

# Where to Go Next: A Spatio-Temporal Gated Network for Next POI Recommendation

Pengpeng Zhao,<sup>1\*</sup> Haifeng Zhu,<sup>1</sup> Yanchi Liu,<sup>2</sup> Jiajie Xu,<sup>1\*</sup> Zhixu Li,<sup>1</sup>  
Fuzhen Zhuang,<sup>3+</sup> Victor S. Sheng,<sup>4</sup> Xiaofang Zhou<sup>5</sup>

<sup>1</sup>Institute of Artificial Intelligence, School of Computer Science and Technology, Soochow University, China

<sup>2</sup>Rutgers University, USA <sup>3</sup>Key Lab of IIP of Chinese Academy of Sciences (CAS), ICT, CAS, Beijing, China

<sup>4</sup>University of Central Arkansas, USA <sup>5</sup>The University of Queensland, Australia

ppzhao@suda.edu.cn, hfzhu@stu.suda.edu.cn, yanchi.liu@rutgers.edu, {zhixuli, xujj}@suda.edu.cn, zhuangfuzhen@ict.ac.cn, ssheng@uca.edu, zxf@itee.uq.edu.au

## Abstract

Next Point-of-Interest (POI) recommendation is of great value for both location-based service providers and users. However, the state-of-the-art Recurrent Neural Networks (RNNs) rarely consider the spatio-temporal intervals between neighbor check-ins, which are essential for modeling user check-in behaviors in next POI recommendation. To this end, in this paper, we propose a new Spatio-Temporal Gated Network (STGN) by enhancing long-short term memory network, where spatio-temporal gates are introduced to capture the spatio-temporal relationships between successive check-ins. Specifically, two pairs of time gate and distance gate are designed to control the short-term interest and the long-term interest updates, respectively. Moreover, we introduce coupled input and forget gates to reduce the number of parameters and further improve efficiency. Finally, we evaluate the proposed model using four real-world datasets from various location-based social networks. The experimental results show that our model significantly outperforms the state-of-the-art approaches for next POI recommendation.

## Introduction

Recent years have witnessed the rapid growth of location-based social network services, such as Foursquare, Facebook Places, Yelp and so on. These services have attracted many users to share their locations and experiences with massive amounts of geo-tagged data accumulated, e.g., 55 million users generated more than 10 billion check-ins on Foursquare until December 2017. These online footprints (or check-ins) provide an excellent opportunity to understand users' mobile behaviors. For example, we can analyze and predict where a user will go next based on historical footprints. Moreover, such analysis can benefit POI holders to predict the customer arrival in the next time period.

Copyright © 2019, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

\* Corresponding author

+ Fuzhen Zhuang is also with University of CAS, Beijing, China

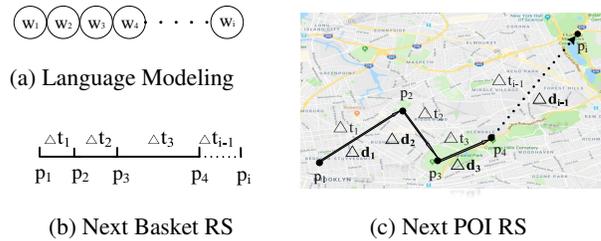


Figure 1:  $w_i$  in (a) represents the  $i$ -th word. In (b),  $p_i$  represents the  $i$ -th item and  $\Delta t$  is time interval between two neighbor items. In (c),  $\Delta d$  further represents distance interval between two successive check-ins.

In the literature, approaches like latent factor model and Markov chain have been widely applied for sequential data analysis and recommendation. Rendle et al.(2010) proposed Factorizing Personalized Markov Chain (FPMC), which bridges matrix factorization and Markov chains together, for next-basket recommendation. Cheng et al.(2013) extended FPMC to embed personalized Markov chain and user movement constraint for next POI recommendation. He et al.(2016) proposed a unified tensor-based latent model to capture the successive check-in behavior by exploring the latent pattern-level preference for each user. Recently, Recurrent Neural Networks (RNNs) have been successfully employed to model sequential data and become state-of-the-art methods. Hidasi et al.(2015) focused on RNN solutions for session-based recommendation task, where no user id exists, and recommendations are made on short session data. Zhu et al.(2017) proposed a variant of Long-Short Term Memory network (LSTM), called Time-LSTM, to equip LSTM with time gates to model time intervals for next item recommendation.

However, none of the above recommendation methods considers both time intervals and geographical distances between neighbor items, which makes next POI recommenda-

tion different from other sequential tasks such as language modeling and next-basket recommender system (RS). As shown in Figure 1, there is no spatio-temporal interval between neighbor words in language modeling, and there is no distance interval between neighbor items in next-basket RS, while there are time and distance intervals between neighbor check-ins in next POI recommendation. Some recent efforts have been made to extend RNNs for modeling dynamic time and distance intervals. For example, HST-LSTM (Kong and Wu 2018) combines spatio-temporal influence into LSTM while it is for general location recommendation. A recent work ST-RNN (Liu et al. 2016a) tried to extend RNN to model the temporal and spatial context for next location prediction. In order to model temporal context, ST-RNN models multi-check-ins in a time window in each RNN cell. Meanwhile, ST-RNN employs time-specific and distance-specific transition matrices to characterize dynamic time intervals and geographical distances, respectively. However, there exists some challenges preventing ST-RNN from becoming the best solution for next POI recommendation.

