Graph-Enhanced Spatial-Temporal Network for Next POI Recommendation

ZHAOBO WANG, YANMIN ZHU, QIAOMEI ZHANG, HAOBING LIU, and CHUNYANG WANG, Department of Computer Science and Engineering, Shanghai Jiao Tong University
TONG LIU, School of Computer Engineering and Science, Shanghai University

The task of next Point-of-Interest (POI) recommendation aims at recommending a list of POIs for a user to visit at the next timestamp based on his/her previous interactions, which is valuable for both location-based service providers and users. Recent state-of-the-art studies mainly employ recurrent neural network (RNN) based methods to model user check-in behaviors according to user's historical check-in sequences. However, most of the existing RNN-based methods merely capture geographical influences depending on physical distance or successive relation among POIs. They are insufficient to capture the high-order complex geographical influences among POI networks, which are essential for estimating user preferences. To address this limitation, we propose a novel Graph-based Spatial Dependency modeling (GSD) module, which focuses on explicitly modeling complex geographical influences by leveraging graph embedding. GSD captures two types of geographical influences, i.e., distance-based and transition-based influences from designed POI semantic graphs. Additionally, we propose a novel Graph-enhanced Spatial-Temporal network (GSTN), which incorporates user spatial and temporal dependencies for next POI recommendation. Specifically, GSTN consists of a Long Short-Term Memory (LSTM) network for user-specific temporal dependencies modeling and GSD for user spatial dependencies learning. Finally, we evaluate the proposed model using three real-world datasets. Extensive experiments demonstrate the effectiveness of GSD in capturing various geographical influences and the improvement of GSTN over state-of-the-art methods.

CCS Concepts: • **Information systems** → **Data mining**; **Spatial-temporal systems**; **Recommender systems;**

Additional Key Words and Phrases: Next point-of-interest recommendation, LSTM, graph embedding

ACM Reference format:
Zhaobo Wang, Yanmin Zhu, Qiaomei Zhang, Haobing Liu, Chunyang Wang, and Tong Liu. 2022. Graph-Enhanced Spatial-Temporal Network for Next POI Recommendation. *ACM Trans. Knowl. Discov. Data.* 16, 6, Article 104 (July 2022), 21 pages.
https://doi.org/10.1145/3513092

This research is supported in part by the 2030 National Key AI Program of China 2018AAA0100503, National Science Foundation of China (No. 62072304, No. 61772341, No. 61832013), Shanghai Municipal Science and Technology Commission (No. 19510760500, No. 21511104700, No. 19511120300), the Oceanic Interdisciplinary Program of Shanghai Jiao Tong University (No. SL2020MS032), Scientific Research Fund of Second Institute of Oceanography, the open fund of State Key Laboratory of Satellite Ocean Environment Dynamics, Second Institute of Oceanography, MNR, GE China, and Zhejiang Aoxin Co. Ltd.

Authors' addresses: Z. Wang, Y. Zhu (corresponding author), Q. Zhang, H. Liu, and C. Wang, Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China; emails: {w19990112, yzhu, qmayzhang, liuhaobing, wangchy}@sjtu.edu.cn; T. Liu, School of Computer Engineering and Science, Shanghai University, Shanghai, China; email: tong_liu@shu.edu.cn.

1556-4681/2022/07-ART104 $15.00
https://doi.org/10.1145/3513092

1 INTRODUCTION

In recent years, **Location-Based Social Networks** (**LBSNs**) such as Foursquare and Gowalla have become popular in our daily life. As a bridge between online social media and the physical world, LBSNs provide a platform for users to share with friends their locations and attach relevant information. The huge number of accumulated check-in sequences facilitates the understanding of user mobile behaviors and preferences for **Points-of-Interest** (**POIs**), e.g., shopping malls and museums. One significant task in this field is to recommend the next POI based on historical check-in sequences, which plays an important role for both users and LBSNs service providers. Thus, POI recommendation gains great attention from researchers in the last few years [5, 18, 39].

The task of next POI recommendation is proposed to achieve accurate personalized POI recommendation [3, 14]. It focuses on recommending next POI that users tend to visit. As a natural extension of general POI recommendation, next POI recommendation task considers user check-ins as successive sequences, which aims at capturing user dynamic preferences. One of the most distinguished features of next POI recommendation is that geographical influences serve as key factors for recommendation [13, 23, 41]. For example, a POI with a long physical distance is less attractive to users due to the physical restriction. These constraints dramatically increase the difficulty of next POI recommendation.

In the current literature, **Recurrent Neural Network** (**RNN**) framework has been widely applied to model sequential data and thus become a mainstream approach for next POI recommendation [2, 9, 19, 34]. However, classical RNN architectures, like vanilla RNN and **Long Short-Term Memory** (**LSTM**), were primarily proposed to capture relations from successive sequences and they fail to handle geographical influences. Researchers concentrate on integrating various spatial information into RNN architectures to improve model performance. Existing methods for next POI recommendation can be roughly grouped into two paradigms. With the assumption that users prefer to visit neighboring POIs with close physical distance, the first type of methods endeavors to depict geographical adjacency among POIs. **Spatio-Temporal Gated Network** (**STGN**) [39] and **Hierarchical Spatial-Temporal LSTM** (**HST-LSTM**) [15] explicitly leverage physical distances between POIs as model inputs. Ke et al. [26] propose a geo-dilated LSTM to exploit geographical influences among non-successive POIs by constructing a novel distance-based input set. The second type of methods assumes that POIs within the same check-in sequences possessing great influence on each other, implicitly modeling these co-occurring relations. For example, **Spatial Temporal RNN** (**ST-RNN**) [18] focuses on extracting spatial information using spatial and temporal transition matrices. Time-LSTM [43], LSPL [32], and TMCA [16] extend LSTM to characterize user dynamic preferences for next POI recommendation.

Although the aforementioned models acquire significant improvement of next POI recommendation performance, they are subject to quantized mapping methods which directly convert spatial interval and successive relation into discrete features. Most of them (such as STGN and HST-LSTM) only simply compute physical distances between successive check-ins within sequences as input to model geographical influences. However, the underlying structure of real-world data is often highly non-linear and hence hardly be accurately approximated by linear calculation. We argue that these methods may fall short in capturing high-order geographical influences. As shown in Figure 1, general user check-in locations consist of different functional sites such as restaurants, entertainment sites, and so on. Assuming that a user's current location is a restaurant and he/she have finished the lunch, it is quite possible for the user to visit an entertainment site rather than another restaurant due to intrinsic characteristics of POIs, despite the industrial site is closer to the user's current location than the entertainment site. Moreover, the range of user check-in behaviors is limited [22]. Users may present less interest in a distant popular entertainment site due

Fig. 1. An illustration of a user's check-in sequence. The next visited POI of users is influenced by both physical distance and intrinsic characteristics of POI. Dashed arrows point to candidate POIs and the solid arrow points to the actual next visited POI.

to the physical distance. In such situations, it is insufficient to fully capture complex geographical influence among POIs only leveraging physical distance or successive relation.

In this article, we aim to explicitly capture different high-order geographical influences among POIs. To this end, we propose a **Graph-based Spatial Dependency** (**GSD**) modeling module based on graph embedding. POI semantic graphs that reflect the pair-wise relation among POIs are constructed as inputs of GSD, which enable GSD to learn various kinds of geographical latent representations based on both successive and non-successive visited POIs. Besides, the POI semantic graphs we constructed are from a global view (i.e., contained all POIs in datasets), which prevents the GSD module from only capturing shallow influences between successive Check-ins. In specific, we capture distance-based and transition-based geographical latent representations to model corresponding geographical influences, respectively.

Moreover, we fuse geographical representations learned by GSD as spatial dependencies with temporal dependencies learned by LSTM-based method and thus propose a novel **Graph-enhanced Spatial-Temporal Network** (**GSTN**) for next POI recommendation. Specifically, GSTN utilizes an LSTM variant named Time-LSTM to capture user-specific temporal dependencies and then fuses both spatial and temporal dependencies to recommend POIs that users are most likely to visit in the next timestamp. We validate the effectiveness of GSTN on three real-world datasets from Gowalla, Foursquare and Brightkite. Extensive experimental results demonstrate that our proposed GSTN model is able to improve the performance of next POI recommendation.

The main contributions of this article are summarized as follows:

— We propose to explicitly model highly complex geographical influences for fully utilizing spatial dependencies in next POI recommendation. Specifically, we design a novel GSD module based on graph embedding to capture high-order complex geographical influences among POIs. Two POI semantic graphs are designed to learn distance-based and transition-based geographical latent representations for each POI.
— We propose a novel GSTN model for next POI recommendation. GSTN focuses on integrating spatial dependencies of POIs and temporal dependencies of users for comprehensively

estimating user preferences. A novel spatial-temporal attention aggregation module is designed to effectively integrate spatial and temporal dependencies.
— We conduct extensive experiments on three real-world LBSN datasets. The results show that our proposed approach outperforms other baseline methods with regard to different metrics.

The remainder of this article is organized as follows. We first give a brief review of related work in Section 2. Then, we formulate the next POI recommendation problem and notations we defined in Section 3. In Section 4, we introduce the proposed GSTN model with detailed descriptions of each module we designed. Extensive experiments are given in Section 5 to evaluate the effectiveness of GSTN. Conclusions and future work are discussed in Section 6.

2 RELATED WORK

2.1 Next POI Recommendation

The next POI recommendation task aims to recommend the next possible visited POI for users, which is based on their historical check-in sequences of his/her visited POIs with geographic information and gentle time constraints. Traditional methods for next POI recommendation include **Matrix Factorization** (**MF**) based methods, **Markov Chain** (**MC**) based methods and geographical representation learning methods.

