajectory $\boldsymbol{Tr}_{poi}$ and semantic trajectory $\boldsymbol{Tr}_{sem}$, respectively, for each sub-trajectory. Thereafter, we consider the long- and short-term preference by modeling the long- and short-term trajectories $\boldsymbol{Tr}_{long}$ and $\boldsymbol{Tr}_{short}$, dependent upon the learned POI and semantic features, before performing the final POI recommendation.

4 HiLS Design

4.1 Overview of HiLS

HiLS aims to learn the long- and short-term preference from user's POI and semantic level trajectories for next POI recommendation. To this end, HiLS mainly involves the following three steps (c.f. Fig. 6):

(1) **Trajectory modeling with hierarchical LSTM**: For each user, we split the check-in records into multiple sub-trajectories. Each influential factor of the check-in record is embedded into a low-dimensional space, which retains

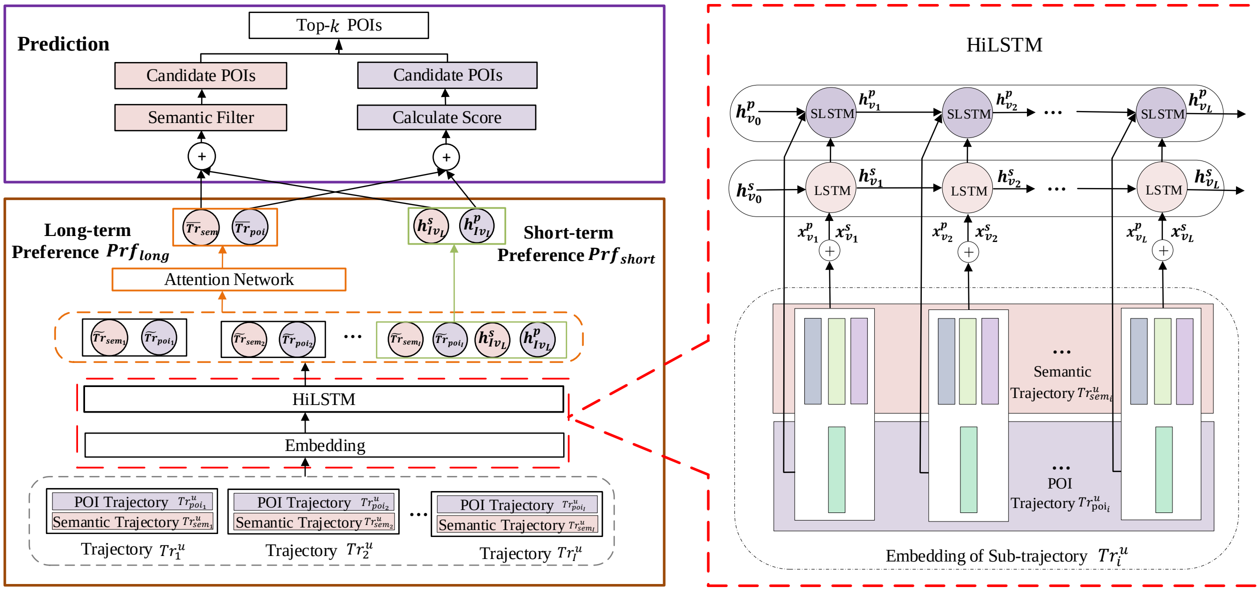


Fig. 6: Overall framework of HiLS. Short-term preference is learned from the current sub-trajectory and long-term preference is learned from historical sub-trajectories

the sequential relations among factors. The check-in embedding, or the concatenation of factors embedding, is then fed into a hierarchical LSTM (HiLSTM) model for learning the sub-trajectory representation and the hidden state of the latest check-in in both POI and semantic levels. The structure of HiLSTM is shown in the right part of Fig. 6.

(2) **Long- and short-term preference learning**: As shown in the bottom left of Fig. 6, given the learned POI and semantic representations, we next learn the long- and short-term preference from POI and semantic levels to preserve the general taste and the status at present. To thoroughly learn the long-term preference, we explore the correlations between historical and current trajectories with attention mechanism. Meanwhile, we utilize the short-term trajectory to capture the check-in sequential correlation with HiLSTM.

(3) **Next POI recommendation**: We get semantic and POI level preference for each user by comprehensively considering the long-term and short-term trajectories in two levels. Then, as shown in the upper left of Fig. 6, the semantic level preference is used for constructing the semantic filter. With the semantic filter, we are able to recommend POIs with suitable categories, even if the user checks in at POIs that have never been visited, and thus improving the recommendation effectiveness. The POI level preference will generate the final recommendation.

4.2 Trajectory Modeling with Hierarchical LSTM

As stated before, POI and semantic level features respectively reflect actual check-in POIs and the user intent, and they have different impacts for the next POI recommendation. Meanwhile, the interplay between POI and semantic trajectories is also significant as the next POI visit is affected by the semantic level intent subject to the actual geographical constraint. So, the first step of HiLS is to learn user preference in POI level and semantic level while considering their interplay. To this end, we design the HiLSTM to learn the features from POI and semantic trajectories.

4.2.1 *Semantic Trajectory Modeling*

The semantic trajectory modeling takes the embedding of semantic trajectory $\boldsymbol{Tr}_{sem}$ as input. The influential factors in semantic trajectory are inseparable as they jointly reflect the user's intentional preference. We use LSTM to capture the complicated sequential correlations or long-range dependencies contained in the sub-trajectory. In the hidden layer, each hidden vector $\boldsymbol{h}_{v_t}^s$ is updated after receiving the current input $\boldsymbol{x}_{v_t}^s$ and the last cell state $\boldsymbol{h}_{v_{t-1}}^s$. In LSTM, we have updates as follows:

$$\boldsymbol{x}_{v_t}^s = [\boldsymbol{e}_t, \boldsymbol{e}_d, \boldsymbol{e}_c], \quad 1 \le t \le L \tag{7}$$

$$\boldsymbol{h}_{v_t}^s = \mathrm{LSTM}(\boldsymbol{x}_{v_t}^s, \boldsymbol{h}_{v_{t-1}}^s), \quad \boldsymbol{h}_{v_0}^s = \boldsymbol{0} \tag{8}$$

where $\mathrm{LSTM}(\cdot)$ denotes one step pass of vector via LSTM; $\boldsymbol{e}_t$, $\boldsymbol{e}_d$, $\boldsymbol{e}_c$ respectively represent the embeddings of time slot t, geographical distance between the current and last POI d, and the category c. $[\cdot]$ denotes the concatenation operation of embeddings (reflected by $\oplus$ in Fig. 6), and the concatenation result $\boldsymbol{x}_{v_t}^s$ is fed for semantic trajectory modeling. $\boldsymbol{h}_{v_L}^s$ is the latest hidden state. The embeddings of influencing factors are randomly initialized and will be trained in the network. The representation learned from the semantic trajectory reflects general intentional preference.

