Hindawi
Complexity
Volume 2021, Article ID 6692313, 13 pages
https://doi.org/10.1155/2021/6692313

*Research Article*

# A Spatial-Temporal Self-Attention Network (STSAN) for Location Prediction

**Shuang Wang**,[1] **AnLiang Li**,[1] **Shuai Xie**,[1] **WenZhu Li**,[1] **BoWei Wang**,[1] **Shuai Yao**,[1] **and Muhammad Asif**[2]

[1]*School of Software, Northeastern University, Shenyang 110000, China*
[2]*Department of Computer Science, Ekha Ghund Degree College Mohmand, Peshawar, KpK 24650, Pakistan*

Correspondence should be addressed to Shuang Wang; wangsh@mail.neu.edu.cn

Received 17 October 2020; Revised 16 March 2021; Accepted 22 March 2021; Published 5 April 2021

Academic Editor: Min Xia



With the popularity of location-based social networks, location prediction has become an important task and has gained significant attention in recent years. However, how to use massive trajectory data and spatial-temporal context information effectively to mine the user's mobility pattern and predict the users' next location is still unresolved. In this paper, we propose a novel network named STSAN (spatial-temporal self-attention network), which can integrate spatial-temporal information with the self-attention for location prediction. In STSAN, we design a trajectory attention module to learn users' dynamic trajectory representation, which includes three modules: location attention, which captures the location sequential transitions with self-attention; spatial attention, which captures user's preference for geographic location; and temporal attention, which captures the user temporal activity preference. Finally, extensive experiments on four real-world check-ins datasets are designed to verify the effectiveness of our proposed method. Experimental results show that spatial-temporal information can effectively improve the performance of the model. Our method STSAN gains about 39.8% Acc@1 and 4.4% APR improvements against the strongest baseline on New York City dataset.

## 1. Introduction

Location-based social networks (LBSNs), such as Foursquare and Gowalla, become increasingly popular, and user-generated digital footprints bring an unprecedented opportunity for exploration of the human mobility patterns. Human mobility and mobility patterns have a high degree of freedom and diversity, so capturing human mobility patterns is a challenging task in LBSN applications, such as personalized recommendation and preference-based route planning.

The traditional methods for location prediction leverage Markov chains (MCs) to capture the transition regularities of human movements [1, 2]. However, the transition probability of the model is predefined, and only the impact of the last check-in activity can be considered. Since deep learning technology has a strong ability of representation learning, recently some location prediction work [3] uses embedding methods and recurrent neural networks (RNNs) to learn location embedding and user representation, respectively. Although the performance has been improved, it cannot solve the problem of gradient vanishing in the training process. To solve this problem, scholars have done a lot of research, using the RNN variants (LSTM [4] and GRU [5]) in the encoder-decoder framework. However, for the sequences over 50 in length, LSTM's perception of further check-in activities is weak, and long-term dependence is still difficult to capture [6]. More recently, the attention mechanism is introduced to dynamically adjust the weight of important records, and it partially resolves the problem that LSTM cannot learn long-distance dependency. However, the one-dimensional attention may neutralize the relationship between vectors, which makes it difficult to discard the

uncorrelated parts in the weighted average vector in order to only remain the highly semantically related content [7].

On the contrary, data records have rich contextual information but are sparse. Spatial and temporal contexts are two key factors [3, 8], which can be effectively used to alleviate the data sparsity. Some studies [6, 9] divide time into 48 hours and encode it into a time feature vector but ignore the influence of spatial context. Kong and Wu [8] encode the time and geographic distance interval as vectors and integrate them into the LSTM module as a time gate and geographic gate to jointly consider the spatial-temporal information. However, this method can only consider the information between adjacent trajectory points in the trajectory sequence, and it cannot tackle the spatial-temporal information between any trajectory points globally and ignores the user's geographical location preference.

To overcome the problems mentioned above, we propose a novel network named STSAN (spatial-temporal self-attention network), which can integrate spatial-temporal information with the self-attention [10] mechanism for location prediction. We design a trajectory attention network to learn users' dynamic trajectory representation, which includes three modules: location attention, spatial attention, and temporal attention. Our model can capture the complicated transitions and the user's preference for geographic location and temporal activity. Finally, different from the previous methods of learning user representation, we represent user trajectory by trajectory points, which are represented in the location embedding space. Our model can avoid compressing user trajectory into a vector, reducing the loss of trajectory information.

The major contributions of this paper can be summarized as follows:

(i) We propose a novel network, STSAN, to capture complex sequential transition regularities and integrate spatial-temporal information for location prediction. Our model can capture the time interval information between any two points in the trajectory and can also sense the user's geographical preference.

(ii) We propose that the trajectory is represented in location latent vector space by the trajectory points instead of the user representation, which avoids the compression and loss of the trajectory information.

(iii) We experimentally evaluate our model using four datasets collected from Foursquare and Gowalla. The experiment results show the superiority of our approach over various baseline approaches, and it is more prominent in sparse datasets. Our model achieves 39.8% Acc@1 and 4.4% APR over VANext [11], which performs the second best on New York City dataset.