First of all, ST-RNN may fail to model spatial and temporal relations of neighbor check-ins properly. ST-RNN adopts time-specific and distance-specific transition matrices between cell hidden states within RNN. Due to data sparsity, ST-RNN cannot learn every possible continuous time intervals and geographical distances but partition them into discrete bins. Secondly, ST-RNN is designed for short-term interests and not well designed for long-term interests. Janach et al.(2015) reported that users' short-term and long-term interests are both significant on achieving the best performance. The short-term interest here means that recommended POIs should depend on recently visited POIs, and the long-term interest means that recommended POIs should depend on all historical visited POIs. Thirdly, it is hard to select the proper width of the time window for different applications in ST-RNN since it models multi-elements in a fixed time period.

To this end, in this paper, we propose a new Spatio-Temporal Gated Network by enhancing long short term memory, named STGN, to model users' sequential visiting behaviors. Time intervals and distance intervals of neighbor check-ins are modeled by time gate and distance gate, respectively. Note that there are two time gates and two distance gates in the STGN model. One pair of time gate and distance gate are designed to exploit time and distance intervals to capture the short-term interest, and the other pair are introduced to memorize time and distance intervals to model the long-term interest. Furthermore, enlightened by Greff et al.(2017), we use the coupled input and forget gates to reduce the number of parameters, making our model more efficient. Experimental results on four real-world datasets show STGN significantly improves next POI recommendation performance.

To summarize, our contributions are listed as follows.

- We propose an innovative gate mechanism way to model spatio-temporal intervals between check-ins under LSTM architecture to learn user's visiting behavior for the next POI recommendation.

- A STGN model is proposed to incorporate carefully designed time gates and distance gates to capture the spatio-temporal interval information between check-ins. As a result, STGN well models user's short-term and long-term interests simultaneously.
- Experiments on four large-scale real-world datasets are conducted to evaluate the performance of our proposed model. Our experimental results show that our method outperforms state-of-the-art methods.

## Related Work

In this section, we discuss related work from two aspects, which are POI recommendation and leveraging neural networks for recommendation.

### POI Recommendation

Different from traditional recommendations (e.g., movie recommendation, music recommendation), POI recommendation is characterized by geographic information and no explicit rating information (Ye et al. 2011; Lian et al. 2014). Moreover, additional information, such as social influence, temporal information, review information, and transition between POIs, has been leveraged for POI recommendation. Ye et al.(2011) integrated the social influence with a user-based collaborative filtering model and modeled the geographical influence by a Bayesian model. Yuan et al.(2013) utilized the temporal preference to enhance the efficiency and effectiveness of the solution. Kurashima et al.(2013) proposed a topic model, in which a POI is sampled based on its topics and the distance to historical visited POIs of a target user. Liu et al.(2016b) exploited users' interests and their evolving sequential preferences with temporal interval assessment to recommend POI in a specified time period.

Next POI recommendation, as a natural extension of general POI recommendation, is recently proposed and has attracted great research interest. Research has shown that the sequential influence between successive check-ins plays a crucial role in next POI recommendation since human movement exhibits sequential patterns. A tensor-based model, named FPMC-LR, was proposed by integrating the first-order Markov chain of POI transitions and distance constraints for next POI recommendation (Cheng et al. 2013). He et al.(2016) further proposed a tensor-based latent model considering the influence of user's latent behavior patterns, which are determined by the contextual temporal and categorical information. Feng et al.(2015) proposed a personalized ranking metric embedding method (PRME) to model personalized check-in sequences for next POI recommendation. Xie et al.(2016) proposed a graph-based embedding learning approach, named GE, which utilized bipartite graphs to model context factors in a unified optimization framework. Chang et al.(2018) utilized a check-in context layer and a text content layer to capture the geographical influence of POIs from the check-in sequence of a user and the characteristics of POIs from the text content.

## Neural Networks for Recommendation

Neural networks are not only naturally used for feature learning to model various features of users or items, but also explored as a core recommendation model to simulate nonlinear, complex interactions between users and items (Wang and Wang 2014; Zhang et al. 2016; He et al. 2017; Cui et al. 2017). Yang et al.(2017a) proposed a deep neural architecture named PACE for POI recommendation, which utilizes the smoothness of semi-supervised learning to alleviate the sparsity of collaborative filtering. Yang et al.(2017b) jointly modeled a social network structure and users' trajectory behaviors with a neural network model. Zhang et al.(2017) tried to learn user's next movement intention and incorporated different contextual factors to improve next POI recommendation. Zhu et al.(2017) proposed a Time-LSTM model and two variants, which equip LSTM with time gates to model time intervals for next item recommendation. Huang et al.(2018) integrated the RNN-based networks with knowledge base enhanced key-value memory network (KV-MN) to capture sequential user preference and attribute-level user preference. Lin et al.(2018) proposed a K-plet recurrent neural network to accommodate multiple sequences jointly to capture global structure and localized relationships at the same time.

A recently proposed ST-RNN, which is closely related to our work, considers spatial and temporal contexts to model user behavior for next location prediction (Liu et al. 2016a). However, our proposed STGN model differs significantly from ST-RNN in two aspects. First, STGN equips the LSTM model with time and distance gates while ST-RNN adds spatio-temporal transition matrices to the RNN model. Second, STGN well models time and distance intervals between neighbor check-ins to extract long-term and short-term interests. However, ST-RNN recommends next POI depending only on POIs in the nearest time window which may be hard to distinguish short-term and long-term interests. A more recent work HST-LSTM (Kong and Wu 2018) combines spatio-temporal influences into LSTM for location prediction. However, HST-LSTM is designed for general location recommendation while our proposed model focuses on next POI recommendation. Moreover, our model is equipped with new time and distance gates while HST-LSTM introduces spatio-temporal factors into exists gates in LSTM.