4.2.2 *POI Trajectory Modeling*

To model POI level trajectory, we input the embedding of POI trajectory $\boldsymbol{Tr}_{poi}$ into HiLSTM. The interplay between POI level and semantic level trajectories is significant as the next POI visit is affected by the semantic level intent, so the key point is how to incorporate semantic intent in every update step. To tackle this issue, we input the features of semantic level hidden states which reflect the user intent at the moment in each step into POI level learning. By doing so, we can model the geographical related POI trajectories while considering the impact of user intent. As shown in the right part of Fig. 6, the update unit for POI trajectory modeling in HiLSTM is named SLSTM, which introduces user intent into POI trajectories modeling. Formally, we propose to add semantic factors into gates of LSTM and the update equations for SLSTM are as follows:

$$\boldsymbol{i}_{v_t} = \sigma(\boldsymbol{W}_{ix}\boldsymbol{x}^p_{v_t} + \boldsymbol{W}_{ih}\boldsymbol{h}^p_{v_{t-1}} + \boldsymbol{W}_{is}F(\boldsymbol{h}^s_{v_{t-1}}) + \boldsymbol{b}_i) \tag{9}$$

$$\boldsymbol{f}_{v_t} = \sigma(\boldsymbol{W}_{fx}\boldsymbol{x}^p_{v_t} + \boldsymbol{W}_{fh}\boldsymbol{h}^p_{v_{t-1}} + \boldsymbol{W}_{fs}F(\boldsymbol{h}^s_{v_{t-1}}) + \boldsymbol{b}_f) \tag{10}$$

$$\boldsymbol{o}_{v_t} = \sigma(\boldsymbol{W}_{ox}\boldsymbol{x}^p_{v_t} + \boldsymbol{W}_{oh}\boldsymbol{h}^p_{v_{t-1}} + \boldsymbol{W}_{os}F(\boldsymbol{h}^s_{v_{t-1}}) + \boldsymbol{b}_o) \tag{11}$$

where $\boldsymbol{h}^s_{v_{t-1}}$ is feature representing user intent of previous state, $\boldsymbol{x}^p_{v_t}$ is the current input embedding of the POI trajectory, $\boldsymbol{h}^p_{v_{t-1}}$ is the previous cell output hidden unit, $\boldsymbol{W}$ is the weight matrix, $\boldsymbol{b}$ is the bias term, and $F(.)$ is a function for semantic factor on POI level modeling defined as:

$$F(\boldsymbol{h}^s_{v_{t-1}}) = \boldsymbol{T}_s\boldsymbol{h}^s_{v_{t-1}} \tag{12}$$

where $\boldsymbol{T}_s$ is the linear transition matrix with respect to the semantic factors.

For POI trajectory modeling, we have updates as:

$$\boldsymbol{x}^p_{v_t} = [\boldsymbol{e}_p], \quad 1 \le t \le L \tag{13}$$

$$\boldsymbol{h}^p_{v_t} = \text{SLSTM}(\boldsymbol{x}^p_{v_t}, \boldsymbol{h}^s_{v_{t-1}}, \boldsymbol{h}^p_{v_{t-1}}), \quad \boldsymbol{h}^p_0 = \boldsymbol{0} \tag{14}$$

where SLSTM$(\cdot)$ denotes one step pass of vector in the POI trajectory, $\boldsymbol{e}_p$ represents the low-dimensional embedding of POI, and $\boldsymbol{h}^p_{v_L}$ is the latest hidden state of the current POI trajectory. The learned features for POI trajectories reflect the user's geographical related check-in preference.

HiLSM consists of LSTM for semantic trajectory modeling and SLSTM for POI trajectory modeling, and meanwhile considering their interplay. The status of HiLSTM at a specific step can be represented by the hidden states $\boldsymbol{h}^p_{v_t}$, $\boldsymbol{h}^s_{v_t}$ of cell. Considering all check-ins in a sub-trajectory, the representations for POI and semantic trajectories are respectively defined as:

$$\widetilde{\boldsymbol{Tr}}_{poi} = \frac{1}{L}\sum_{t=1}^{L}\boldsymbol{h}^p_{v_t}, \quad \widetilde{\boldsymbol{Tr}}_{sem} = \frac{1}{L}\sum_{t=1}^{L}\boldsymbol{h}^s_{v_t} \tag{15}$$

where $\boldsymbol{h}^p_{v_t}$ and $\boldsymbol{h}^s_{v_t}$ are the hidden state of each step in sub-trajectories for POI level and semantic level features; $\widetilde{\boldsymbol{Tr}}_{poi}$ and $\widetilde{\boldsymbol{Tr}}_{sem}$ are the representations for POI and semantic trajectories, respectively.

Overall, the whole process to learn both semantic and POI level features while considering their interplay by HiLSTM for each sub-trajectory is:

$$\widetilde{\boldsymbol{Tr}}_{poi}, \widetilde{\boldsymbol{Tr}}_{sem}, \boldsymbol{h}^s_{v_L}, \boldsymbol{h}^p_{v_L} = \text{HiLSTM}(\boldsymbol{Tr}_{sem}, \boldsymbol{Tr}_{poi}) \tag{16}$$

where HiLSTM$(\cdot)$ denotes the learning for a sub-trajectory in two levels. Note that the output of HiLSTM includes representations for both the latest status and the sub-trajectory.

4.3 Long and Short-Term Preference Modeling

After we obtained the POI and semantic representations, we next learn the long- and short-term preference from POI and semantic levels to preserve the general taste and the status at present (c.f. Fig. 6). We learn the long-term preference from the long-term trajectory with specially designed attention mechanism [7], [8], [17]. Meanwhile, we utilize the short-term trajectory to capture the short-term preference reflecting sequential correlation. HiLSTM learns POI and semantic levels features for each sub-trajectory, and thus different from existing efforts, we learn long and short-term preference in these two levels.

4.3.1 *Short-Term Preference Modeling*

Given multiple sub-trajectories $Tr = \{\boldsymbol{Tr}_1, \boldsymbol{Tr}_2 ..., \boldsymbol{Tr}_I\}$ of a user, the latest check-in sub-trajectory $\boldsymbol{Tr}_{short} = \{\boldsymbol{Tr}_I\}$ is taken for learning the short-term preference. The hidden states of LSTM in each step encode the status at the moment of the mobility, and so we regard the latest hidden state as the short-term preference for next mobility prediction. Specifically, for short-term preference modeling, we get the latest semantic level and POI level hidden states $\boldsymbol{h}^s_{Iv_L}$ and $\boldsymbol{h}^p_{Iv_L}$ of the current sub-trajectory with HiLSTM as the short-term preference in both levels. Besides, we also get the trajectory representation $\widetilde{\boldsymbol{Tr}}_{poi}$ and $\widetilde{\boldsymbol{Tr}}_{sem}$ by Eq. (15) in these two levels. The trajectory representations are used for the attention layer in latter long-term preference modeling. Formally, we input $\boldsymbol{Tr}_{short}$ into the HiLSTM and the learned short-term preference for current status as follows:

$$\boldsymbol{Prf}_{short} = \boldsymbol{h}^s_{Iv_L}, \boldsymbol{h}^p_{Iv_L} \tag{17}$$

$$\widetilde{\boldsymbol{Tr}}_I = \widetilde{\boldsymbol{Tr}}_{poi_I}, \widetilde{\boldsymbol{Tr}}_{sem_I} \tag{18}$$

where $\boldsymbol{Prf}_{short}$ represents the short-term preference in POI level and semantic level, and $\widetilde{\boldsymbol{Tr}}_I$ represents the trajectory representation of the latest trajectories in two levels.