Most of the traditional methods are based on MF [25] approach to achieve recommendation. The main purpose of MF is to factorize the user-POI interaction matrix into two-low rank matrices which can be approximately considered as user and item latent representation, respectively. However, Factorization based methods neglect the temporal relation among check-ins, it is hard for them to recommend appropriate next visited POI.

Due to the prominent capture ability of user sequential mobility patterns, MC becomes one of the widely used methods. MC based methods leverage the estimated transition probability generated by past check-ins to recommend the next visited POI. A state-of-the-art method here for next POI recommendation is **Factorizing Personalized Markov Chains** (**FPMC**) [24], which employs POI transition matrix and MF to model transition relation among successive check-ins and user general preferences. FPMC model transition matrix as first-order MC, which considers the relation between current POI and previous one. Based on FPMC, **Factorizing Personalized Markov Chains with Localized Regions** (**FPMC-LR**) [3] further introduces region restriction to only consider POIs within the defined region as candidates. De et al. [17] break the regional restriction and employs FPMC to model long-term and short-term preferences of users for next POI recommendation.

In addition, with the development of representation learning in recent years, geographical representation models that depict geographical influence among POIs are proposed in this field [1, 4, 7, 33, 42]. They have demonstrated that the usage of POI geolocation information to be beneficial for estimating user preferences. A recently proposed GeoIE model [30] is closely related to our GSTN model due to the usage of geographical influence types. GeoIE exploits ingoing and outgoing influences to characterize each specific POI and generate recommendation lists for users based on inner-product results of geographical influences. Our proposed GSTN has several significant differences. First, GSTN focuses on recommending next POI for users, it is a method for sequential POI recommendation. Whereas GeoIE aims to infer user potential interests in unvisited POIs, it is for non-sequential POI recommendation and does not model the sequential dependency at all. Second, GSTN learns geographical latent representations through novel proposed GSD module while GeoIE models geographical influences resorting to inner-product. The algorithm and optimization objectives exist noticeable differences between both methods. Third, only GSTN proposes to integrate spatial and temporal information for user preference estimation. GeoIE merely leverages

geographical influences for POI recommendation. Our experiment results also demonstrate that only utilizing geographical influences is insufficient for next POI recommendation.

2.2 Recurrent Neural Network Based Recommendation

RNN is a successful tool to handle complex dependencies in long-range sequences and has shown state-of-the-art performance in recommendation systems and mobility prediction [31, 35, 36]. Existing methods make a great effort to leverage spatial and temporal information for next POI recommendation. STRNN [18] firstly incorporates RNN units with temporal and spatial matrices, which are parameterized by specific spatial and temporal distances among nearby POIs in check-in sequences. STRNN also proposes to adopt linear interpolation to alleviate the problem of over-abundant parameters. HST-LSTM [15] extends all of the three gates in LSTM cells to utilize spatial-temporal distances as inputs. Besides, HST-LSTM proposes a hierarchical structure for LSTM to capture user dynamic preferences. Time-LSTM [43] equips LSTM with additional time gates to capture interval information between successive items for sequential recommendation. Inspired by Time-LSTM, STGN [39] further integrates additional time gates and physical distance gates into LSTM structure to explicitly model spatial-temporal distance between successive check-in activities for next POI recommendation. To jointly model the long- and short-term preferences, Su et al. [39] develop the state-of-the-art **Long- and Short-Term Preference Modeling** (**LSTPM**) model, which utilizes a context-aware network architecture to explore temporal and spatial relations between previous and current trajectories.

Though these methods effectively involve spatial factors for user preference estimation, they simply adopt shallow quantized mapping methods to capture geographical influences and cannot completely benefit from spatial dependencies. The modeling capability of RNN-based structure for temporal and spatial dependencies is overlooked to some extent [11]. To address this limitation, we explicitly capture highly non-linear geographical influences among successive and non-successive POIs that are extracted via graph embedding and then integrate spatial and temporal dependencies for user preference estimation.

3 PROBLEM FORMULATION

In this section, we give the relevant notations and formalize the problem definition of next POI recommendation as follows.

Definition 1 (Point-of-Interest). A POI is a spatial site (e.g., a restaurant) associated with two attributes: a unique identifier l and geographical coordinates $(longitude, latitude)$ tuple, i.e., (lon, lat).

Definition 2 (Check-in Sequence). In a typical next POI recommendation scenario, Let $\mathcal{U} = \{u_1, u_2, \ldots, u_M\}$ denotes the set of all users and $\mathcal{L} = \{l_1, l_2, \ldots, l_N\}$ denotes the set of all POIs. The historical check-in sequence of user u is defined as $H^u = \{(l_i^u, t_{l_i^u}) | i = 1, 2, \ldots, n\}$, where n is the number of check-in behaviors of user u. Each tuple $(l_i^u, t_{l_i^u})$ represents the i^{th} check-in behavior of user $u \in \mathcal{U}$ with POI $l_i^u \in \mathcal{L}$ and timestamp $t_{l_i^u}$.

Definition 3 (Distance-based Graph). A weighted undirected POI semantic graph $G_D = (V, E_D, A_D)$, where V represents the set of all POIs, i.e, $V = \mathcal{L}$. $E_D \in V \times V$ is the set of edges in the graph G_D. The set of E_D includes an edge $e_{D_{i,j}}$ that describes the distance-based relation between vertices v_i and v_j. $A_D \in \mathbb{R}^{N \times N}$ represents a weighted adjacency matrix where the value of $A_{D_{ij}}$ measures distance-based relation between POI l_i and l_j.

Definition 4 (Transition-based Graph). A weighted directed POI semantic graph $G_T = (V, E_T, A_T)$, where V and E_T are the set of POIs and edges in G_T, respectively. The value of $A_{T_{i,j}}$ in

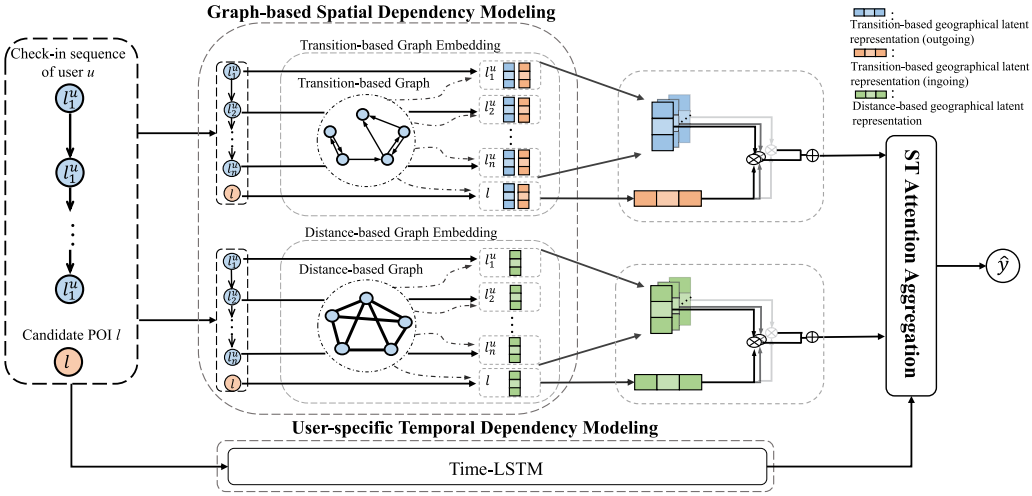


Fig. 2. The overall Architecture of GSTN. Spatial dependency modeling is conducted based on two kinds of POI semantic graphs proposed in the article. Temporal dependency modeling is conducted based on Time-LSTM. ST (i.e., spatial-temporal) attention aggregation adopts attention mechanism to refine geographical influences. Both spatial and temporal dependencies are integrated to estimate user preference.

the matrix A_T denotes transition-based relation between POI l_i and l_j, if $A_{T_{i,j}} \neq 0$ then $e_{T_{i,j}} \in E_T$ and vice versa.

Problem Statement. Given a set of POIs $\mathcal{L}$ and a set of users $\mathcal{U}$, each $u \in \mathcal{U}$ has check-in sequence H^u. The task of next POI recommendation aims to recommend the most interesting POI for user $u \in \mathcal{U}$ at the next timestamp. Specifically, The goal of our model is to estimate user preferences for candidate POIs based on check-in sequences. Furthermore, we generate a top-k recommendation list by ranking the estimated preferences of all candidate POIs in descending order.

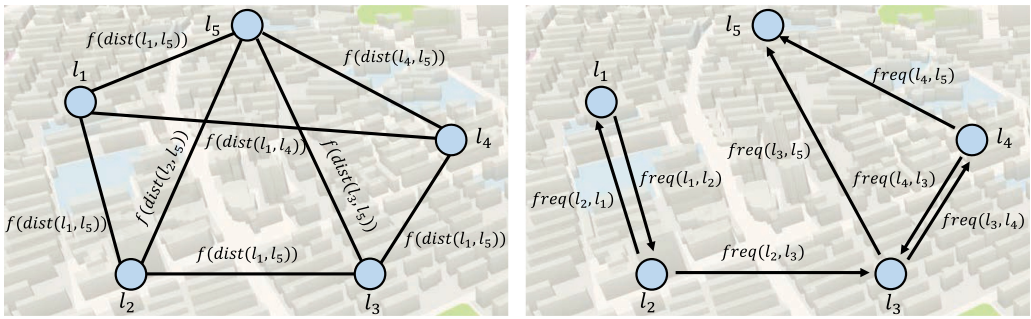
4 GRAPH-ENHANCED SPATIAL-TEMPORAL NETWORK

In this section, we describe the proposed GSTN model in detail. Figure 2 depicts the overall architecture of GSTN. In what follows, we elaborate on the main components of our model, including (1) GSD modeling module, which learns both distance-based and transition-based geographical latent representations to capture spatial dependencies; (2) user-specific temporal dependency modeling module, which explores temporal dependencies from historical check-in sequences; (3) spatial-temporal attention aggregation module, which adaptively aggregate relevant geographical latent representations to update spatial dependencies; (4) prediction module, which integrates spatial and temporal dependencies for estimating user preferences.