## Preliminaries

In this section, we first give the formal problem definition of next POI recommendation, and then briefly introduce LSTM.

### Problem Formulation

Let  $\mathbb{U} = \{u_1, u_2, \dots, u_M\}$  be the set of  $M$  users and  $\mathbb{V} = \{v_1, v_2, \dots, v_N\}$  be the set of  $N$  POIs. For user  $u$ , she has a sequence of historical POI visits up to time  $t_{i-1}$  represented as  $H_i^u = \{v_{t_1}^u, v_{t_2}^u, \dots, v_{t_{i-1}}^u\}$ , where  $v_{t_i}^u$  means user  $u$  visit POI  $v$  at time  $t_i$ . The goal of next POI recommendation is to recommend a list of unvisited POIs for a user to visit next at time point  $t_i$ . Specifically, a higher prediction score of a user  $u$  to an unvisited POI  $v_j$  indicates a higher probability

that the user  $u$  would like to visit  $v_j$  at time  $t_i$ . According to prediction scores, we can recommend top- $k$  POIs to user  $u$ .

## LSTM

LSTM (Hochreiter and Schmidhuber 1997), a variant of RNN, is capable of learning short and long-term dependencies. LSTM has become an effective and scalable model for sequential prediction problems, and many improvements have been made to the original LSTM architecture. We use the basic LSTM model in our approach for the concise and general purpose, and it is easy to extend to other variants of LSTM. The basic update equations of LSTM are as follows:

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i), \quad (1)$$

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f), \quad (2)$$

$$\tilde{c}_t = \tanh(W_c[h_{t-1}, x_t] + b_c), \quad (3)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t, \quad (4)$$

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o), \quad (5)$$

$$h_t = o_t \odot \tanh(c_t), \quad (6)$$

where  $i_t, f_t, o_t$  represent the input, forget and output gates of the  $t$ -th object, deciding what information to store, forget and output, respectively.  $c_t$  is the cell activation vector representing cell state, which is the key to LSTM.  $x_t$  and  $h_t$  represent the input feature vector and the hidden output vector, respectively.  $\sigma$  represents a sigmoid layer to map the values between 0 to 1, where 1 represents "complete keep this" while 0 represents "completely get rid of this".  $W_i, W_f, W_o$  and  $W_c$  are the weights of gates.  $b_i, b_f, b_o$  and  $b_c$  are corresponding biases. And  $\odot$  represents for the element-wise (Hadamard) product. The update of cell state  $c_t$  has two parts. The former part is the previous cell state  $c_{t-1}$  that is controlled by forget gate  $f_t$ , and the latter part is the new candidate value scaled by how much to add state value.

## Our Approach

In this section, we first propose a Spatio-Temporal Gated Network (STGN) by enhancing long-short term memory, which utilizes time and distance intervals to model user's short-term interest and long-term interest simultaneously. Then, we improve STGN with coupled input and output gates for efficiency.

### Spatio-Temporal Gated Network

When using LSTM for next POI recommendation,  $x_t$  represents user's last visited POI, which can be exploited to learn user's short-term interest. While  $c_{t-1}$  contains the information of user's historical visited POIs, which reflect user's long-term interest. However, how much the short-term interest determines where to go next heavily depends on the time interval and the geographical distance between the last POI and the next POI. Intuitively, a POI visited long time ago and long distance away has little influence on next POI,

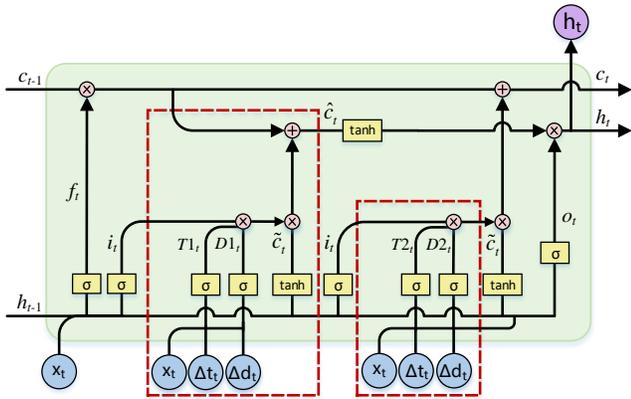


Figure 2: STGN has two time gates and two distance gates, i.e.,  $T1_t, T2_t, D1_t$  and  $D2_t$ .  $T1_t$  and  $D1_t$  are designed to model time and distance intervals for short-term interests while  $T2_t$  and  $D2_t$  are to model time and distance intervals for long-term interest.

and vice versa. In our proposed spatio-temporal gated network model, we use time gate and distance gate to control the influence of the last visited POI on next POI recommendation. Furthermore, the time gate and the distance gate can also help to store time and distance intervals in cell state  $c_t$ , which memorizes user's long-term interest. In this way, we utilize time and distance intervals to model user's short-term interest and long-term interest simultaneously.