4.3.2 *Long-Term Preference Modeling*

The long-term preference modeling intends to mine the periodic trends captured from historical sub-trajectories or the long-term trajectory $\boldsymbol{Tr}_{long} = \{\boldsymbol{Tr}_1, \boldsymbol{Tr}_2 ..., \boldsymbol{Tr}_{I-1}\}$, which largely represents the personal general preference. We first input all sub-trajectories of Tr into HiLSTM to get the representations $\widetilde{\boldsymbol{Tr}}_{poi_i}$ and $\widetilde{\boldsymbol{Tr}}_{sem_i}$ in POI and semantic level for every sub-trajectory $\widetilde{\boldsymbol{Tr}}_i$, where i represents the i_{th} sub-trajectory. Formally, the exact value consists of $\left\{\widetilde{\boldsymbol{Tr}}_{poi_1}, \widetilde{\boldsymbol{Tr}}_{poi_2}, ..., \widetilde{\boldsymbol{Tr}}_{poi_{I-1}}\right\}$ and $\left\{\widetilde{\boldsymbol{Tr}}_{sem_1}, \widetilde{\boldsymbol{Tr}}_{sem_2}, ..., \widetilde{\boldsymbol{Tr}}_{sem_{I-1}}\right\}$. After getting the historical sub-trajectories representation of long-term trajectory, we further integrate representation with attention mechanism to get long-term preference. By using the attention mechanism, we can focus on the relevant historical trajectories selectively.

The learned representation for historical sub-trajectories and the latest sub-trajectory, which is used to select relevant historical sub-trajectories, are the input of attention layer. The attention computation [16], [28] for each historical sub-trajectory $\boldsymbol{Tr}_i$ is defined as:

$$a_i = \frac{Sim(\widetilde{\boldsymbol{Tr}}_i, \widetilde{\boldsymbol{Tr}}_I)}{\sum_{j=1}^{I-1} Sim(\widetilde{\boldsymbol{Tr}}_j, \widetilde{\boldsymbol{Tr}}_I)} \qquad (i = 1, 2, ..., I-1) \tag{19}$$

where $Sim(\widetilde{\boldsymbol{Tr}}_i, \widetilde{\boldsymbol{Tr}}_I) = \widetilde{\boldsymbol{Tr}}_i \cdot \widetilde{\boldsymbol{Tr}}_I^{\mathrm{T}}$ calculates the relatedness of the i_{th} trajectory $\boldsymbol{Tr}_i$ and the latest current trajectory $\boldsymbol{Tr}_I$, and $\sum_{j=1}^{I-1} Sim(\widetilde{\boldsymbol{Tr}}_j, \widetilde{\boldsymbol{Tr}}_I)$ is used for normalization. After obtaining the attention coefficients a_i, we compute the long-term

Algorithm 1 HiLS for POI recommendation

Input: POI trajectory: $\left\{\boldsymbol{Tr}_{poi}^{u_1}, \boldsymbol{Tr}_{poi}^{u_2}, ...\right\}$; Semantic trajectory: $\{\boldsymbol{Tr}_{sem}^{u_1}, \boldsymbol{Tr}_{sem}^{u_2}, ...\}$
Output: Top-k POIs for each user

Divide the trajectories into training set and testing set
Train the model $\mathcal{H}$ by Algorithm 2 with training set
while $u \in \{u_1, u_2, ...\}$ **do**
  while each p in testing set **do**
    Calculate category score by Eq. (23) and filter candidate POIs
    Calculate POI score by Eq. (22) and recommend top-k POIs for user u
    Update θ by minimizing the objective function L

Algorithm 2 Training algorithm for HiLS

Input: POI trajectory for training: $\left\{\boldsymbol{Tr}_{poi}^{u_1}, \boldsymbol{Tr}_{poi}^{u_2}, ...\right\}$; Semantic trajectory for training: $\{\boldsymbol{Tr}_{sem}^{u_1}, \boldsymbol{Tr}_{sem}^{u_2}, ...\}$
Output: Trained Model $\mathcal{H}$

Initialize the embeddings of trajectories
Initialize the parameters
while $u \in \{u_1, u_2, ...\}$ **do**
  while each $\boldsymbol{Tr}_{poi_i}$ and $\boldsymbol{Tr}_{sem_i}$ **do**
    Put $\boldsymbol{Tr}_{poi_i}$ and $\boldsymbol{Tr}_{sem_i}$ into HiLSTM.
    Get the representation for $\widetilde{\boldsymbol{Tr}}_{poi_i}$, $\widetilde{\boldsymbol{Tr}}_{sem_i}$ and the last hidden states $\boldsymbol{h}_{lv_L}^{s}$, $\boldsymbol{h}_{lv_L}^{p}$
    Get the user preference by Eq. (21)
    Update θ by minimizing the objective function L
Output the trained model $\mathcal{H}$

preference $\boldsymbol{Prf}_{long}$, a sum of all historical sub-trajectories, as:

$$\boldsymbol{Prf}_{long} = \sum_{i=1}^{I-1} a_i \widetilde{\boldsymbol{Tr}}_i \tag{20}$$

It is noticed that, each sub-trajectory $\boldsymbol{Tr}_i$ actually consists of two-level representations $\boldsymbol{Tr}_{poi_i}$ and $\boldsymbol{Tr}_{sem_i}$, so the long-term preference $\boldsymbol{Prf}_{long}$ also contains two levels (c.f. Fig. 6). Formally, $\boldsymbol{Prf}_{long} = \overline{\boldsymbol{Tr}}_{poi}, \overline{\boldsymbol{Tr}}_{sem}$. After we get the personal long-term and short-term preference, they are merged as the final preference. We draw a weight calculation to obtain the final preference:

$$\boldsymbol{Prf} = \boldsymbol{W}_{short}\boldsymbol{Prf}_{short} \oplus \boldsymbol{W}_{long}\boldsymbol{Prf}_{long} \tag{21}$$

where $\boldsymbol{W}_{short}$ and $\boldsymbol{W}_{long}$ are weighted matrixes for long and short-term modeling, respectively, and they are learned automatically without pre-specifying in the experiments. $\oplus$ is the concatenation of $\boldsymbol{W}_{short}\boldsymbol{Prf}_{short}$ and $\boldsymbol{W}_{long}\boldsymbol{Prf}_{long}$. Both $\boldsymbol{Prf}_{short}$ and $\boldsymbol{Prf}_{long}$ involve semantic and POI level information, so $\boldsymbol{Prf} = \boldsymbol{Prf}_{poi}, \boldsymbol{Prf}_{sem}$, described by Eq. (21).

4.4 POI Recommendation with HiLS

Before generating the final recommendation list, as shown in Algorithm 1, we first filter out candidate POIs most likely to be selected by users. Inspired by the existing work [15], where they construct category filters to improve recommendation, we make the best of semantic level preference to construct a semantic filter. With the semantic level features which reflect the user intent, we further construct the semantic filter to preliminarily filter out POIs that are in consist with user intent before recommendation. In this way, the semantic level features will be made better use of to help improve the recommendation effectiveness. Specifically, we preliminarily filter out POIs according to the check-in category by semantic features. After getting the candidate POIs, we rank the scores of all POIs for the final POIs list.