4.1 Graph-Based Spatial Dependency Modeling

Empirically, user preferences for individual POIs are limited by geographical constraints. To fully depict complex geographical influences, we capture different types of geographical latent representations based on POI semantic graphs. Specifically, two types of geographical latent representations are learned in this module including distance-based geographical latent representation and transition-based geographical latent representation.

4.1.1 Distance-Based Geographical Latent Representation. The first law of geography [28] points out that "nearby things are more related than distance thing". Some studies [6, 37, 38] also



(a) Distance-based Graph (b) Transition-based Graph

Fig. 3. Illustration of two types of POI semantic graphs. $f(\cdot)$ represents Gaussian kernel function.

show that check-in behaviors of users present aggregation in geographical space, i.e., the next check-in behavior generally occurs in the place close to the current check-in place. Figure 3(a) illustrates how distance information of POIs is used to construct the distance-based graph.

We first calculate the physical distance between POIs from their unique location and construct a symmetrical graph $G_D = (V, E_D, A_D)$, the entry of A_D is defined as

$$A_{D_{i,j}} = \begin{cases} exp\left(-\frac{dist(l_i,l_j)^2}{2\sigma^2}\right), & 0 < dist(l_i,l_j) < \Delta d \\ 0, & otherwise \end{cases}, \tag{1}$$

where $dist(l_i, l_j)$ is the Euclidean distance between POI l_i and l_j, the value of $A_{D_{i,j}}$ is calculated by a Gaussian kernel if $dist(l_i, l_j)$ less than distance threshold Δd. We employ Gaussian kernel function since it not only maps features to high dimensional space but also depicts the inverse correlation between geographical distance and users' visited intent, i.e., users are less interested in a distant POI.

Based on the distance-based graph G_D, we utilize graph embedding to capture distance-based geographical latent representation $\boldsymbol{d} \in \mathbb{R}^k$ of each POI, where k is the dimension of latent representation. Compared with directly utilizing distance between successive check-ins, graph embedding methods adopt non-linear aggregation functions to learn representation of each POI many times, which enables GSD module to capture high-order and non-linear geographical influences. Inspired by [27], for any two connected POIs l_i and l_j in G_D, the conditional probability $p_1(l_i|l_j)$ can be defined as follows:

$$p_1(l_i|l_j) = \frac{exp(\boldsymbol{d_i}^T \boldsymbol{d_j})}{\sum_{l_k \in V} exp(\boldsymbol{d_i}^T \boldsymbol{d_k})}, \tag{2}$$

Furthermore, its empirical probability can be defined as

$$\hat{p}_1(l_i|l_j) = \frac{A_{D_{i,j}}}{\sum_{l_k \in V} A_{D_{i,k}}}, \tag{3}$$

To learn the optimal latent representations of POIs, we adopt KL-divergence to minimize the distance between p_1 and $\hat{p_1}$. Specifically, we omit some constants and minimize the following loss function:

$$L_1 = -\sum_{e_{D_{i,j}} \in E_D} A_{D_{i,j}} \log p_1(l_i|l_j), \tag{4}$$

4.1.2 Transition-Based Geographical Latent Representation. Intrinsic characteristics of POIs are significant factors that affect user preferences. For example, the frequency of (wine bar - entertainment site) is higher than the frequency of (wine bar - industry site). Figure 3(b) illustrates how transition information is used to construct the transition-based graph, we consider the user check-in sequence H as one type of transition relation among different POIs and encode it with directed graph $G_T = (V, E_T, A_T)$, where $e_{T_{i,j}} \in E_T$ describe the direct transition relation from l_i to l_j. The entry of A_T is defined as

$$A_{T_{i,j}} = freq(l_i, l_j), \tag{5}$$

where $freq(l_i, l_j)$ is the transition frequency between POI l_i and l_j among all users. We only count transition records with time interval less than 5 days since successive check-ins with long time interval may not be correlated.

Inspired by [30], we model the intrinsic characteristics of each POI through two transition-based influences, i.e., ingoing and outgoing influences. Ingoing influence represents the capacity of a POI to spread visitors to others and outgoing influence represents the capacity of a POI to receive visitors from others. We utilize the transition-based graph G_T as input to capture those influences. Specifically, for each POI l_i in G_T, The probability $p_2(l_i|l_j)$ of transition from POI l_j to l_i is defined as follows:

$$p_2(l_i|l_j) = \frac{exp(\boldsymbol{g}_j^T \boldsymbol{h}_i)}{\sum_{l_k \in V} exp(\boldsymbol{g}_k^T \boldsymbol{h}_i)}, \tag{6}$$

where $\boldsymbol{g} \in \mathbb{R}^k$ is the latent representation of outgoing influence and $\boldsymbol{h} \in \mathbb{R}^k$ is the latent representation of ingoing influence. Similar to the distance-based latent representation part, the empirical probability of transition between POI l_i and POI l_j and the loss function is given as

$$\hat{p}_2(l_i|l_j) = \frac{A_{T_{i,j}}}{\sum_{l_k \in V} A_{T_{i,k}}}, \tag{7}$$

$$L_2 = -\sum_{e_{T_{i,j}} \in E_T} A_{T_{i,j}} \log p_2(l_i|l_j), \tag{8}$$

To be mentioned, when the number of edges in G_T and G_T is large, updating all the trainable parameters of spatial dependency modeling module through loss functions L1 and L2 (i.e., Equations (4) and (8)) are costly. Hence, following the idea of Wang et al. [30], we adopt the approach of negative sampling and modify L1 and L2 as follows:

$$L_1 = -\sum_{e_{D_{ij}} \in E_D} A_{D_{i,j}} \left\{ log\sigma\left(\boldsymbol{d}_j^T \boldsymbol{d}_i\right) + \sum_{k \in NEG(i)} \left[log\sigma\left(-\boldsymbol{d}_k^T \boldsymbol{d}_i\right)\right] \right\}, \tag{9}$$

$$L_2 = -\sum_{e_{T_{ij}} \in E_T} A_{T_{i,j}} \left\{ log\sigma\left(\boldsymbol{g}_j^T \boldsymbol{h}_i\right) + \sum_{k \in NEG(i)} \left[log\sigma\left(-\boldsymbol{g}_k^T \boldsymbol{h}_i\right)\right] \right\}, \tag{10}$$

where $\sigma(\cdot)$ is the sigmoid function. $NEG(i)$ represents the negative sampling set relative to POI l_i, which includes the whole non-adjacent node of POI l_i. To be mentioned, we select different negative samples for distance-based and transition-based modeling modules, respectively. The number of negative samples is set to 5.

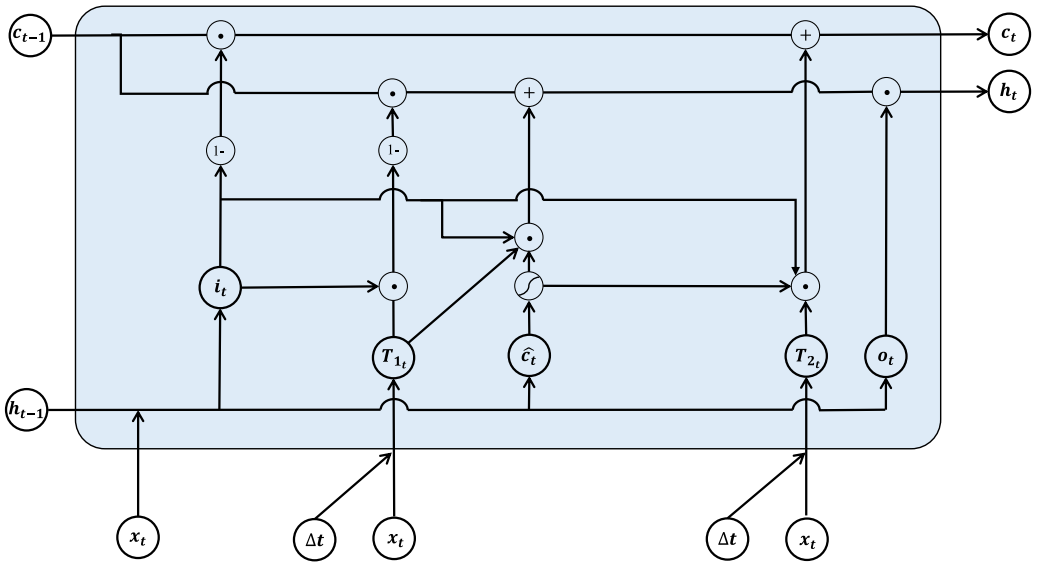


Fig. 4. The structure of Time-LSTM [43].

4.2 User-Specific Temporal Dependency Modeling

When modeling temporal dependencies of users, an intuitive method is to selectively extract the most significant information from check-in sequences of users. Therefore, we focus on modeling temporal dependencies via LSTM-based methods that solve the problem of gradient vanishing and can capture long-term preferences. First, we encode all the POIs in set $\mathcal{L}$ into $\mathcal{X} = \{\boldsymbol{x_i}|i = 1, 2, \ldots, N\}$ using an embedding layer, where $x_i \in \mathbb{R}^k$ is a k dimension embedding vector for the POI $l_i \in \mathcal{L}$. Specifically, $\boldsymbol{x}$ is randomly initialized and is trainable in the networks.