As shown in two dotted red rectangles in Figure 2, we add two time gates and two distance gates to LSTM, denoted as  $T1_t, T2_t, D1_t$  and  $D2_t$  respectively.  $T1_t$  and  $D1_t$  are used to control the influence of the latest visited POI on next POI, and  $T2_t$  and  $D2_t$  are used to capture time and distance intervals to model user's long-term interest. Based on LSTM, we add equations for time gates and distance gates as follows:

$$T1_t = \sigma(x_t W_{xt_1} + \sigma(\Delta t_t W_{t_1}) + b_{t_1}), \quad (7)$$

$$s.t. W_{xt_1} \leq 0$$

$$T2_t = \sigma(x_t W_{xt_2} + \sigma(\Delta t_t W_{t_2}) + b_{t_2}), \quad (8)$$

$$D1_t = \sigma(x_t W_{xd_1} + \sigma(\Delta d_t W_{d_1}) + b_{d_1}), \quad (9)$$

$$s.t. W_{xd_1} \leq 0$$

$$D2_t = \sigma(x_t W_{xd_2} + \sigma(\Delta d_t W_{d_2}) + b_{d_2}). \quad (10)$$

We then modify Eq. (4)-(6) to:

$$\hat{c}_t = f_t \odot c_{t-1} + i_t \odot T1_t \odot D1_t \odot \tilde{c}_t, \quad (11)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot T2_t \odot D2_t \odot \tilde{c}_t, \quad (12)$$

$$o_t = \sigma(W_o[h_{t-1}, x_t] + \Delta t_t W_{to} + \Delta d_t W_{do} + b_o), \quad (13)$$

$$h_t = o_t \odot \tanh(\hat{c}_t), \quad (14)$$

where  $\Delta t_t$  is the time interval and  $\Delta d_t$  is the distance interval. Besides input gate  $i_t$ ,  $T1_t$  can be regarded as an input

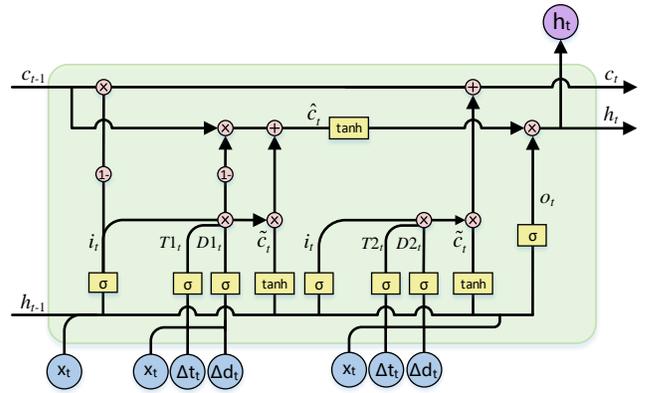


Figure 3: A variant of STGN using coupled input and forget gates.

information filter considering time interval, and  $D1_t$  can be regarded as another input information filter considering distance interval. We add a new cell state  $\hat{c}_t$  to store the result, then transfer to the hidden state  $h_t$  and finally influences next recommendation. Along this line,  $\hat{c}_t$  is filtered by time gate  $T1_t$  and distance gate  $D1_t$  as well as input gate  $i_t$  on current recommendations.

Cell state  $c_t$  is used to memory users general interest, i.e., long-term interest. We designed a time gate and a distance gate to control the cell state  $c_t$  update.  $T2_t$  first memorizes  $\Delta t_t$  then transfers to  $c_t$ , further to  $c_{t+1}, c_{t+2}, \dots$ . So  $T2_t$  helps store  $\Delta t_t$  to model user long-term interest. In the similar way,  $D2_t$  memorizes  $\Delta d_t$  and transfers to cell state  $c_t$  to help model user long-term interest. In this way,  $c_t$  captures user long-term interest by memorizing not only the order of user's historical visited POIs, but also the time and distance interval information between neighbor POIs. Modeling distance intervals can help capture user's general spatial interest, while modeling time intervals helps capture user's periodical visiting behavior.

Normally, a more recently visited POI with a shorter distance should have a larger influence on choosing next POI. To incorporate this knowledge in the designed gates, we add constraints  $W_{xt_1} \leq 0$  and  $W_{xd_1} \leq 0$  in Eq. (7) and Eq. (9). Accordingly, if  $\Delta t_t$  is smaller,  $T1_t$  would be larger according to Eq. (7). In the similar way, if  $\Delta d_t$  is shorter,  $D1_t$  would be larger according to Eq. (9). For example, if time and distance intervals are smaller between  $x_t$  and next POI, then  $x_t$  better indicates the short-term interest, thus its influence should be increased. If  $\Delta t_t$  or  $\Delta d_t$  is larger,  $x_t$  would have a smaller influence on the new cell state  $\hat{c}$ . In this case, the short-term interest is uncertain, so we should depend more on the long-term interests. It is why we set two time gates and two distance gates to distinguish the short-term and long-term interests update.

### Variation of Coupled Input and Forget Gates

Enlightened by (Greff et al. 2017), we propose another version of STGN, named STGCN, to reduce the number of parameters and improve efficiency. STGCN uses coupled input and forget gates instead of separately deciding what to forget and what new information to add, as shown in Figure 3.

Specifically, we remove the forget gate, and modify Eq. (11) and Eq. (12) to:

$$\begin{aligned} \hat{c}_t = & (1 - i_t \odot T1_t \odot D1_t) \odot c_{t-1} \\ & + i_t \odot T1_t \odot D1_t \odot \tilde{c}_t, \end{aligned} \quad (15)$$

$$c_t = (1 - i_t) \odot c_{t-1} + i_t \odot T2_t \odot D2_t \odot \tilde{c}_t. \quad (16)$$

Since time gate  $T1_t$  and distance gate  $D1_t$  are regarded as input filters, we replace the forget gate with  $(1 - i_t \odot T1_t \odot D1_t)$  in Eq. (15).  $T2_t$  and  $D2_t$  are used to store time intervals and distance intervals respectively, thus we use  $(1 - i_t)$  in Eq. (16).