The aim of HiLS is to recommend the most likely to be visited POI by the user at the next time, and so we use a linear transition matrix to get the final score for each POI. To build the semantic filter, we adopt the same method to get the score for the next activity category. The calculation probability is as follows:

$$Score_{poi} = \text{softmax}(\boldsymbol{T}_{poi}\boldsymbol{Prf}_{poi}) \tag{22}$$

$$Score_{sem} = \text{softmax}(\boldsymbol{T}_{sem}\boldsymbol{Prf}_{sem}) \tag{23}$$

where $\boldsymbol{T}_{poi}$ and $\boldsymbol{T}_{sem}$ are the linear transition matrices for the final scores; $\boldsymbol{Prf}_{poi}$ and $\boldsymbol{Prf}_{sem}$ are respectively the POI level and semantic level preference for the user. Consequently, the recommended POI is the one with the largest probability to be visited by the user at the next step and so does the category filter. Given the check-in records of a user, the objective function is as follows:

$$L = -\sum_{m=1}^{M} \log(Score_{poi_m}) - \alpha \sum_{m=1}^{M} \log(Score_{sem_m}) \tag{24}$$

where the objective function is organized in log likelihood, M is the number for training samples of a user, $Score_{poi_m}$ and $Score_{sem_m}$ are respectively the recommendation probability for POIs and categories; α controls HiLS updating with the semantic level, and is set to 1 in our case to incorporate both POI and semantic level updates. In practice, we use Backward Propagation Through Time (BPTT) and Adam [30] to train it. Algorithm 1 describes both the training and testing procedures of HiLS. The raw check-ins of each user are preprocessed into POI sub-trajectories $\{\boldsymbol{Tr}_{poi}^{u_1}, \boldsymbol{Tr}_{poi}^{u_2}, ...\}$ and semantic sub-trajectories $\{\boldsymbol{Tr}_{sem}^{u_1}, \boldsymbol{Tr}_{sem}^{u_2}, ...\}$, and then divided into training set and testing set. Algorithm 2 describes the training algorithm for HiLS. To capture the personal preference, we first learn features for each sub-trajectory in POI and semantic levels meanwhile considering their interplay by HiLSTM. Then, the short-term and long-term preference are designed to preserve user current status and stable preference, and the final recommendation are their synthetical consideration.

5 Performance Evaluation

5.1 Experiment Setup

5.1.1 Datasets

We utilize two real-world datasets New York (NYC) and Tokyo (TKY) collected by [19] from Foursquare to evaluate the performance of our POI recommendation. The datasets are widely adopted in existing works [15]–[17], [31], and the details are described in Table 1. For each user u, we chronologically split the check-in data into two parts, where

TABLE 1: Statistic of Datasets

Dataset	#Users	#POIs	#Check-ins
New York	12,062	11,422	443,284
Tokyo	14,441	16,265	1,311,614

the first 70% for training, and the remaining 30% for testing. For validation, we pick each check-in in the testing dataset as the current POI and the next visit will be calculated by HiLS. The next check-in in the testing dataset is regarded as the ground truth. If one of the top-k POIs outputted by HiLS fits the ground truth, it is then regarded as successful.

5.1.2 Baselines

We compare HiLS with the following six methods.

- ***DeepMove*** [7]. DeepMove adopts two modules for preference learning, the current module captures the complicated sequential information in the current trajectory and the historical attention module chooses the most related trajectory history as the periodicity representation.
- ***PLSPL*** [17]. PLSPL considers personalized dependencies on long- and short-term preference for different users and integrates different influence of locations and categories for POI recommendation.
- ***ST-LSTM*** [32]. ST-LSTM combines spatial-temporal influence into LSTM to mitigate the problem of data sparsity.
- ***LSTPM*** [8]. LSTPM proposes a geo-dilated LSTM to exploit the geographical relations among non-consecutive POIs.
- ***STAN*** [33]. To learn the interaction between non-adjacent location and non-consecutive check-ins, STAN exploits relative spatio-temporal information of all check-ins with self-attention layer along the trajectories.

For all baselines, the parameter settings are initialized the same as reported in their original works. For DeepMove, we select historical attention module with sequential encode module which shows best results to capture periodicity.

5.1.3 Metrics

We use two standard recommendation evaluation metrics that are commonly adopted in existing studies [14], [15], [34], [35], namely *Recall@k (Rec@k)* and *NDCG@k*, to measure the performance of the next POI recommendation task. The former computes the ratio of true positive samples in all positive samples that the user is really interested:

$$Recall@k = \frac{1}{|\mathcal{U}|}\sum_{u\in\mathcal{U}}\frac{|R_u^k \cap T_u|}{|T_u|} \tag{25}$$

where R_u^k is the set of top-k next POIs in the recommendation list for user u, T_u is u's actually truth set of next POI, $\mathcal{U}$ is the set of users and $|\mathcal{U}|$ is the number of total users. The latter measures the quality of top-k list:

$$DCG@k = rel_1 + \sum_{i=2}^{k}\frac{rel_i}{\log(i+1)} \tag{26}$$

$$NDCG@k = \frac{DCG@k}{IDCG@k} \tag{27}$$

where rel_i is the relevance of the POI at position i, and k is the recommend list length, and is set k = 1, 5, 10 in our experiments. If the POI at position i is the ground truth next POI, $rel_i = 1$; otherwise, $rel_i = 0$. *DCG@*k evaluates the accuracy of sorting without the consideration of the recommendation list and the number of truly valid results, and *NDCG@k* is normalized *DCG@k*. *IDCG@k* is the maximum value of all possible recommendation list with length k. Compared with *Recall@k*, *NDCG@*k considers the position of true positive samples in top-k recommendation list. We compute the average performance of *Recall@k* and *NDCG@k* as the final values.

5.1.4 Experiment Settings

In our experiments, the learning rate is set to 0.001. All the parameters are optimized using the gradient descent optimization algorithm Adam with batch size of 32. According to check-in time, we evenly divide weekends and weekdays respectively into 24 portions as in previous works [8], [22], [34]. For α, we empirically set it to 1. All embedding dimensions of latent vector are set the same ($D_l = D_t = D_c = D_g$) for the two datasets. The hidden state and cell state are initialized as zero. To make fair comparison, the number of unit layer is set to 1 as in [8]. The hyper-parameters are empirically estimated and then determined by correlated experiments. The embedding dimension is set to 500. The time window width and pre-defined trajectory length are respectively $\Delta t = 72h$ and $L = 10$. The category number for constructing semantic filter is 100. All models are trained until convergence. We implement our method and baselines with PyTorch 1.9.1 on NVIDIA GeForce GTX 1050 Ti GPUs and Intel(R) Core(TM) i5-8300H CPUs.

5.2 Performance on Next POI Recommendation

We first conduct experiments to evaluate the performance of our HiLS with state-of-the-art methods, and the results are shown in Table 2. The parameter settings are exactly the same for HiLS and other baselines. In each column, the optimal (resp. sub-optimal) preference is bold (resp. marked with $*$). The *Rec*@1 and *NDCG*@1 have the same value, so we only show one. From the statistics, we can observe that the proposed HiLS achieves better performance than all baselines in terms of every metrics on both datasets, and LSTPM is the sub-optimal methods. As shown in Table 2, HiLS has an improvement of 6.35% and 5.06% in terms of *Rec*@5 and *NDCG*@5 compared to LSTPM on TKY dataset, which demonstrates the effectiveness of HiLSTM in our model. The improvement mainly benefits from the effectively preference learning reflecting user intents and geographical constrained actual check-ins.

We can also observe from Table 2 that HiLS and LSTPM outperform DeepMove and PLSPL. The results show that the long-term preference deriving from historical check-ins plays an important role in capturing personal representation. DeepMove and PLSPL perform worse because when learning long-term preference, they use all history check-ins as a sequence. However, the amount of user historical check-ins is pretty large, and some information may get loss when directly learning such long sequences. We use attention module to learn the trajectory-level long-term preference,

TABLE 2: Performance of all the comparison methods on the two real-world datasets measured Rec@k and NDCG@k.