Intuitively, users prefer to visit a nearby POI rather than a remote POI. It is a reasonable assumption that the last POI is a significant factor that affects the next recommendation. In this light, we utilize Time-LSTM [43] to encode the whole information in check-in sequences. Figure 4 illustrates the structure of Time-LSTM. Given an input $\boldsymbol{x}_t^u$ at timestamp t, the output of Time-LSTM hidden layer $\boldsymbol{h}_t^u$ is calculated as following functions:

$$\hat{\boldsymbol{c}}_t = tanh\left(\boldsymbol{W}_{cx}\boldsymbol{x}_t^u + \boldsymbol{W}_{ch}\boldsymbol{h}_{t-1} + \boldsymbol{b}_c\right), \tag{11}$$

$$\boldsymbol{T}_{1_t} = \sigma\left(\boldsymbol{W}_{x_1}\boldsymbol{x}_t^u + \sigma\left(\Delta\boldsymbol{t}_t\boldsymbol{W}_{t_1}\right) + \boldsymbol{b}_1\right), s.t. \boldsymbol{W}_{t_1} \leq \boldsymbol{0}, \tag{12}$$

$$\boldsymbol{T}_{2_t} = \sigma\left(\boldsymbol{W}_{x_2}\boldsymbol{x}_t^u + \sigma\left(\Delta\boldsymbol{t}_t\boldsymbol{W}_{t_2}\right) + \boldsymbol{b}_2\right), \tag{13}$$

$$\tilde{\boldsymbol{c}_t} = (1 - \boldsymbol{i}_t \odot \boldsymbol{T}_{1_t}) \odot \boldsymbol{c}_{t-1} + \boldsymbol{i}_t \odot \boldsymbol{T}_{1_t} \odot \sigma(\hat{\boldsymbol{c}_t}), \tag{14}$$

$$\boldsymbol{c}_t = (1 - \boldsymbol{i}_t) \odot \boldsymbol{c}_{t-1} + \boldsymbol{i}_t \odot \boldsymbol{T}_{2_t} \odot \sigma(\hat{\boldsymbol{c}_t}), \tag{15}$$

$$\boldsymbol{h}_t^u = \boldsymbol{o}_t \odot tanh(\tilde{\boldsymbol{c}_t}), \tag{16}$$

where $\boldsymbol{i}_t$, $\boldsymbol{o}_t$, $\boldsymbol{c}_t$ represent input gate, output gate, and cell state of LSTM, respectively. Δt is the time interval between two adjacent check-ins. Two novel time gates, $\boldsymbol{T}_{1_t}$ and $\boldsymbol{T}_{2_t}$ are designed to emphasize the influence of current interest on the next POI. The constraint $\boldsymbol{W}_{t_1} \leq 0$ characterizes the negative correlation between influence of last POI l_{t-1} on current POI recommendations i.e., if l_{t-1} is visited within a long time, it should have little influence on current POI recommendations.

Note that users' interests are constantly shifting over time in real-life scenarios, check-in behaviors a long time ago might rarely imply current user preferences. Besides, mining special temporal features from the whole check-ins are costly and inefficient. Therefore, we truncate the historical sequences and only consider the most recent twenty visited records as model input.

4.3 Spatial-Temporal Attention Aggregation

Considering that the input of GSD module (i.e., distance-based and transition-based graphs) are constructed from all users' check-ins, the two types of geographical latent representations modeled by GSD module mainly represent geographical influences from a global view. However, different users may have their own intrinsic preferences for POIs. Besides, the same user may have different preferences for the same POIs at different timestamps. To accurately aggregate relevant geographical latent representations, we design a spatial-temporal attention aggregation module to learn user personalized preferences for geographical influences of each visited POI.

To provide personal information for different users, We utilize the hidden state $\boldsymbol{h_t^u}$ learned by Time-LSTM to depict the intrinsic preference of user u at timestamp t, and guide the aggregation of geographical latent representations. Specifically, for distance-based representations of user check-in sequence $(\boldsymbol{d_1}, \boldsymbol{d_2}, \ldots, \boldsymbol{d_t})$ and outgoing influence representations of user check-in sequence $(\boldsymbol{g_1}, \boldsymbol{g_2}, \ldots, \boldsymbol{g_t})$, we adopt attention mechanism [29] to calculate user personal geographical latent representations as follows:

$$\boldsymbol{d_u} = \sum_{i=1}^{t} \alpha_i (\boldsymbol{W_{d_1}} \boldsymbol{d_i}), \tag{17}$$

$$\boldsymbol{g_u} = \sum_{i=1}^{t} \beta_i (\boldsymbol{W_{g_1}} \boldsymbol{g_i}), \tag{18}$$

where $\boldsymbol{d_u}, \boldsymbol{g_u}$ represents user personal distance-based and outgoing influence latent representations, respectively. $\boldsymbol{W_{d_1}}, \boldsymbol{W_{g_1}} \in \mathbb{R}^{k \times k}$ are trainable weight matrices. α_i, β_i are weight coefficients that are calculated as follows:

$$\alpha_i = softmax(\frac{(\boldsymbol{W_{d_2}} \boldsymbol{h_t^u})(\boldsymbol{W_{d_3}} \boldsymbol{d_i})^T}{\sqrt{k}}), \tag{19}$$

$$\beta_i = softmax(\frac{(\boldsymbol{W_{g_2}} \boldsymbol{h_t^u})(\boldsymbol{W_{g_3}} \boldsymbol{g_i})^T}{\sqrt{k}}), \tag{20}$$

where $\boldsymbol{W_{d_2}}, \boldsymbol{W_{d_3}}, \boldsymbol{W_{g_2}}, \boldsymbol{W_{g_3}} \in \mathbb{R}^{k \times k}$ are trainable weight matrices, k is the dimension of latent representations. Figure 5 illustrates this module. To be mentioned, $\boldsymbol{h_t^u}$ is utilized to calculate the weight coefficient of each geographical latent representation, which enable GSTN to model user dynamic preferences on geographical influences at different timestamp.

After the attention aggregation operation, we combine the global and the personal part to compute the geographical influences of visited POIs in check-in sequence H^u of a given user u to a target POI as follows:

$$I_d = \boldsymbol{d_u^T} \boldsymbol{d_j} + \frac{1}{|H^u|} \sum_{l_k \in H^u} \boldsymbol{d_k^T} \boldsymbol{d_j}, \tag{21}$$

$$I_t = \boldsymbol{g_u^T} \boldsymbol{h_j} + \frac{1}{|H^u|} \sum_{l_k \in H^u} \boldsymbol{g_k^T} \boldsymbol{h_j}, \tag{22}$$

where I_d represents the distance-based influence of H^u to the destination l_j, I_t represents the transition-based influence of H_t^u to the destination l_j, $|H^u|$ represents the length of check-in

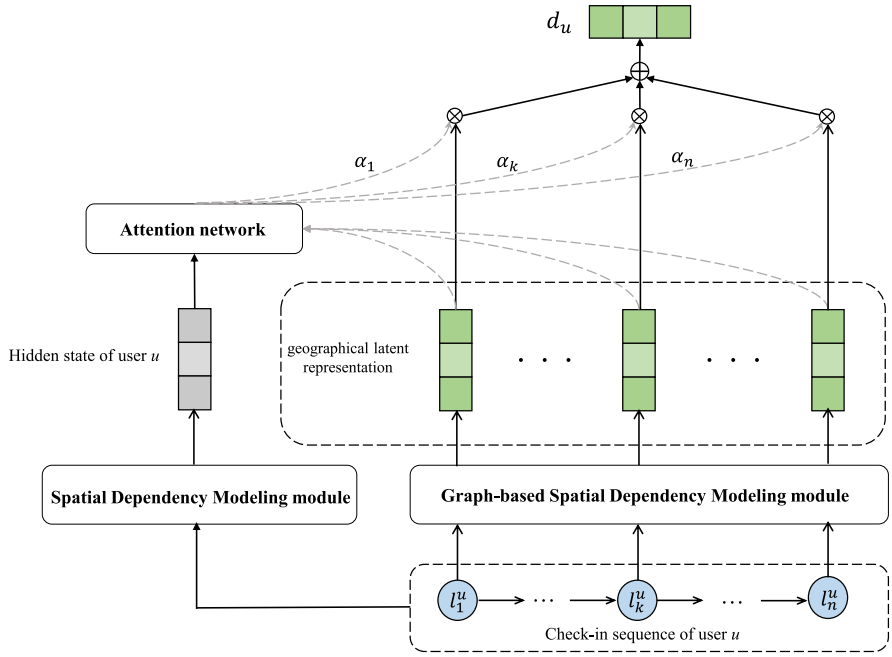


Fig. 5. Illustration of spatial-temporal attention aggregation.

sequence. Since the characters of each visited POI reflect user preferences in varying degrees, all of them may have implicit influences for target POI l_j. Thus, we compute the average of inner product results as global part in Equations (21) and (22).

4.4 Prediction and Optimization

4.4.1 User Preference Estimation. We recommend the next POI for users using the obtained Time-LSTM hidden state $\boldsymbol{h}_t^u$ and geographical latent representations of both distance-based and transition-based influences. Specifically, The probability $\hat{y}_{u_i,j}^{t+1}$ of user u_i visiting l_j in timestamp $t+1$ is calculated as follows:

$$r_{u_i,j}^{t+1} = \boldsymbol{x}_j^T \boldsymbol{h}_t^u + I_d + I_t, \tag{23}$$

$$\hat{y}_{u_i,j}^{t+1} = \frac{exp(r_{u_i,j}^{t+1})}{\sum_{l_k \in L} exp(r_{u_i,k}^{t+1})}, \tag{24}$$

where $r_{u_i,j}^{t+1}$ is the estimated preference of u_i for l_j. Especially, temporal dependencies are represented with the hidden state extracted from user check-in sequences and multiplying it by target POI embedding x_j. Spatial dependencies are represented with I_d and I_t, which influence user preferences from physical distance and geographical characteristic aspects, respectively. Thus, we emphasize that our estimation fuses both temporal and spatial dependencies.