## Training

The way we adapt our model to next POI recommendation is as follows. Firstly we transform  $H^u$  to  $[(v_1^u, t_2^u - t_1^u, d(l_1, l_2)), (v_2^u, t_3^u - t_2^u, d(l_2, l_3)), \dots, (v_n^u, t_q^u - t_n^u, d(l_n, l_q))]$ . Then  $x_t$  in STGN is equivalent to  $v_t^u$ ,  $\Delta t_t$  is equivalent to  $t_{t+1}^u - t_t^u$ , and  $\Delta d_t$  is equivalent to  $d(l_{t+1}, l_t)$ , where  $d(\cdot, \cdot)$  is the function computing the distance between two geographical points. Moreover, we make use of all users' behavioral histories for learning and recommendation. We leverage the mini-batch learning method, and train the model on users' existing histories until convergence. The model output is a probability distribution on all POIs calculated by  $h_t$  and  $v_t^u$ . And then we take a gradient step to optimize the loss based on the output and one-hot representations of  $v_{t+1}^u$ .

We use Adam, a variant of Stochastic Gradient Descent (SGD), to optimize the parameters in STGN, which adapts the learning rate for each parameter by performing smaller updates for frequent parameters and larger updates for infrequent parameters. We use the projection operator described in (Rakhlín, Shamir, and Sridharan 2012) to meet the constraints  $W_{t_1} \leq 0$  in Eq. (7) and  $W_{d_1} \leq 0$  in Eq. (9). If we have  $W_{t_1} > 0$  during the training process, we set  $W_{t_1} = 0$ . And parameter  $W_{d_1}$  is set in the same way.

The computational complexity of learning LSTM models per weight and time step with the stochastic gradient descent (SGD) optimization technique is  $O(1)$ . Hence, the LSTM algorithm is very efficient, with an excellent update complexity of  $O(W)$ , where  $W$  is the number of weights and can be calculated as  $W = n_c \times n_c \times 4 + n_i \times n_c \times 4 + n_c \times n_o + n_c \times 3$ , where  $n_c$  is the number of memory cells,  $n_i$  is the number of input units, and  $n_o$  is the number of output units. Similarly, STGN computational complexity is also  $O(W)$  and can be calculated as  $W = n_c \times n_c \times 5 + n_i \times n_c \times 8 + n_c \times n_o + n_c \times 9$ .

## Experiments

In this section, we conduct experiments to evaluate the performance of our proposed model STGN on four real-world datasets. We first briefly depict the datasets, followed by baseline methods. Finally, we present our experimental results and discussions.

### Dataset

We use four public LBSNs datasets that have user-POI interactions of users and locations of POIs. The statistics of the

Table 1: Statistics of the four datasets

Dataset	#user	#POI	#Check-in	Density
CA	49,005	206,097	425,691	0.004%
SIN	30,887	18,995	860,888	0.014%
Gowalla	18,737	32,510	1,278,274	0.209%
Brightkite	51,406	772,967	4,747,288	0.012%

four datasets are listed in Table 1. CA is a Foursquare dataset from users whose homes are in California, collected from January 2010 to February 2011 and used in Gao et al. (2012). SIN is a Singapore dataset crawled from Foursquare used by (Yuan et al. 2013). Gowalla and Brightkite are two widely used LBSN datasets, which have been used in many related research papers. We eliminate users with fewer than 10 check-ins and POIs visited by fewer than 10 users in the four datasets. Then, we sort each user's check-in records according to timestamp order, taking the first 70% as the training set, the remaining 30% as the testing set.

### Baseline Methods

We compare our proposed model STGN with eight representative methods for next POI recommendation.

- **FPMC-LR** (Cheng et al. 2013): It combines the personalized Markov chains with the user movement constraints around a localized region. It factorizes the transition tensor matrices of all users and predicts next location by computing the transition probability.
- **PRME-G** (Feng et al. 2015): It utilizes the Metric Embedding method to avoid drawbacks of the MF. Specifically, it embeds users and POIs into the same latent space to capture the user transition patterns.
- **GE** (Xie et al. 2016): It embeds four relational graphs (POI-POI, POI-Region, POI-Time, POI-Word) into a shared low dimensional space. The recommendation score is then calculated by a linear combination of inner products for these contextual factors.
- **RNN** (Zhang et al. 2014): This method leverages the temporal dependency in user's behavior sequence through a standard recurrent structure.
- **LSTM** (Hochreiter and Schmidhuber 1997) This is a variant of RNN model, which contains a memory cell and three multiplicative gates to allow long-term dependency learning.
- **GRU** (Cho et al. 2014): This is a variant of RNN model, which is equipped with two gates to control the information flow.
- **ST-RNN** (Liu et al. 2016a): Based on the standard RNN model, ST-RNN replaces the single transition matrix in RNN with time-specific transition matrices and distance-specific transition matrices to model spatial and temporal contexts.

<http://snap.stanford.edu/data/loc-gowalla.html>  
<http://snap.stanford.edu/data/loc-brightkite.html>

- **HST-LSTM** (Kong and Wu 2018): It introduces spatio-temporal factors into gate mechanism in LSTM to mitigate data sparsity problem. Since we do not have session information in our application scenario, we use its ST-LSTM version here.

### Evaluation Metrics

To evaluate the performance of our proposed model ST-LSTM and compare with the seven baselines described above, we use two standard metrics Acc@K and Mean Average Precision (MAP). These two metrics are popularly used for evaluating recommendation results, such as (Liu et al. 2016a; He et al. 2016; Xie et al. 2016). Note that for an instance in testing set, Acc@K is 1 if the visited POI appears in the set of top-K recommendation POIs, and 0 otherwise. The overall Acc@K is calculated as the average value of all testing instances. In this paper, we choose  $K = \{1, 5, 10, 15, 20\}$  to illustrate different results of Acc@K.