	NYC					TKY				
	$Rec@1$	$Rec@5$	$Rec@10$	$NDCG@5$	$NDCG@10$	$Rec@1$	$Rec@5$	$Rec@10$	$NDCG@5$	$NDCG@10$
DeepMove	*0.1820	0.4033	0.4734	0.2993	0.3222	0.1266	0.2737	0.3368	0.2037	0.2241
PLSPL	0.1648	0.3522	0.4178	0.2643	0.2856	0.1397	0.3204	0.3967	0.2347	0.2594
ST-LSTM	0.1724	0.3906	0.4672	0.2877	0.3127	*0.1602	0.3435	0.4159	0.2564	0.2799
LSTPM	0.1815	*0.4232	*0.5123	*0.3082	*0.3373	0.1597	*0.3482	*0.4213	*0.2587	*0.2824
STAN	0.0871	0.2633	0.3867	0.1765	0.2216	0.0971	0.2086	0.2856	0.1602	0.1922
HiLS	**0.1941**	**0.4348**	**0.5278**	**0.3211**	**0.3513**	**0.1626**	**0.3703**	**0.4562**	**0.2718**	**0.2996**

which avoids too long sequence and preserves the periodic pattern. LSTPM also designs a nonlocal network structure to learn the trajectory-level long-term preference, and so the performance is suboptimal. STAN performs worst of all competitors, mainly because the next POI prediction problem is highly related to the user current and recent status. STAN does not consider the time interval of the historical check-ins from the current position. Comparing with ST-LSTM, which combines spatial-temporal influence into LSTM, HiLS considers different influence of factors and their holistic effect. Therefore, HiLS achieves better results.

5.3 Performance of HiLS Variants for Ablation Study

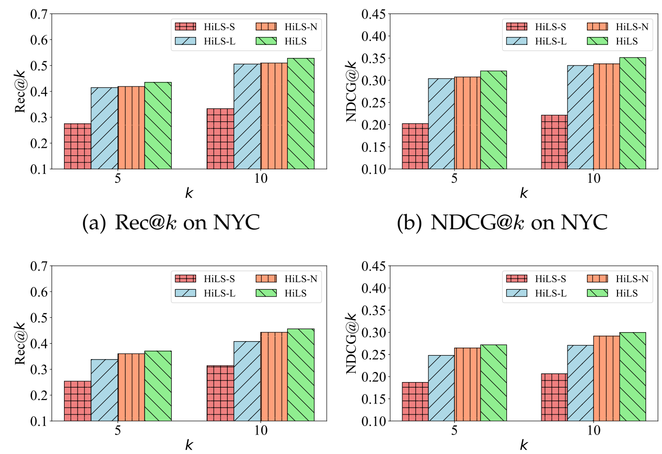
In order to verify the effectiveness of several key parts designed in HiLS, we further conduct ablation study by comparing some variants of our model as follows:

- ***HiLS-L***: This variant removes the short-term component of HiLS and keeps only the long-term component.
- ***HiLS-S***: This variant removes the long-term component of HiLS and only the short-term component is remained.
- ***HiLS-N***: This variant ignores the interplay between semantic level and POI level learning in HiLSTM module, and two completely independent LSTMs are used instead respectively for semantic level and POI level learning.
- ***HiLS-NH***: This variant ignores the semantic level and POI level learning in HiLSTM module and only one LSTM is used instead of HiLSTM, to model all influencing factors.

5.3.1 Effect of Long- and Short-term Components

We show the performance of HiLS-L and HiLS-S in Fig. 7. It can be observed that, HiLS-L outperforms HiLS-S. This is because short-term preference reflects user's current state derived from current sub-trajectory and only reflects the consecutiveness of current visits, while long-term preference reflects the general taste of the user and considers the personality. Furthermore, HiLS shows the best result, which demonstrates the effectiveness of considering both long- and short- term preference.

The attention mechanism is used for long-term preference modeling. The visualization of relevance representing attention level on POI and semantic sub-trajectories is shown in Fig. 8. It can be observed that the attention distributions of POI and semantic sub-trajectories are different. Fig. 8(a) describes the attention at POI level, and shows that



(a) Rec@k on NYC (b) NDCG@k on NYC

(c) Rec@k on TKY (d) NDCG@k on TKY

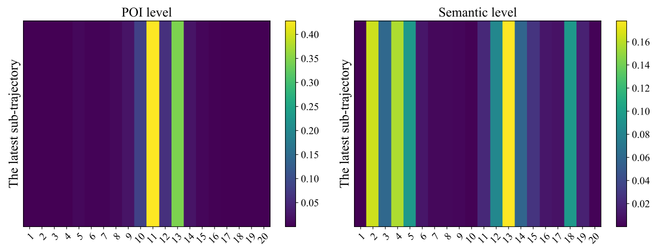
Fig. 7: Performance of HiLS variants for ablation study.

the 10th and 13th historical POI sub-trajectories are mostly related to the current sub-trajectory, which indicates that the user's current sub-trajectory overlaps more with the POIs of these two sub-trajectories. Compared with the attention at POI level, the attention at the semantic level is associated with more sub-trajectories. The 2nd, 4th and 13th historical semantic sub-trajectories are mostly related to the current one. Sub-trajectories not related at the POI level are likely to be related at the semantic level. Therefore, more historical sub-trajectories can be associated at the semantic level. With semantic level features, recommendations can be effective even when users check in at POIs they never visit before.

5.3.2 Effect of Interplay of two-level Representations

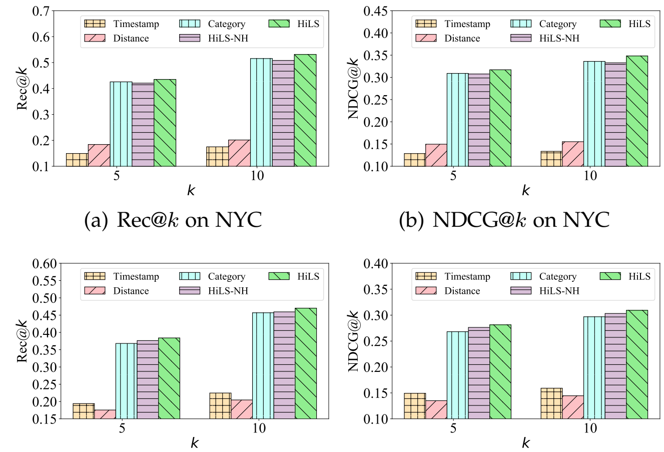
A key point of our model lies in the consideration of the interplay between POI level and semantic level representation. In this section, we examine the effectiveness of HiLSTM which considers the interplay of user intends and actual check-ins that constrained by geographical location. The trajectory learning process in variant HiLS-N is designed to two parallel LSTMs. They are completely independent, and other components in HiLS-N and HiLS are the same. We also learn semantic trajectory in this variant, as it will be used for the category filter.

From the results in Fig. 7, it is clear that the proposed HiLSTM structure yields better performance than purely parallel LSTM structure. This verifies the effectiveness of modeling the interplay of two-level representations. Comparing with HiLS-N, HiLS gains performance increase of



(a) POI level (b) Semantic level

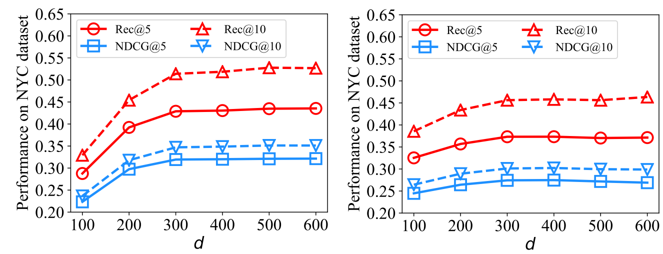
Fig. 8: Visualization of relevance for sub-trajectories.