4.4.2 Optimization. To optimize all parameters θ used for preference estimation, we construct a cross entropy-based loss based on ground truth visited POI, which is given as

$$L_3 = -\sum_{u_i \in U} \sum_{t \in H^u} \sum_{k \in \{j\} \cup NEG(j)} \left(y_{u_i,k}^{t+1} log\left(\hat{y}_{u_i,k}^{t+1}\right) + \left(1 - y_{u_i,k}^{t+1}\right) log\left(1 - \hat{y}_{u_i,k}^{t+1}\right)\right) + \lambda ||\theta||_2, \tag{25}$$

ALGORITHM 1: Training Procession of GSTN

Input : check-in sequences $H = \{H^{u_1}, H^{u_2}, \ldots, H^{u_n}\}$, G_d, G_t, learning rate $\{l_1, l_2, l_3\}$, epoch_size, batch_size, λ
Output: learned GSTN model

Initialize parameters $\theta = \{V, G, H, W_*, b_*\}$
while $iter \leq epoch\_size$ **do**
  while $batch < train\_size//batch\_size + 1$ **do**
    Sample from distance graph G_d;
    Optimize loss function L_1 and update V;
  end
  while $batch < train\_size//batch\_size + 1$ **do**
    Sample from transition graph G_t;
    Optimize loss function L_2 and update G, H;
  end
  while $batch < train\_size//batch\_size + 1$ **do**
    Compute $\boldsymbol{h}_{\boldsymbol{u}}^{\boldsymbol{t}}$ based on Equation (16);
    Estimate user preference $\hat{y}_{u_i,j}^{t+1}$ based on Equation (24);
    Optimize loss function L_3 and Update θ;
  end
end

where $y_{u_i,k}^{t+1}$ is the ground truth visited probability of u in timestamp $t+1$, which equals to 1 if $k = j$ and 0 for otherwise. $\lambda||\theta||_2$ represents L_2-regularization used for preventing overfitting.

So far, we have three loss functions in the proposed model. We adopt Adam optimizer to optimize all the trainable parameters by minimizing these loss functions. Although these two graph embedding modules can be optimized separately and then combined the learned latent representations into the prediction phase. This two-phase training generally results in irreversible information loss. For the information lost in the first phase, it cannot be recovered in the second phase. Therefore, we design an end-to-end model to integrate these two parts. Specifically, we utilize the joint learning method to iteratively optimize these three loss functions. Detailed description is provided in Algorithm 1.

5 EXPERIMENT

In this section, we conduct experiments on three real-world LBSN datasets to evaluate the performance of GSTN. We first briefly introduce the three datasets, baseline methods, and evaluation metrics. Then, we compare GSTN with these state-of-the-art baselines and present the experiment results. Finally, we analyze the impact of hyperparameters on model performance.

5.1 Experimental Setup

5.1.1 Datasets. We use three widely used real-world datasets from Gowalla, Foursquare, and Brightkite for evaluation, which are collected from Liu et al. [20] and Zhao et al. [39]. The detailed statistic information is illustrated in Table 1.

(1) Foursquare[1]**:** The Foursquare dataset contains check-ins generated by Foursquare users whose homes are in Singapore from August 2010 to July 2011. The dataset includes 194,108 check-in records by 2,321 users and 5,596 POIs.

[1]https://sites.google.com/site/yangdingqi/home/foursquare-dataset.

Table 1. Statistic of Datasets

Statistics	Foursquare	Gowalla	Brightkite
#user	2,321	10,162	1,850
#POIs	5,596	24,250	1,672
Avg.# check-ins per user	83.6	44.97	140.12
Avg.# check-ins per POI	34.7	18.8	155.04

(2) Gowalla[2]**:** The Gowalla dataset contains check-ins collected from Gowalla users in California and Nevada from February 2009 to October 2010. The dataset includes 456,988 check-in records by 10,162 users and 24,250 POIs.

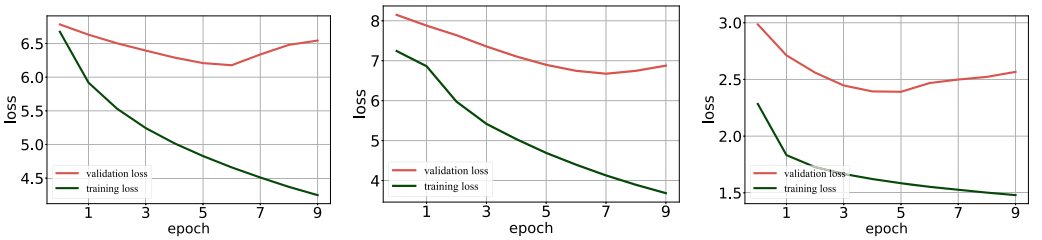
(3) Brightkite[3]**:** The Brightkite dataset contains check-ins collected from global Brightkite users from May 2008 to October 2010. The dataset includes 259,219 check-in records by 1,850 users and 1,672 POIs.

We sort each user's check-in sequences in chronological order. Each dataset is divided into train and test sets. The first 70% check-ins of each user is taken as the training set and the last 20% check-ins is taken as test set, the remaining 10% is taken as validation set.

5.1.2 Baselines. To evaluate the effectiveness of our model, we compare GSTN with the following representative work for next POI recommendation.

— **MF** [25] is one of the classical conventional methods and has been widely used in recommendation tasks.
— **GeoIE** [30] is one of the state-of-the-art model for POI recommendation.
— **FPMC** [24] is the state-of-the-art MC-based model for recommendation tasks. It fuses multiple interaction influences into MC algorithm for better recommendation performance.
— **LSTM** [10] is a variant of RNN model with LSTM cell, which leverages three additional gates to capture long-term dependencies. It shows superiority in handling sequential data.
— **TMCA** [16] is a LSTM-based method which incorporates multiple kinds of contexts for next POI recommendation. For fairness of comparison, we remove the POI categories information since no other methods utilize it.
— **HST-LSTM** [15] combines standard LSTM model with specially design spatial-temporal factor to mitigate data sparsity problem. We remove the hierarchical extension since no session information in our application scenario.
— **Time-LSTM** [43] is a state-of-the-art LSTM-based method for successive recommendation. It models the long and short time context among check-ins through two temporal gates. We adopt the third version since it achieves the best performance on both the original study and our study.
— **STGN** [39] is a state-of-the-art LSTM-based method. It equips LSTM with time and distance gates to considers both spatial and temporal intervals between successive check-ins.
— **ARNN** [8] is a state-of-the-art LSTM-based method. It leverages category information to construct a knowledge graph for next POI recommendation.
— **LSTPM** [26] is a state-of-the-art LSTM-based method, which captures long-term and short-term preferences with a nonlocal network and a geo-dilated RNN, respectively.

[2]http://snap.stanford.edu/data/loc-gowalla.html.
[3]http://snap.stanford.edu/data/loc-brightkite.html.



(a) Foursquare (b) Gowalla (c) Brightkite

Fig. 6. Loss variations of three datasets.

To be mentioned, we do not compare our model with several classical next POI recommendation models such as ST-RNN and PRME, since it has been widely proved that our recurrent baselines adopted in our experiments surpass these methods.

5.1.3 Evaluation Metrics. To evaluate the performance of our proposed GSTN and baselines described above, we employ two widely used evaluation metrics: *Recall@K* and Normalized Discounted Cumulative Gain (*NDCG@K*). *Recall@K* measures the number of correct POIs in the top-k rank list. Noted that it is only one ground truth for each instance in next POI recommendation tasks, *Recall@K* is 1 if the target POI presents in the ranking list of top-k recommendation POIs and 0 otherwise. *NDCG@K* is used to measure the quality of ranking lists. We compute *NDCG@K* for each instance as follows:

$$NDCG@K = \begin{cases} \dfrac{1}{log_2(Rank_i + 1)}, & Rank_i \leq K \\ 0, & Rank_i > K, \end{cases} \tag{26}$$

where $Rank_i$ represents the position of target POI l_i in the ranking list. The overall *Recall@K* and *NDCG@K* are evaluated as the average value of all test instances. These evaluation metrics reflect the comprehensive recommendation ability of each model.

In this article, we report *Recall@K* and *NDCG@K* with the popular $K \in \{2, 5, 10\}$. Each metric is calculated 10 times and averaged. We show the training loss and validation loss variations in Figure 6.

5.1.4 Settings. For FPMC, HST-LSTM, TMCA, Time-LSTM, STGN, and LSTPM we obtain the source codes from the authors or open-source projects. We implement the remaining methods ourselves. For all baselines, We perform proper hyper-parameter tuning by the grid search strategy. We search the learning rate in $\{0.01, 0.005, 0.001, 0.0005, 0.0001\}$, the coefficients λ of l_2 regularization in $\{10^{-3}, 10^{-4}, 10^{-5}\}$, the dropout rate in $\{0, 0.1, 0.2, 0.3, 0.4, 0.5\}$. In our model, the sequence length is fixed to 20 and the distance threshold is set as 1 km, we will give analysis about these two hyper-parameters in the following section. In training process, all weight variables are randomly initialized with uniform distribution. The number of dimensions of latent representation is fixed to 64 and the number of negative samples K is set as 5. The hidden state and cell state are initialized as zero. We optimize our proposed three loss functions in Section 4 with Adam optimizer. The learning rate is set as 0.001. The coefficients λ is set as 10^{-4}. The batch size is set as 128. In evaluation process, we employ all POIs in set $\mathcal{L}$ as candidates for each testing instance and rank them based on the calculated user preferences. We implement our model using Tensorflow library v1.11.