### Results and Discussions

**Method Comparison.** The performance of our proposed model STGN and the eight baselines on four datasets evaluated by Acc@K and MAP is shown in Table 2. The cell size and the hidden state size are set to 128 in our experiments. The number of epochs is set to 100, and the batch size is set to 10 for our proposed model. For the parameters of other baselines, we follow the best settings in their papers.

From the experimental results, we can see the following observations: RNN performs better than Markov chain method FPMC-LR and embedding method PRME-G, due to its capability in modeling sequential data and user interests using RNN cell. Both LSTM and GRU slightly improve the performance compared with RNN because of their advantages in modeling long-term interests. The result of GE is not good for missing social and textual information in our datasets. The performance of the state-of-the-art method ST-RNN is close to the standard RNN method, which may be caused by the difficulty of manually setting the windows of time and distance intervals. HST-LSTM performs better than ST-RNN. It proves the effectiveness of the idea of combining spatial-temporal factors with gates mechanism. Our proposed STGN and STGCN model all perform significantly better than existing state-of-the-art methods evaluated here on the four datasets in all metrics. Specifically, STGCN outperforms the Markov chain based methods considerably by a large margin. Moreover, STGCN consistently outperforms five RNN-based methods: RNN, LSTM, GRU, ST-RNN, and HST-LSTM. The performance gains provided by STGCN over these five counterparts are about 34.8% - 68.6%, 16.3% - 80.0%, 32.9% - 97.3% and 2.5% - 34.2% in terms of Acc@1 metric on CA, SIN, Gowalla, and Brightkite respectively. The significant improvement indicates that the mechanism to model temporal and spatial contexts in STGCN can better catch the user’s sequential behaviors and is effective for the task of next POI recommendation. This is because we add time and distance gates to integrate time and distance intervals into the model. Moreover, STGCN not only reduces the number of parameters but also marginally improve the performance compared with STGN.

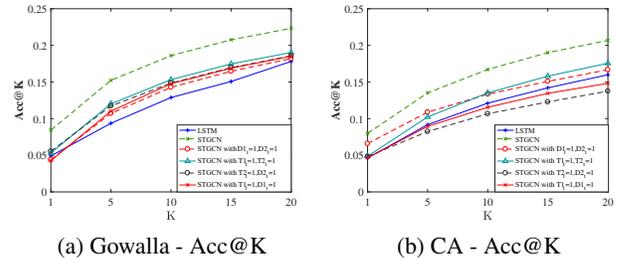


Figure 4: The performance with different time and distance gates in STGCN

**Effectiveness of Time and Distance Gates.** There are two time gates and two distance gates in our STGCN model. We first investigate the effectiveness of time and distance gates on modeling time and distance intervals. Specifically, we set  $D1_t = 1$  and  $D2_t = 1$ , in Eq. (9) and Eq. (10), respectively. That is, we close two distance gates and only consider the time intervals. Similarly, we set  $T1_t = 1$  and  $T2_t = 1$ , in Eq. (7) and Eq. (8), respectively. That is, we close two time gates and only consider distance information. From Figure 4, we can see that the time gates and distance gates have similar importances on both datasets (i.e., Gowalla and CA). Moreover, they both are critical for improving the recommendation performances.

We also investigate the effectiveness of time and distance gates on modeling short-term and long-term interests. We set  $T2_t = 1$  and  $D2_t = 1$ , in Eq. (8) and Eq. (10), which means we close time and distance gates on long-term interests and only activate time and distance gates on short-term interest. Similarly, we set  $T1_t = 1$  and  $D1_t = 1$ , in Eq. (7) and Eq. (9), which means we close time and distance gates for short-term interest. As shown in Figure 4, we can observe that they all perform worse than original STGCN, which means that time and distance intervals are not only critical to short-term interests but also important to long-term interests. Distance intervals may help model user general spatial preference and time intervals may help to model user long-term periodical behavior.

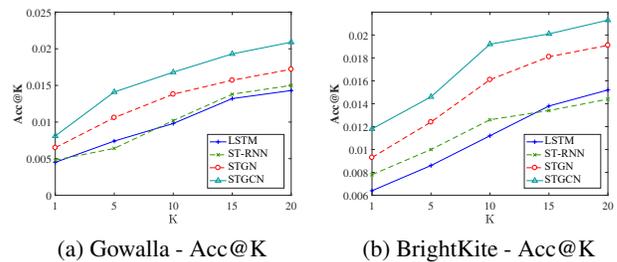


Figure 5: The performance of cold start on two datasets

**Performance of Cold Start.** We also evaluate the performance of STGN by comparing with other next POI recommendation competitors for cold-start users. If a user just vis-

Table 2: Evaluation of next POI recommendation in terms of Acc@K and MAP on four datasets

	CA				SIN			
	Acc@1	Acc@5	Acc@10	MAP	Acc@1	Acc@5	Acc@10	MAP
FPMC-LR	0.0378	0.0493	0.0784	0.1791	0.0395	0.0625	0.0826	0.1724
PRME-G	0.0422	0.065	0.0813	0.1868	0.0466	0.0723	0.0876	0.1715
GE	0.0294	0.0329	0.0714	0.1691	0.0062	0.0321	0.0607	0.1102
RNN	0.0475	0.0901	0.1138	0.1901	0.1321	0.1867	0.2043	0.2186
LSTM	0.0486	0.0937	0.1276	0.1975	0.1261	0.1881	0.2019	0.2123
GRU	0.0483	0.0915	0.1216	0.1934	0.1237	0.1921	0.1992	0.2101
ST-RNN	0.0505	0.0922	0.1232	0.2075	0.1379	0.1957	0.2091	0.2239
HST-LSTM	0.0594	0.1088	0.1372	0.2208	0.1920	0.2504	0.2794	0.3570
STGN	0.0716	0.1232	0.1508	0.2265	0.2157	0.2653	0.2954	0.3570
STGCN	<b>0.0801</b>	<b>0.1308</b>	<b>0.1612</b>	<b>0.2556</b>	<b>0.2232</b>	<b>0.2737</b>	<b>0.3017</b>	<b>0.3608</b>