(a) Rec@k on NYC (b) NDCG@k on NYC

(c) Rec@k on TKY (d) NDCG@k on TKY

Fig. 9: Performance of each factor.



(a) Preference on NYC (b) Preference on TKY

Fig. 10: Effect of the embedding dimension size d.

5.26% and 7.90% respectively on NYC and TKY datasets on *NDCG*@5. Since we consider the interplay between semantic and POI level modeling, when user's check-in tendency is not obvious in POI level, we are more likely to recommend suitable POIs to the user. For example, if a user always checks in at p_3 after consecutively at p_1 and p_2, p_3 will be mostly recommended when he visits the same POIs with POI level modelling. However, when the user moves to a new place or dislikes to check in at the same POI for many times, in this case, our proposed HiLS will perform better.

5.3.3 *Effect of Each Factor on Next POI Recommendation*

The next POI recommendation is influenced by multiple factors. To investigate the contributions of each factor for semantic level feature learning, we conduct a comparative study for each factor. We first examine each factor for next POI recommendation, and then input all factors including semantic level and POI level factors into the same model for experiments.

From the results in Fig. 9, it can be observed that category is the main contribution for semantic level features, while distance or timestamp alone will lead to the performance decrease. Users have their preference after checking in at a specific POI and the category of POI will reflect meaning information of preference; however, purely timestamp and geographical distance may bring some misleading information for semantic level feature learning. HiLS combining with timestamp, category and distance as semantic level influencing factors brings the best results. Comparing with the variant HiLS-N which includes only POI level features, timestamp and distance will degrade the performance in expectation. Furthermore, HiLS-NH shows the results of inputting all factors into the same model with long and short-term preference structure, but HiLS with semantic level features and POI level feature performs better for considering the interactions of factors that some of them are related and some are related to their joint effects.

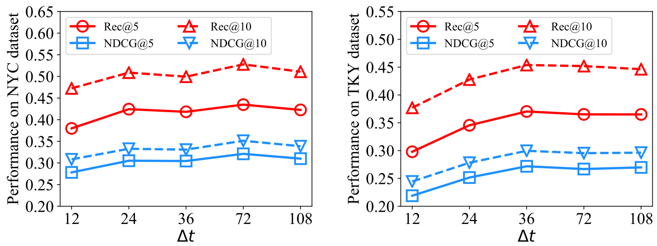
5.4 Evaluation on Parameter Sensitivity

5.4.1 *Effect of Embedding Dimension*

In this section, we show the effect of dimension size on the performance of HiLS. The dimension of hidden units and latent vectors are set to be the same. Usually, a larger value suggests a stronger expression ability, though too large values may result in over-fitting. We tune the size d from 100 to 600 with a step of 100. The results of HiLS with different embedding dimensions regarding to *Rec@k* and *NDCG@k* are shown in Fig. 10. It can be observed that the performance gradually increases with the size of embedding dimensions increasing and converges after more than 300. Then the performance increases slightly when d reaches 500.

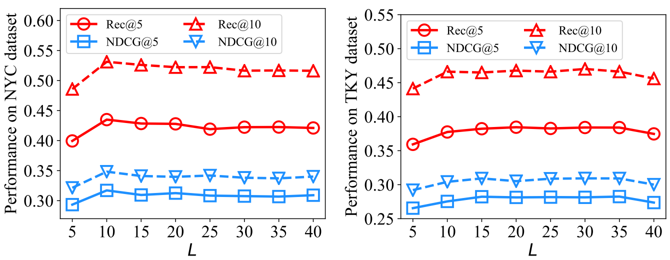
5.4.2 *Effect of Time Window and Sub-trajectory Length*

During data preprocessing, we split historical check-ins into multiple sub-trajectories by the time interval Δt and predefined trajectory length L which both affects the partitioning for meaningful sub-trajectories. We tune the time window width for optimal partitioning and set the time interval from $12h$ to $108h$. As shown in Fig. 11, the best results are achieved with $\Delta t = 72h$ almost in all metrics in NYC dataset. A very smaller time window will result in many individual check-ins and thus leads to lots of consecutiveness information loss. However, the performance deteriorates when Δt increases more than $72h$, and the time interval exceeds too much such that wrong transition information is introduced. The results of the sub-trajectory length with varying L is illustrated in Fig. 12. We can see that the performance gets better as the length increases from 5 and then gradually gets stable when more than 10 in both datasets. Besides, we can see an obvious drop in TKY dataset when $L > 35$, which means the sub-trajectory contains too many discontinuous check-ins.



(a) Preference on NYC (b) Preference on TKY

Fig. 11: Effect of the time window width Δt.



(a) Preference on NYC (b) Preference on TKY

Fig. 12: Effect of the trajectory length L.

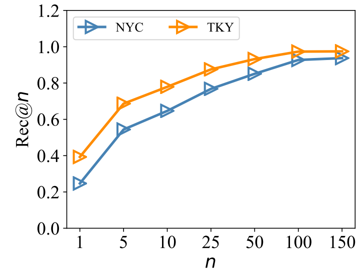
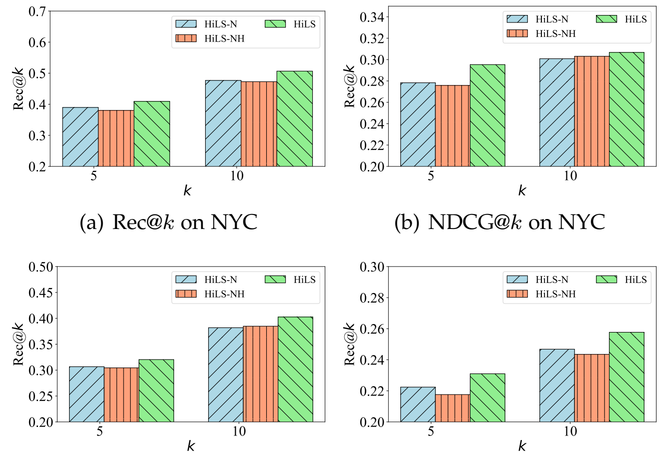


Fig. 13: Effect of the number of categories n.



(a) Rec@k on NYC (b) NDCG@k on NYC

(c) Rec@k on TKY (d) NDCG@k on TKY

Fig. 14: Performance of next new POI recommendation.

5.4.3 *Effect of Category Number*

We use the category filter to pre-filter categories first before giving the POI recommendation. Therefore, we conduct experiment on two datasets to examine the performance on next category recommendation. We use the same metric *Rec@k* as next POI recommendation to evaluate the effectiveness. It can be seen in Fig. 13, the recall of predicted top-k categories reach more than 90% on NYC dataset and even 97% for TKY dataset. Most POIs have been included when k reaches 100. To ensure the effectiveness of the category filter and avoid the loss, the number of categories for the category filter is therefore suggested to be 100.

5.5 Performance on Raw POI Recommendation

The users always check in at new POIs that have never been visited before. To further verify the effectiveness of incorporating the interaction of influencing factors, we evaluate the performance of HiLS for next POI recommendation in the scenario of users currently visiting raw POIs, specifically the POIs that have never been visited before. As shown in Fig. 14, HiLS and all variants show effectiveness as the consecutiveness of POIs is learned by short-term modeling and the next POIs to be recommended is likely to be ones that has already been visited. First, we compare HiLS with HiLS-N, the performance of HiLS is consistently better than HiLS-N, which demonstrates the effectiveness of semantic level features for next visit prediction. Then we compare HiLS with HiLS-NH which inputs all factors into the same model, we can see that the performance of HiLS is better than HiLS-NH, as considering the correlation and interaction of factors will help preserve more useful information. Compared with HiLS-N, which considers only one key factor, directly inputting all factors into the same model in HiLS-NH will even deteriorate the performance. Obviously, it is necessary to consider the different influence of factors and their holistic effect to improve the performance.