Table 2. Performance Comparison of Different Approaches in Terms of Recall@K and NDCG@K

Foursquare												
Metrics	MF	FPMC	GeoIE	LSTM	Time-LSTM	TMCA	HST-LSTM	STGN	ARNN	LSTPM	GSTN	Improve
Recall@2	0.0925	0.1217	0.1202	0.1296	0.1518	0.1536	0.1426	0.14	0.1402	0.1639	**0.1832***	11.77%
Recall@5	0.1418	0.2014	0.1856	0.1948	0.2222	0.2233	0.2116	0.2062	0.2056	0.2484	**0.2713***	9.21%
Recall@10	0.1816	0.2512	0.2431	0.2522	0.2788	0.2832	0.2703	0.2655	0.2661	0.3231	**0.3416***	5.72%
NDCG@2	0.0811	0.1032	0.1013	0.1141	0.1354	0.1369	0.1286	0.124	0.1164	0.1456	**0.1622***	11.4%
NDCG@5	0.1032	0.1449	0.1355	0.1449	0.167	0.1683	0.1601	0.1537	0.1565	0.1832	**0.2019***	10.2%
NDCG@10	0.1175	0.1568	0.1481	0.1634	0.1853	0.1876	0.1783	0.1729	0.1762	0.2074	**0.2246***	8.29%
Gowalla												
Metrics	MF	FPMC	GeoIE	LSTM	Time-LSTM	TMCA	HST-LSTM	STGN	ARNN	LSTPM	GSTN	Improve
Recall@2	0.0824	0.0836	0.1038	0.1049	0.1112	0.1103	0.1088	0.1097	0.1069	0.1321	**0.139***	5.22%
Recall@5	0.1238	0.1322	0.1349	0.1566	0.1659	0.172	0.1654	0.1703	0.1552	0.1883	**0.2146***	13.96%
Recall@10	0.1611	0.1715	0.1802	0.2026	0.2324	0.2283	0.2148	0.2262	0.2096	0.2498	**0.2791***	11.73%
NDCG@2	0.0727	0.0744	0.0802	0.092	0.0926	0.097	0.0913	0.0956	0.0961	0.1087	**0.1201***	10.48%
NDCG@5	0.0912	0.0953	0.1078	0.1152	0.1233	0.125	0.1192	0.1228	0.1155	0.1376	**0.1541***	13.22%
NDCG@10	0.1033	0.1055	0.1126	0.1299	0.1411	0.1427	0.1356	0.1408	0.1378	0.1642	**0.1749***	6.51%
Brightkite												
Metrics	MF	FPMC	GeoIE	LSTM	Time-LSTM	TMCA	HST-LSTM	STGN	ARNN	LSTPM	GSTN	Improve
Recall@2	0.502	0.5724	0.5703	0.5879	0.6527	0.6353	0.6303	0.6601	0.6372	0.6471	**0.6984***	9.6%
Recall@5	0.5767	0.6868	0.6561	0.6780	0.7164	0.6722	0.6856	0.6997	0.6866	0.7338	**0.7671***	4.53%
Recall@10	0.6205	0.7297	0.7078	0.7386	0.7532	0.7131	0.7318	0.7365	0.7365	0.7638	**0.8008***	4.84%
NDCG@2	0.4824	0.5268	0.5352	0.5358	0.5971	0.5563	0.5443	0.5541	0.5458	0.6217	**0.6619***	6.46%
NDCG@5	0.5116	0.5669	0.5743	0.5642	0.6473	0.633	0.6158	0.6261	0.6146	0.6667	**0.6945***	4.17%
NDCG@10	0.5487	0.6034	0.5911	0.6284	0.6580	0.6487	0.6526	0.6556	0.6471	0.6864	**0.7047***	2.67%

Bold scores are the best results for each metric, while the second best scores are underlined. Improvement indicates the relative improvements of our model over the best baselines. $*$ represents significance level p-value< 0.05 of comparing GSTN with the best baseline.

5.2 Performance Comparison

The recommendation performance of different methods on three datasets in terms of *Recall@K* and *NDCG@K* are illustrated in Table 2. We also conduct a two-tailed T-test with the p-value of 0.05 to show the significance of performance gain of GSTN. From the experiment results, we have following observations:

— Compared with conventional recommendation methods including MF and FPMC, RNN-based methods achieve superior performance in general. It indicates that the capability of RNN in capturing sequential patterns is useful for modeling dynamic user preferences.

— Time-LSTM, TCMA, HST-LSTM, STGN, ARNN and LSTPM achieve better performance compared with LSTM. This demonstrates the effectiveness of leveraging spatial and temporal dependencies. Note that there is no additional POI category information in our application scenario and other baselines. For fairness, we do not construct knowledge graph in ARNN. Time-LSTM performs worse than TMCA, HST-LSTM, STGN, and LSTPM since it only models temporal dependencies and ignores spatial dependencies. In contrast, other methods achieve considerable performance by integrating both spatial and temporal information into the LSTM architecture. However, TMCA, HST-LSTM, and STGN only utilize the distance information between successive POIs as model input, which is not sufficient to depict the complex geographical influences. To conquer this problem, GSTN leverage the newly proposed GSD module to explicitly model the spatial dependency among POIs, which enables GSTN to achieve superior performance compared with these state-of-the-art methods.

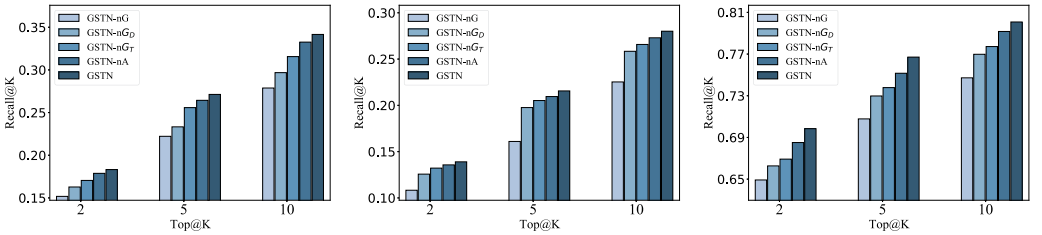
Table 3. Performance Comparison of Different Approaches with SW-EVAL Strategy

	Foursquare				Gowalla				Brightkite			
Metrics	TMCA	STGN	LSTPM	GSTN	TMCA	STGN	LSTPM	GSTN	TMCA	STGN	LSTPM	GSTN
Recall@2	0.1402	0.1324	0.1572	**0.1682***	0.1418	0.1391	0.1524	**0.1634***	0.6685	06844	0.6852	**0.7355***
Recall@5	0.2074	0.1743	0.2328	**0.2496***	0.2003	0.2029	0.2136	**0.2423***	0.7108	0.7442	0.7621	**0.7868***
Recall@10	0.2533	0.2345	0.3031	**0.3168***	0.2677	0.2698	0.2910	**0.3166***	0.7536	0.7758	0.7917	**0.8361***
NDCG@2	0.1152	0.0982	0.1328	**0.1460***	0.1274	0.1303	0.1358	**0.1508***	0.6146	0.6518	0.6528	**0.7091***
NDCG@5	0.1421	0.1324	0.1674	**0.1811***	0.1565	0.1579	0.1711	**0.1866***	0.6587	0.6822	0.6852	**0.7325***
NDCG@10	0.1648	0.1558	0.1913	**0.2028***	0.1839	0.1866	0.1947	**0.2109***	0.6803	0.6953	0.7043	**0.7431***

Bold scores are the best results for each metric, while the second best scores are underlined. Improvement indicates the relative improvements of our model over the best baselines. $*$ represents significance level p-value< 0.05 of comparing GSTN with the best baseline.

— Meanwhile, we observe that the overall performance of all methods on Gowalla is lower than that on other datasets, which is caused by the high data sparsity of Gowalla and thus more difficult to model user preferences from check-in sequences. In this case, additional geographical information can significantly improve performance. We also observe that the performance gains provided by GSTN over these baselines are slightly higher over Gowalla than over Foursquare and Brightkite. This further verifies that GSTN can capture more geographical influences from sparse data compared with other methods.
— For enhancing the ability of the next POI recommendation, LSTPM utilizes a geo-dilated RNN to exploit geographical influences among non-successive POIs as short-term preferences. By comparison with other baselines, LSTPM achieves better performances on the whole evaluation metrics, which strongly illustrates the importance of capturing highly non-linear geographical influences among POIs. However, LSTPM only utilizes RNN based method to implicitly model the geographical influences within successive check-ins and overlook the geographical influences among neighboring POIs. In contrast, GSTN captures geographical latent representations based on distance-based and transition-based graphs, which enable it to capture geographical influences among successive and non-successive POIs thoroughly.
— Our proposed GSTN consistently achieves the best performance over three datasets. In particular, GSTN gains significant improvement over the best baseline LSTPM on all datasets respectively (e.g., 5.72%, 11.73% and 4.84% in term of *Recall*@10). We credit this improvement to the proposed GSD module which enables GSTN to effectively capture distance-based and transition-based geographical influences among successive and non-successive POIs from POI semantic graphs for next POI recommendation. Moreover, GSTN adopts a novel spatial-temporal attention aggregation module to integrate spatial and temporal dependencies, which further improves the performance of recommendation.

Moreover, we also adopt the **Sliding-Window temporal splitting evaluation** (**SW-EVAL**) strategy [12] to thoroughly verify the effectiveness of GSTN. SW-EVAL proposes to split the training set, validation set and test set according to timestamps of check-ins, which may avoid unfair biases when comparing different recommendation methods. For each dataset, we use the check-ins of each user in the first 70% timestamp of the whole dataset as training set and the check-ins of in last 20% timestamp as test set, the remaining 10% as the validation set. We show the results in Table 3. With the SW-EVAL strategy, GSTN still achieves the best performance compared with state-of-the-art methods, which further confirms the superiority of GSTN.