	Gowalla				Brightkite			
	Acc@1	Acc@5	Acc@10	MAP	Acc@1	Acc@5	Acc@10	MAP
FPMC-LR	0.0293	0.0524	0.0849	0.1745	0.1634	0.2475	0.3164	0.33
PRME-G	0.0334	0.0652	0.0869	0.1916	0.1976	0.2993	0.3495	0.3115
GE	0.0174	0.06	0.0947	0.1973	0.0521	0.1376	0.2118	0.2602
RNN	0.0473	0.0892	0.1207	0.1998	0.3401	0.4087	0.432	0.413
LSTM	0.0503	0.0967	0.1241	0.2004	0.3575	0.4146	0.4489	0.4303
GRU	0.0498	0.0931	0.1289	0.2045	0.331	0.4007	0.4377	0.4042
ST-RNN	0.0519	0.0953	0.1304	0.2187	0.3672	0.4231	0.4477	0.4369
HST-LSTM	0.0702	0.1366	0.1676	0.2414	0.4336	0.4783	0.4999	0.5476
STGN	0.0835	0.1522	0.1879	0.2443	0.4389	0.4807	0.5035	0.5266
STGCN	<b>0.0933</b>	<b>0.1644</b>	<b>0.2020</b>	<b>0.2557</b>	<b>0.4443</b>	<b>0.4953</b>	<b>0.5231</b>	<b>0.5626</b>

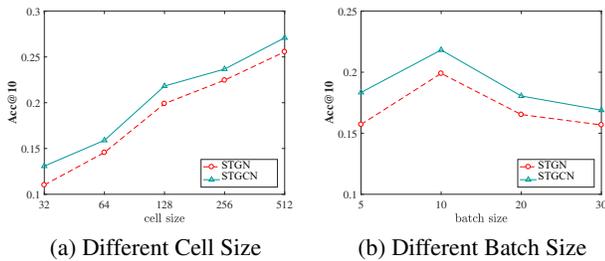


Figure 6: The performance with different cell sizes and batch sizes on Gowalla

its a few POIs in the datasets, which means we can hardly learn user preference on POIs, we think the user is a cold case. Specifically, we take users with less than 5 check-ins as a cold user in our experiments. We conduct the experiments on two datasets (i.e., Gowalla and BrightKite) and use Acc@K as the measure metric. As shown in Figure 5, we can observe that STGCN and STGN perform much better than the other two under cold start scenario, and STGN performs the best among all methods. The reason is that STGN and STGCN model long-term interests as well as short-term interests with considering time and distance intervals, which proves that our method can work out well with sparse data.

**Impact of Parameters.** In the standard RNN, different cell sizes and batch sizes may lead to different performances. We investigate the impact of these two parameters for STGN and STGCN. We vary cell sizes and batch sizes to observe the performance and the training time of our proposed two models. We only show the impact of the two parameters on Gowalla dataset due to space constraint. As shown in Figure 6, increasing the cell size can improve our model in terms of the Acc@10 metric, and a proper batch size can help achieve the best performance. The cell size determines the model complexity, and the cell with a larger size may fit the data better. Moreover, a small batch size may lead to local optimum, and a big one may lead to insufficient updating of parameters in our two models.

## Conclusions

In this paper, a spatio-temporal gated network, named STGN, was proposed for next POI recommendation by enhancing long short term memory network. In STGN, time and distance intervals between neighbor check-ins, which are essential to describe user behaviors, were modeled using newly introduced spatio-temporal gates. Specifically, we added a new cell state, and so there are two cell states to memorize users' short-term and long-term interests, respectively. we designed a pair of time and distance gates to

control user's short-term interest update and another pair of gates to control the long-term interest update. Furthermore, we coupled time and distance gates to reduce number of parameters and improve STGN efficiency. Experimental results on four large-scale real-world datasets demonstrated the effectiveness of our model, which performed better than the state-of-the-art methods. In future work, we would incorporate more context information such as social network and textual description content into the model to further improve the next POI recommendation performance.

## Acknowledgments

This research was partially supported by the NSFC(61876117, 61876217, 61872258, 61728205, 61772242) and the Suzhou Science and Technology Development Program(SYG201803).