6 Related Work

6.1 Next POI Recommendation

Next POI recommendation is highly related to the sequential pattern of users. Early works [36], [37] on sequential recommendation usually model the sequential influence by Markov chain. For example, Cheng et al. [36] exploit the personalized Markov chain in the check-in trajectory. FPMC [37] predicts the next action based on personalized transition graphs over underlying Markov chains.

RNN and its variants like LSTM have achieved great success to deal with sequential data, so they are more widely used to characterize the sequential feature of trajectories recently. To incorporate more influential factors, existing studies use different methods to consider them. ST-RNN [38] incorporates temporal and spatial contexts with time-specific matrices in RNN. STGN [39] introduces two pairs of time gate and distance gate to control the updates. ST-LSTM [32] introduces spatial-temporal relations into LSTM and further proposes a hierarchical extension HST-LSTM in an encoder-decoder manner. LSTPM [8] considers geographical relations among recently visited POIs by geo-dilated LSTM. ASPPA [35] can automatically identify each semantic sub-trajectories by an attention-based stacked

RNN. ARNN [27] seamlessly integrates RNN and attention mechanism in a unified framework. DMAN [28] derives dynamic memory-based attention network and recurrent attention network for modeling dynamic preference. RCR [31] designs a gated recurrent units structure to learn latent representation. CHA [37] explores the category hierarchy knowledge graph of POIs to learn robust representations.

Different from these studies, our model captures both POI and semantic level features which have different impacts for next POI recommendation while considering the interplay between them. In this way, our method considers the user intent free from geographical constrain and actual check-in POIs in the physical world. Therefore, we are more likely to recommend suitable POIs to the user when the user check-in tendency is not that distinct in the POI level.

6.2 Long and Short-term Preference Modeling

The long-term preference reflects the general taste of users, while the short-term behaviors reflect the recent preference of user's current state. Therefore, it is necessary to consider both for next POI recommendation. DeepMove [7] adopts two modules for preference learning: the current module captures the complicated sequential information in the current trajectory and the historical attention module chooses the most related trajectory history as the periodicity representation. HOPE [40] adapts LSTM for in out-of-town short-term preference modeling and asymmetric-SVD for long-term preference modeling. LSPL [16] uses attention mechanism to learn user long-term preference. Their extension work [17] considers the personalized weights on different parts for different users. LSTTM [29] builds two graphs the global long-term graph and internal short-term graph for online recommendation. DeepMove regards all historical check-ins as a trajectory; however, a too long trajectory is difficult to learn and not conducive to the expression of trajectory. So, the recent work LSTPM [8] divides the historical check-ins into multiple trajectories and uses non-local network to capture the influence of each historical trajectory on the current one.

In our method, we jointly learn user long- and short-term preference on POI level and semantic level for preserving multi-level check-in patterns. Besides, we learn the long-term preference from historical sub-trajectories by attention mechanism with the latest visited sub-trajectory which effectively selects the useful historical information.

7 Conclusion

In this paper, we have presented a novel framework for long- and short-term preference learning for next POI recommendation. To consider the interplay of the POI and semantic level features, we design a hierarchical LSTM to guide the learning process, where the long-term preference is learned by exploring the correlations between historical and current trajectories with attention mechanism, while the short-term trajectory is utilized to capture the check-in sequential correlation with the hierarchical LSTM. Experimental results on two real-world datasets demonstrate the effectiveness and superiority of the proposed framework. We plan to delve deeper into understanding the distinct impacts of POI and semantic levels on current visits, thereby improving the approach to attention calculation during the learning process of long-term preferences.

References

[1] H. Jiang, J. Li, P. Zhao, F. Zeng, Z. Xiao, and A. Iyengar, "Location privacy-preserving mechanisms in location-based services: A comprehensive survey," *ACM Computing Surveys*, vol. 54, no. 1, pp. 1–36, 2021.

[2] K. Huguenin, I. Bilogrevic, J. S. Machado, S. Mihaila, R. Shokri, I. Dacosta, and J.-P. Hubaux, "A predictive model for user motivation and utility implications of privacy-protection mechanisms in location check-ins," *IEEE Transactions on Mobile Computing*, vol. 17, no. 4, pp. 760–774, 2018.

[3] X. Ma, F. Ding, K. Peng, Y. Yang, and C. Wang, "CP-Link: Exploiting continuous spatio-temporal check-in patterns for user identity linkage," *IEEE Transactions on Mobile Computing*, vol. 22, no. 8, pp. 4594–4606, 2023.

[4] P. Sánchez and A. Bellogín, "Point-of-interest recommender systems based on location-based social networks: A survey from an experimental perspective," *ACM Computing Surveys*, vol. 54, no. 11, pp. 223:1–223:37, 2022.

[5] Z. Li, W. Cheng, H. Xiao, W. Yu, H. Chen, and W. Wang, "You are what and where you are: Graph enhanced attention network for explainable POI recommendation," in *Proceedings of ACM CIKM*, 2021, pp. 3945–3954.

[6] X. Rao, L. Chen, Y. Liu, S. Shang, B. Yao, and P. Han, "Graph-flashback network for next location recommendation," in *Proceedings of ACM SIGKDD*, 2022, pp. 1463–1471.

[7] J. Feng, Y. Li, C. Zhang, F. Sun, F. Meng, A. Guo, and D. Jin, "DeepMove: Predicting human mobility with attentional recurrent networks," in *Proceedings of WWW*, 2018, pp. 1459–1468.

[8] K. Sun, T. Qian, T. Chen, Y. Liang, Q. V. H. Nguyen, and H. Yin, "Where to go next: Modeling long-and short-term user preferences for point-of-interest recommendation," in *Proceedings of AAAI*, 2020, pp. 214–221.

[9] Y. Lin, H. Wan, S. Guo, and Y. Lin, "Pre-training context and time aware location embeddings from spatial-temporal trajectories for user next location prediction," in *Proceedings of AAAI*, 2021, pp. 4241–4248.

[10] L. Zhang, Z. Sun, J. Zhang, Y. Lei, C. Li, Z. Wu, H. Kloeden, and F. Klanner, "An interactive multi-task learning framework for next POI recommendation with uncertain check-ins," in *Proceedings of IJCAI*, 2021, pp. 3551–3557.

[11] L. Huang, Y. Ma, S. Wang, and Y. Liu, "An attention-based spatiotemporal LSTM network for next POI recommendation," *IEEE Transactions on Services Computing*, vol. 14, no. 6, pp. 1585–1597, 2019.

[12] S. Yang, J. Liu, and K. Zhao, "GETNext: Trajectory flow map enhanced transformer for next POI recommendation," in *Proceedings of ACM SIGIR*, 2022, pp. 1144–1153.

[13] X. F. S. W. J. Y. Xiaolin Wang, Guohao Sun, "Modeling spatio-temporal neighbourhood for personalized point-of-interest recommendation," in *Proceedings of IJCAI*, 2022, pp. 3530–3536.

[14] Z. Dong, X. Meng, and Y. Zhang, "Exploiting category-level multiple characteristics for POI recommendation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 2, pp. 1488–1501, 2023.