(a) Foursquare (b) Gowalla (c) Brightkite

Fig. 7. Ablation test results on three datasets.

5.3 Ablation Study

To assess the contribution of each component in GSTN to the performance gains, we further conduct ablation tests. We remove each component at a time and obtain four variants as follows.

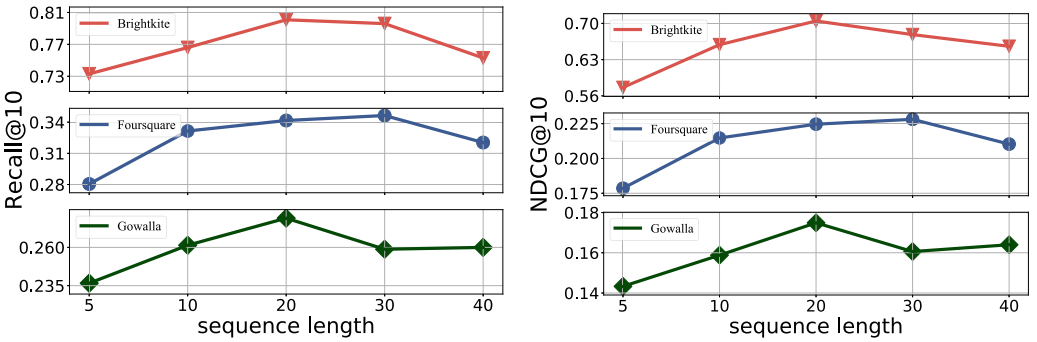
— GSTN-*nG* This variant removes GSD and only considers temporal dependencies.
— GSTN-nG_D This variant removes distance-based influence and only considers transition-based influence and temporal dependencies.
— GSTN-nG_T This variant removes transition-based influence and only considers distance-based influence and temporal dependencies.
— GSTN-*nA* This variant removes the spatial-temporal attention aggregation module and only retains the simple additional aggregation of distance-based and transition-based influence.

Figure 7 shows the results of ablation tests on three datasets. From the overall results, we have the following observations. First, GSTN-*nG* always achieves the worst performance over each dataset. The considerable decrease in performance of GSTN-*nG* compared with GSTN demonstrates that the necessity of spatial dependencies for next POI recommendation. It also further verifies that our proposed GSD module can improve performance by capturing high non-linear geographical influence from different views.

Second, without considering transition-based influence, GSTN-nG_T achieves low performance. This result indicates that intrinsic characteristics of POIs do affect the intention of users for the next visited POI, and GSTN can explicitly model this influence from the transition-based graph effectively. We also notice that leveraging transition-based latent representations obtains considerable performance improvement over Gowalla. One possible reason is that the average check-ins of each POI over Gowalla is low. It is insufficient for LSTM-based methods to model user preferences from sparse successive check-ins, while introducing transition-based latent representations can alleviate this problem effectively.

Third, comparing GSTN and GSTN-nG_D, performance degradation is observed after removing the distance-based graph. It indicates that distance-based influences are important factors to infer user preferences. GSTN adopts the graph structure to capture effective latent representations of distance-based influences, which contributes to the performance gains. Noticed that GSTN-nG_D is less competitive than GSTN-nG_T over all datasets. The possible reason is that the transition-based influence between successive POIs could be learned from sequential patterns captured by LSTM network to some extent.

Finally, we find GSTN-*nA* always achieves better performance compared with GSTN-nG_D and GSTN-nG_T. Considering GSTN-*nA* leverages both the two types of geographical influences proposed in this article through additional aggregation. This result further verifies that the significance of utilizing multiple different geographical influences and the effectiveness of our proposed GSD module for capturing geographical influences. However, although GSTN-*nA* outperforms all



(a) Recall (b) NDCG

Fig. 8. Effect of check-in sequence length Δl.

state-of-the-art baselines, it is inferior to GSTN that includes the attention aggregation module. It suggests the necessity of considering user preferences for geographical influences. Moreover, we notice that the attention module leads to relatively high performance gains over Brightkite dataset. One possible reason is that Brightkite dataset includes sufficient check-ins for each user, which makes it is easier for GSTN to capture user preferences on geographical influences through the attention module.

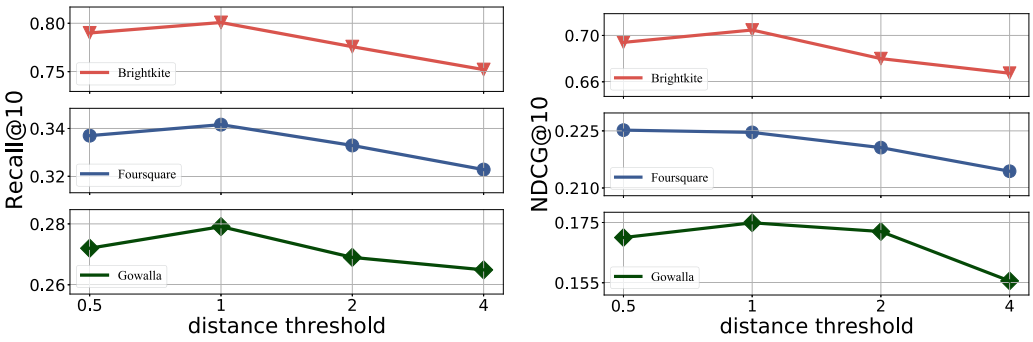
5.4 Influence of Hyper-Parameter

To investigate the impact of different critical hyper-parameter settings, we vary check-in sequence length Δl, distance threshold Δd and time intervals of transition records Δt to evaluate our proposed model on Foursquare and Gowalla, respectively.

We first vary the check-in sequence length Δl from 5 to 40, the result is shown in Figure 8. We can observe that the performance improves with the increase of the check-in length within a certain range. The best performance over Foursquare is achieved when Δl is set to 30 and then decreases. We attribute the performance degradation of further increasing check-in length to excessive noises brought by irrelevant histories. Besides, The best performance over Gowalla is obtained when Δl is set to 20. The difference in optimal sequence length between the two datasets is due to their property. In order to keep consistency, we set check-in length to 20 in our experiments.

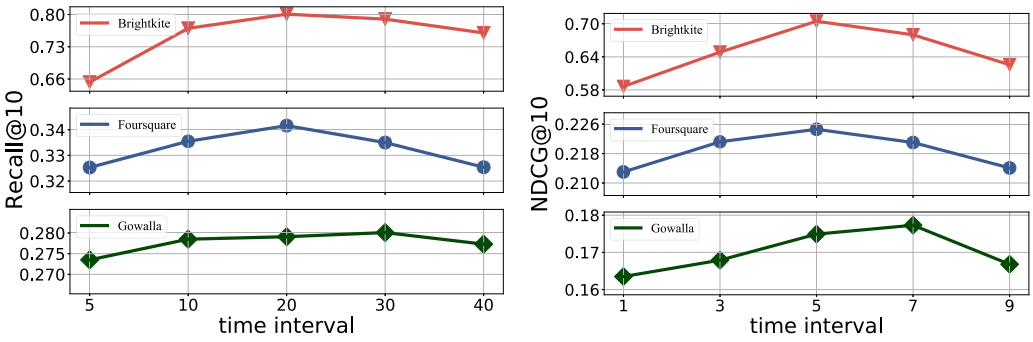
Next, we vary the distance threshold Δd in the distance-based graph from 0.5 km to 4 km. Figure 9 shows the results. Compared with check-in length, our proposed model is less sensitive to Δd. In specific, the best performance is obtained when Δd is set to 1km. Relatively huge performance degradation is observed when Δd is set to 4 km or higher, which is consistent with our intuition, i.e., the influence between two distant POIs is small. This result demonstrates that setting Δd to 1 km is sufficient for capturing distance-based geographical influences.

Finally, we investigate the effectiveness of different time intervals of transition records Δt. This hyper-parameter determines how many transition records are used to construct the transition-based graph. For example, if $\Delta t = 5$, all records with time intervals less than 5 days will be counted. Figure 10 shows the results. Our results demonstrate that the time interval between successive check-ins is important for modeling geographical influences, which is consistent with several previous work [21, 40]. Specifically, GSTN achieves the best performance when Δt is set to 5 over Foursquare, further increased Δt may include irrelevant noises. For Gowalla dataset, the best performance is achieved when $\Delta t = 7$. Considering the data in Gowalla is relatively sparse, we attribute this difference to the inherent properties of datasets.



(a) Recall (b) NDCG

Fig. 9. Effect of distance threshold Δd.



(a) Recall (b) NDCG

Fig. 10. Effect of time interval Δt.

6 CONCLUSION AND FUTURE WORK

In this work, we proposed a GSTN that captures the spatial and temporal dependencies for next POI recommendation. Different from prior methods that only consider spatial features by integrating distance intervals into LSTM architecture, we further propose to model various high-order geographical influences as spatial dependencies. Specifically, we first design a GSD modeling module that leverages POI semantic graphs to capture transition-based and distance-based influences and learns latent representations for POIs. Meanwhile, we employ time-LSTM to obtain user-specific temporal dependencies. Extensive experiments on three datasets demonstrate the effectiveness of GSTN. For future work, we will further explore other context information, i.e., POI category and social network besides the spatial-temporal dependencies to improve the performance of the next POI recommendation.

REFERENCES

[1] Buru Chang, Gwanghoon Jang, Seoyoon Kim, and Jaewoo Kang. 2020. Learning graph-based geographical latent representation for point-of-interest recommendation. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management, CIKM 2020*. 135–144.