## References

- Chang, B.; Park, Y.; Park, D.; Kim, S.; and Kang, J. 2018. Content-aware hierarchical point-of-interest embedding model for successive POI recommendation. In *IJCAI-18*, 3301–3307.
- Cheng, C.; Yang, H.; Lyu, M. R.; and King, I. 2013. Where you like to go next: Successive point-of-interest recommendation. In *IJCAI*, volume 13, 2605–2611.
- Cho, K.; van Merriënboer, B.; Gülçehre, Ç.; Bahdanau, D.; Bougares, F.; Schwenk, H.; and Bengio, Y. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *EMNLP*, 1724–1734.
- Cui, Q.; Wu, S.; Huang, Y.; and Wang, L. 2017. A hierarchical contextual attention-based GRU network for sequential recommendation. *CoRR* abs/1711.05114.
- Feng, S.; Li, X.; Zeng, Y.; Cong, G.; Chee, Y. M.; and Yuan, Q. 2015. Personalized ranking metric embedding for next new poi recommendation. In *IJCAI*, 2069–2075.
- Gao, H.; Tang, J.; and Liu, H. 2012. gscorr: modeling geo-social correlations for new check-ins on location-based social networks. In *CIKM*, 1582–1586. ACM.
- Greff, K.; Srivastava, R. K.; Koutník, J.; Steunebrink, B. R.; and Schmidhuber, J. 2017. LSTM: A search space odyssey. *IEEE Transactions on Neural Networks and Learning Systems* 28(10):2222–2232.
- He, J.; Li, X.; Liao, L.; Song, D.; and Cheung, W. K. 2016. Inferring a personalized next point-of-interest recommendation model with latent behavior patterns. In *AAAI*, 137–143.
- He, X.; Liao, L.; Zhang, H.; Nie, L.; Hu, X.; and Chua, T.-S. 2017. Neural collaborative filtering. In *WWW*, 173–182. International World Wide Web Conferences Steering Committee.
- Hidasi, B.; Karatzoglou, A.; Baltrunas, L.; and Tikk, D. 2015. Session-based recommendations with recurrent neural networks. *CoRR* abs/1511.06939.
- Hochreiter, S., and Schmidhuber, J. 1997. Long short-term memory. *Neural computation* 9(8):1735–1780.
- Huang, J.; Zhao, W. X.; Dou, H.; Wen, J.; and Chang, E. Y. 2018. Improving sequential recommendation with knowledge-enhanced memory networks. In *SIGIR-18*, 505–514.
- Jannach, D.; Lerche, L.; and Jugovac, M. 2015. Adaptation and evaluation of recommendations for short-term shopping goals. In *RecSys'16*, 211–218. ACM.
- Kong, D., and Wu, F. 2018. Hst-ilstm: A hierarchical spatial-temporal long-short term memory network for location prediction. In *IJCAI*, 2341–2347.
- Kurashima, T.; Iwata, T.; Hoshida, T.; Takaya, N.; and Fujimura, K. 2013. Geo topic model: joint modeling of user's activity area and interests for location recommendation. In *WSDM*, 375–384. ACM.
- Lian, D.; Zhao, C.; Xie, X.; Sun, G.; Chen, E.; and Rui, Y. 2014. Geomf: joint geographical modeling and matrix factorization for point-of-interest recommendation. In *SIGKDD*, 831–840. ACM.
- Lin, X.; Niu, S.; Wang, Y.; and Li, Y. 2018. K-plet recurrent neural networks for sequential recommendation. In *SIGIR-18*, 1057–1060.
- Liu, Q.; Wu, S.; Wang, L.; and Tan, T. 2016a. Predicting the next location: A recurrent model with spatial and temporal contexts. In *AAAI*, 194–200.
- Liu, Y.; Liu, C.; Liu, B.; Qu, M.; and Xiong, H. 2016b. Unified point-of-interest recommendation with temporal interval assessment. In *KDD*, 1015–1024.
- Rakhlin, A.; Shamir, O.; and Sridharan, K. 2012. Making gradient descent optimal for strongly convex stochastic optimization. In *ICML'12*, 1571–1578.
- Rendle, S.; Freudenthaler, C.; and Schmidt-Thieme, L. 2010. Factorizing personalized markov chains for next-basket recommendation. In *WWW*, 811–820. ACM.
- Wang, X., and Wang, Y. 2014. Improving content-based and hybrid music recommendation using deep learning. In *MM*, 627–636. ACM.
- Xie, M.; Yin, H.; Wang, H.; Xu, F.; Chen, W.; and Wang, S. 2016. Learning graph-based poi embedding for location-based recommendation. In *CIKM*, 15–24. ACM.
- Yang, C.; Bai, L.; Zhang, C.; Yuan, Q.; and Han, J. 2017a. Bridging collaborative filtering and semi-supervised learning: A neural approach for poi recommendation. In *SIGKDD*, 1245–1254. ACM.
- Yang, C.; Sun, M.; Zhao, W. X.; Liu, Z.; and Chang, E. Y. 2017b. A neural network approach to jointly modeling social networks and mobile trajectories. *ACM TOIS* 35(4):36:1–36:28.
- Ye, M.; Yin, P.; Lee, W.-C.; and Lee, D.-L. 2011. Exploiting geographical influence for collaborative point-of-interest recommendation. In *SIGIR*, 325–334. ACM.
- Yuan, Q.; Cong, G.; Ma, Z.; Sun, A.; and Thalmann, N. M. 2013. Time-aware point-of-interest recommendation. In *SIGIR*, 363–372. ACM.
- Zhang, Y.; Dai, H.; Xu, C.; Feng, J.; Wang, T.; Bian, J.; Wang, B.; and Liu, T.-Y. 2014. Sequential click prediction for sponsored search with recurrent neural networks. In *AAAI*, 1369–1375.
- Zhang, F.; Yuan, N. J.; Lian, D.; Xie, X.; and Ma, W.-Y. 2016. Collaborative knowledge base embedding for recommender systems. In *SIGKDD*, 353–362. ACM.
- Zhang, Z.; Li, C.; Wu, Z.; Sun, A.; Ye, D.; and Luo, X. 2017. Next: A neural network framework for next poi recommendation. *arXiv preprint arXiv:1704.04576*.
- Zhu, Y.; Li, H.; Liao, Y.; Wang, B.; Guan, Z.; Liu, H.; and Cai, D. 2017. What to do next: Modeling user behaviors by time-ilstm. In *IJCAI-17*, 3602–3608.