[15] F. Yu, L. Cui, W. Guo, X. Lu, Q. Li, and H. Lu, "A category-aware deep model for successive POI recommendation on sparse check-in data," in *Proceedings of WWW*, 2020, pp. 1264–1274.

[16] Y. Wu, K. Li, G. Zhao, and X. Qian, "Long-and short-term preference learning for next POI recommendation," in *Proceedings of CIKM*, 2019, pp. 2301–2304.

[17] Y. Wu, K. Li, G. Zhao, and Q. Xueming, "Personalized long-and short-term preference learning for next POI recommendation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 4, pp. 1944–1957, 2020.

[18] H. A. Rahmani, M. Naghiaei, A. Tourani, and Y. Deldjoo, "Exploring the impact of temporal bias in point-of-interest recommendation," in *Proceedings of ACM RecSys*, 2022, pp. 598–603.

[19] D. Yang, D. Zhang, V. W. Zheng, and Z. Yu, "Modeling user activity preference by leveraging user spatial temporal characteristics in LBSNs," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 45, no. 1, pp. 129–142, 2014.

[20] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[21] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proceedings of the National Academy of Sciences*, vol. 79, no. 8, pp. 2554–2558, 1982.

[22] D. Yao, C. Zhang, J. Huang, and J. Bi, "Serm: A recurrent model for next location prediction in semantic trajectories," in *Proceedings of CIKM*, 2017, pp. 2411–2414.

[23] Y. Yu, X. Si, C. Hu, and J. Zhang, "A review of recurrent neural networks: LSTM cells and network architectures," *Neural Computation*, vol. 31, no. 7, pp. 1235–1270, 2019.

[24] Z. Xiao, H. Fang, H. Jiang, J. Bai, V. Havyarimana, H. Chen, and L. Jiao, "Understanding private car aggregation effect via spatio-temporal analysis of trajectory data," *IEEE Transactions on Cybernetics*, vol. 53, no. 4, pp. 2346–2357, 2023.

[25] W. Long, T. Li, Z. Xiao, D. Wang, R. Zhang, A. C. Regan, H. Chen, and Y. Zhu, "Location prediction for individual vehicles via exploiting travel regularity and preference," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 5, pp. 4718–4732, 2022.

[26] J. Li, F. Zeng, Z. Xiao, Z. Zheng, H. Jiang, and Z. Li, "Social relationship inference over private vehicle mobility data," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 6, pp. 5221–5233, 2021.

[27] Q. Guo, Z. Sun, J. Zhang, and Y.-L. Theng, "An attentional recurrent neural network for personalized next location recommendation," in *Proceedings of AAAI*, 2020, pp. 83–90.

[28] Q. Tan, J. Zhang, N. Liu, X. Huang, H. Yang, J. Zhou, and X. Hu, "Dynamic memory based attention network for sequential recommendation," in *Proceedings of AAAI*, 2021, pp. 4384–4392.

[29] R. Xie, Y. Wang, R. Wang, Y. Lu, Y. Zou, F. Xia, and L. Lin, "Long short-term temporal meta-learning in online recommendation," in *Proceedings of WSDM*, 2022, pp. 1168–1176.

[30] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[31] G. Yang, Y. Cai, and C. K. Reddy, "Spatio-temporal check-in time prediction with recurrent neural network based survival analysis," in *Proceedings of IJCAI*, 2018, pp. 2976–2983.

[32] D. Kong and F. Wu, "HST-LSTM: A hierarchical spatial-temporal long-short term memory network for location prediction." in *Proceedings of IJCAI*, 2018, pp. 2341–2347.

[33] Y. Luo, Q. Liu, and Z. Liu, "STAN: Spatio-temporal attention network for next location recommendation," in *Proceedings of WWW*, 2021, pp. 2177–2185.

[34] H. Sun, J. Xu, K. Zheng, P. Zhao, P. Chao, and X. Zhou, "MFNP: A meta-optimized model for few-shot next POI recommendation," in *Proceedings of IJCAI*, 2021, pp. 3017–3023.

[35] K. Zhao, Y. Zhang, H. Yin, J. Wang, K. Zheng, X. Zhou, and C. Xing, "Discovering subsequence patterns for next POI recommendation." in *Proceedings of IJCAI*, 2020, pp. 3216–3222.

[36] C. Cheng, H. Yang, M. R. Lyu, and I. King, "Where you like to go next: Successive point-of-interest recommendation," in *Proceedings of IJCAI*, 2013, pp. 2605–2611.

[37] S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme, "Factorizing personalized markov chains for next-basket recommendation," in *Proceedings of WWW*, 2010, pp. 811–820.

[38] Q. Liu, S. Wu, L. Wang, and T. Tan, "Predicting the next location: A recurrent model with spatial and temporal contexts," in *Proceedings of AAAI*, 2016, pp. 194–200.

[39] P. Zhao, H. Zhu, Y. Liu, J. Xu, Z. Li, F. Zhuang, V. S. Sheng, and X. Zhou, "Where to go next: A spatio-temporal gated network for next POI recommendation," in *Proceedings of AAAI*, 2019, pp. 5877–5884.

[40] H. Sun, J. Xu, R. Zhou, W. Chen, L. Zhao, and C. Liu, "HOPE: a hybrid deep neural model for out-of-town next POI recommendation," *World Wide Web*, vol. 24, no. 5, pp. 1749–1768, 2021.

Chen Wang (S'10-M'13-SM'19) received the B.S. and Ph.D. degrees from the Department of Automation, Wuhan University, China, in 2008 and 2013, respectively. From 2013 to 2017, he was a postdoctoral research fellow in the Networked and Communication Systems Research Lab, Huazhong University of Science and Technology, China. Thereafter, he joined the faculty of Huazhong University of Science and Technology where he is currently an associate professor. His research interests are in the broad areas of Internet of Things, data mining, and mobile computing, with a recent focus on privacy issues in intelligent systems. He is a senior member of IEEE and ACM.

Mengting Yuan received the B.E. degree from Wuhan University of Technology, China, in 2020. She is currently pursuing the M.S. degree in Electronics and Information Engineering at Huazhong University of Science and Technology, China. Her research interests focus on big data analytics in the context of spatio-temporal data and social network data.

Yang Yang (M'17) received the B.E. and M.S. degrees from Wuhan University of Technology, China, in 2009 and 2012, respectively. He received the Ph.D. degree from Huazhong University of Science and Technology, China, in 2017. He is a visiting research fellow at the Department of Computing, The Hong Kong Polytechnic University since 2020. He is currently an associate professor in the School of Artificial Intelligence at Hubei University, China. His research interests include wireless networks, mobile computing and edge computing.

Kai Peng received the B.S., M.S., and Ph.D. degrees from Huazhong University of Science and Technology, China, in 1999, 2002, and 2006, respectively. He is now the faculty of Huazhong University of Science and Technology as a full professor. His current research interests are in the areas of wireless networking and big data processing.

Hongbo Jiang is currently a Distinguished Professor and Vice Dean of the College of Computer Science and Electronic Engineering, Hunan University. He ever was a professor at Huazhong University of Science and Technology. He received his Ph.D. from Case Western Reserve University in 2008. His research concerns computer networking, especially algorithms and protocols for wireless and mobile networks. been serving in the editorial board of IEEE/ACM Transactions on Networking, IEEE Transactions on Mobile Computing, ACM Transactions on Sensor Networks, IEEE Transactions on Network Science and Engineering and IEEE Transactions on Intelligent Transportation Systems. He is an elected Member of Academia Europaea and Fellow of IET. He is Senior Member of ACM and Senior Member of IEEE.