[2] Ming Chen, Wen-Zhong Li, Lin Qian, Sang-Lu Lu, and Dao-Xu Chen. 2020. Next POI recommendation based on location interest mining with recurrent neural networks. *Journal of Computer Science and Technology* 35, 3 (2020), 603–616.

[3] Chen Cheng, Haiqin Yang, Michael R. Lyu, and Irwin King. 2013. Where you like to go next: Successive point-of-interest recommendation. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence, IJCAI 2013*. 2605–2611.

[4] Giannis Christoforidis, Pavlos Kefalas, Apostolos Papadopoulos, and Yannis Manolopoulos. 2018. Recommendation of points-of-interest using graph embeddings. In *Proceedings of the 2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA)*. IEEE, 31–40.

[5] Jie Feng, Yong Li, Chao Zhang, Funing Sun, Fanchao Meng, Ang Guo, and Depeng Jin. 2018. DeepMove: Predicting human mobility with attentional recurrent networks. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web, WWW 2018*. 1459–1468.

[6] Shanshan Feng, Xutao Li, Yifeng Zeng, Gao Cong, Yeow Meng Chee, and Quan Yuan. 2015. Personalized ranking metric embedding for next new POI recommendation. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*. 2069–2075.

[7] Shanshan Feng, Lucas Vinh Tran, Gao Cong, Lisi Chen, Jing Li, and Fan Li. 2020. HME: A hyperbolic metric embedding approach for next-POI recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2020*. 1429–1438.

[8] Qing Guo, Zhu Sun, Jie Zhang, and Yin-Leng Theng. 2020. An attentional recurrent neural network for personalized next location recommendation. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence, AAAI 2020*, Vol. 34. 83–90.

[9] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2016. Session-based recommendations with recurrent neural networks. In *Proceedings of the 4th International Conference on Learning Representations, ICLR 2016*.

[10] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation* 9, 8 (1997), 1735–1780.

[11] Haoji Hu, Xiangnan He, Jinyang Gao, and Zhi-Li Zhang. 2020. Modeling personalized item frequency information for next-basket recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2020*. 1071–1080.

[12] Olivier Jeunen, Koen Verstrepen, and Bart Goethals. 2018. Fair O line evaluation methodologies for implicit-feedback recommender systems with MNAR data. In *Proceedings of the REVEAL 18 Workshop on Ofine Evaluation for Recommender Systems (RecSys'18)*.

[13] Xu Jiao, Yingyuan Xiao, Wenguang Zheng, Hongya Wang, and Youzhi Jin. 2019. R2SIGTP: A novel real-time recommendation system with integration of geography and temporal preference for next point-of-interest. In *Proceedings of the World Wide Web Conference, WWW 2019*. 3560–3563.

[14] H. Jing, L. Xin, L. Liao, D. Song, and W. K. Cheung. 2016. Inferring a personalized next point-of-interest recommendation model with latent behavior patterns. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence*.

[15] Dejiang Kong and Fei Wu. 2018. HST-LSTM: A hierarchical spatial-temporal long-short term memory network for location prediction. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*. 2341–2347.

[16] Ranzhen Li, Yanyan Shen, and Yanmin Zhu. 2018. Next point-of-interest recommendation with temporal and multi-level context attention. In *Proceedings of the IEEE International Conference on Data Mining, ICDM 2018*. IEEE, 1110–1115.

[17] Defu Lian, Vincent Wenchen Zheng, and Xing Xie. 2013. Collaborative filtering meets next check-in location prediction. In *Proceedings of 22nd International World Wide Web Conference, WWW'13*. 231–232.

[18] Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. 2016. Predicting the next location: A recurrent model with spatial and temporal contexts. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence*. 194–200.

[19] Tongcun Liu, Jianxin Liao, Zhigen Wu, Yulong Wang, and Jingyu Wang. 2019. A geographical-temporal awareness hierarchical attention network for next point-of-interest recommendation. In *Proceedings of the 2019 on International Conference on Multimedia Retrieval, ICMR 2019*. 7–15.

[20] Wei Liu, Zhi-Jie Wang, Bin Yao, and Jian Yin. 2019. Geo-ALM: POI recommendation by fusing geographical information and adversarial learning mechanism. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*. 1807–1813.

[21] Xin Liu, Yong Liu, and Xiaoli Li. 2016. Exploring the context of locations for personalized location recommendations.. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence, IJCAI 2016*. 1188–1194.

[22] Yiding Liu, Tuan-Anh Pham, Gao Cong, and Quan Yuan. 2017. An experimental evaluation of point-of-interest recommendation in location-based social networks. *Proceedings of the VLDB Endowment* 10, 10 (2017), 1010–1021.

[23] Tieyun Qian, Bei Liu, Quoc Viet Hung Nguyen, and Hongzhi Yin. 2019. Spatiotemporal representation learning for translation-based POI recommendation. *ACM Transactions on Information Systems (TOIS)* 37, 2 (2019), 1–24.

[24] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2010. Factorizing personalized Markov chains for next-basket recommendation. In *Proceedings of the 19th International Conference on World Wide Web, WWW 2010*. 811–820.

[25] Ruslan Salakhutdinov and Andriy Mnih. 2008. Bayesian probabilistic matrix factorization using Markov chain Monte Carlo. In *Proceedings of the 25th International Conference on Machine Learning, ICML 2008*, Vol. 307. 880–887.
[26] Ke Sun, Tieyun Qian, Tong Chen, Yile Liang, Quoc Viet Hung Nguyen, and Hongzhi Yin. 2020. Where to go next: Modeling long- and short-term user preferences for point-of-interest recommendation. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence, AAAI 2020*. 214–221.
[27] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. 2015. LINE: Large-scale information network embedding. In *Proceedings of the 24th International Conference on World Wide Web, WWW 2015*. 1067–1077.
[28] Waldo R. Tobler. 1970. A computer movie simulating urban growth in the Detroit region. *Economic Geography* 46, sup1 (1970), 234–240.
[29] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the Annual Conference on Neural Information Processing Systems, NIPS 2017*. 5998–6008.
[30] Hao Wang, Huawei Shen, Wentao Ouyang, and Xueqi Cheng. 2018. Exploiting POI-specific geographical influence for point-of-interest recommendation. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence, IJCAI 2018*. 3877–3883.
[31] Qinyong Wang, Hongzhi Yin, Tong Chen, Zi Huang, Hao Wang, Yanchang Zhao, and Nguyen Quoc Viet Hung. 2020. Next point-of-interest recommendation on resource-constrained mobile devices. In *Proceedings of the WWW'20: The Web Conference 2020*. 906–916.
[32] Yuxia Wu, Ke Li, Guoshuai Zhao, and Xueming Qian. 2019. Long- and short-term preference learning for next POI recommendation. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019*. 2301–2304.
[33] Min Xie, Hongzhi Yin, Hao Wang, Fanjiang Xu, Weitong Chen, and Sen Wang. 2016. Learning graph-based POI embedding for location-based recommendation. In *Proceedings of the 25th ACM International Conference on Information and Knowledge Management, CIKM 2016*. 15–24.
[34] Dingqi Yang, Benjamin Fankhauser, Paolo Rosso, and Philippe Cudré-Mauroux. 2020. Location prediction over sparse user mobility traces using RNNs: Flashback in hidden states!. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence, IJCAI 2020*. 2184–2190.
[35] Huaxiu Yao, Fei Wu, Jintao Ke, Xianfeng Tang, Yitian Jia, Siyu Lu, Pinghua Gong, Jieping Ye, and Zhenhui Li. 2018. Deep multi-view spatial-temporal network for taxi demand prediction. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*. 2588–2595.
[36] Feng Yu, Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. 2016. A dynamic recurrent model for next basket recommendation. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2016*. 729–732.
[37] Lu Zhang, Zhu Sun, Jie Zhang, Yu Lei, Chen Li, Ziqing Wu, Horst Kloeden, and Felix Klanner. 2020. An interactive multi-task learning framework for next POI recommendation with uncertain check-ins. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence, IJCAI 2020* 301, 985 (2020), 13954.
[38] Kangzhi Zhao, Yong Zhang, Hongzhi Yin, Jin Wang, Kai Zheng, Xiaofang Zhou, and Chunxiao Xing. 2020. Discovering subsequence patterns for next POI recommendation. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence, IJCAI 2020*. 3216–3222.
[39] Pengpeng Zhao, Haifeng Zhu, Yanchi Liu, Jiajie Xu, Zhixu Li, Fuzhen Zhuang, Victor S. Sheng, and Xiaofang Zhou. 2019. Where to go next: A spatio-temporal gated network for next POI recommendation. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence, AAAI 2019*. 5877–5884.
[40] Shenglin Zhao, Tong Zhao, Irwin King, and Michael R. Lyu. 2017. Geo-teaser: Geo-temporal sequential embedding rank for point-of-interest recommendation. In *Proceedings of the 26th International Conference on World Wide Web Companion*. 153–162.
[41] Shenglin Zhao, Tong Zhao, Haiqin Yang, Michael R. Lyu, and Irwin King. 2016. STELLAR: Spatial-temporal latent ranking for successive point-of-interest recommendation. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence*.
[42] Fan Zhou, Ruiyang Yin, Kunpeng Zhang, Goce Trajcevski, Ting Zhong, and Jin Wu. 2019. Adversarial point-of-interest recommendation. In *Proceedings of the World Wide Web Conference, WWW 2019*. 3462–34618.
[43] Yu Zhu, Hao Li, Yikang Liao, Beidou Wang, Ziyu Guan, Haifeng Liu, and Deng Cai. 2017. What to do next: Modeling user behaviors by time-LSTM. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence, IJCAI 2017*. 3602–3608.

Received August 2021; revised November 2021; accepted January 2022