## 2. Related Work

*2.1. Location Prediction.* Extensive studies have been dedicated to model human mobility via large-scale trajectory data recorded by GPS, cellular towers, and location-based service. The traditional methods are based on the Markov chains (MCs) model which represents the individual's movement behavior as a Markov model. The MCs calculate a state (location) transition matrix and predict the next location based on the previously visited location [2, 10]. Although time-series information is considered, only a short-range time-series relationship is modeled, which limits its prediction ability. With the widespread development of deep learning research, many neural network models are used to discover the user's mobility patterns. The PRME [12] algorithm embedded users and locations into the hidden space to explore a similar relationship, but it can only model the short-range relationship of the sequence. Recently, recurrent neural networks and their variants have been widely used to capture long-term sequence effects. The STRNN [3] used the RNN model combined with temporal and spatial contextual information to predict the next location. The recurrent neural network model based on spatial-temporal features [13] can automatically extract the internal representation of spatial and temporal features and combine RNN structure for modeling people's movement behavior. It is very difficult to deal with long sequence data in the RNN model, so recently RNN variants (LSTM and GRU) have been proposed to avoid the gradient disappearance of conventional RNN. The HST-LSTM model [8] embedded time and space vectors into LSTM in the framework of encoder and decoder to predict the next location of users. With the further development of deep learning technology, an attention mechanism has been proposed to enhance the learning of RNN and CNN structures on the long-term dependence of longer sequences. The DeepMove [6] used the attention mechanism to enhance RNN to capture the user's mobility and location preference. The VANext [11] used CNN and attention to learn the periodical patterns of historical trajectory to make the next location prediction. The AMF [14] combined a personalized federated learning model with an attention network for location prediction.

*2.2. Attention.* The attention mechanism is widely used in classify images [15], machine translation [16], and various NLP tasks [17–19]. In 2017, the Google teams propose a self-attention model [10] to learn text representation, and self-attention outperforms RNN model with an attention mechanism in sequence to sequence task. Since then, self-attention has been widely used in the recommendation system. In 2017, ATRank [7] used self-attention to capture the impact of users' different behaviors, model user behaviors, and apply it to downstream recommendation tasks. In 2018, Zhang et al. [20] used self-attention to learn the relationship between items and items in user history interaction and made the next item recommendation. In 2018, self-attention is used as a sequence recommendation model, and self-attention is used to capture long-term semantic sequences and attention makes its predictions based on relatively few actions [21]. Xia et al. [22, 23] applied attention in satellite image segmentation to extract useful information and ignore useless information. A global attention module was used to weight low-level features in water segmentation

task. MFANet [24] used a multilevel feature attention module to provide information for the low-level features by using high-level features to generate new features in the segmentation of remote sensing images.

The standard self-attention can only deal with some simple sequence data, such as sentence sequence or timeseries, but it cannot deal with more complex trajectory data for the location prediction problem because trajectory contains timestamps, geographical location coordinate, and other heterogeneous contexts. We propose a trajectory attention mechanism, which integrates geographic and temporal features based on standard self-attention to learn user trajectory representation. Trajectory attention includes three modules: location attention, which learns the relevance between the location vectors in the trajectory; spatial attention, which uses spatial features to learn the user's preference for spatial location; and temporal attention, which captures different time preferences among users.

## 3. Preliminaries

In this section, we first introduce some concepts that are necessary for subsequent discussion and then give an overview of the issues discussed in this paper. Finally, a brief introduction of our model framework is given.

*3.1. Problem Formulation.* Let $U$ and $V$ denote the sets of users and POIs and $|U|$ and $|V|$ represent the number of users and POIs, respectively.

Definition 1 (POI). Point of interest (POI) is a location in a coordinate system, which contains location identification $v$ and the geographical position information (latitude and longitude coordinates).

Definition 2 (trajectory point). A check-in record contains the user identification $u$, POI $v$, and check-in timestamp $t$. The user $u$ visited POI $v$ at $t$ is defined as a tuple $\mathrm{pt}^u = \langle v, t \rangle$, which is also called a user's trajectory point.

Definition 3 (trajectory). Given all trajectory points of the user $u$, the trajectory is a sequence $\mathrm{traj}^u = \mathrm{pt}_1, \ldots, \mathrm{pt}_{L^u}$, where $L^u$ is the length of the user's trajectory sequence and $\mathrm{pt}_i$ is the $i$-th trajectory point of the user $u$.

Location prediction problem. Given the set $U$ of users and the set $V$ of POIs and the current trajectory sequence $\mathrm{traj}^u$ for a user $u \in U$, the problem is to predict the location $v_{L^u+1} \in V$ in the $(L^u + 1)$-th timestamp of user $u$.

*3.2. Basic Framework.* Figure 1 shows the architecture of STSAN, which consists of three major components: (1) trajectory feature processing, (2) trajectory attention module, and (3) prediction.

(i) Trajectory feature processing: we first embed each POI into a low-dimensional vector representation using an embedding method. To build a spatial-temporal user trajectory prediction model, the timestamp and the POI's coordinates are taken as the numerical characteristics of the model.

(ii) Trajectory attention module: we use trajectory attention for learning trajectory representation. Specifically, the location attention calculates the relevance between the location vectors in the trajectory, to capture user's mobility patterns; then, the temporal attention uses the time feature to calculate the time correlation between the trajectory points and to capture the user's mobility patterns. The spatial attention uses spatial features to learn the spatial relationship between POIs and capture the geographic preference of users for POIs. We use the linear function to integrate three output matrices of attention module. Finally, the output of this module is trajectory representation.

(iii) Prediction: the trajectory representation is represented by trajectory points, which are represented in the location embedding space. The inner product is used to calculate the correlation between the trajectory and the location and then infer the user's next visit POI.

## 4. Methodology

*4.1. Trajectory Feature Processing.* To build a spatial-temporal user preference model, we would like to encode the trajectory according to the spatial and temporal characteristics of the user's activities. Word2vec [25] is an effective and scalable method to learn embedding representations by modeling words' contextual correlations in word sentences. We encode each location identification into a low-dimensional vector. For each location, the embedding method outputs the embedded feature vector $\mathbf{p_i}$. The formula representation is as follows:

$$\mathbf{p_i} = a_i W_p, \tag{1}$$

where $a_i$ is the one-hot representation of the $i$-th candidate location and $W_p \in \mathbb{R}^{|V| \times d_p}$ is the location embedding matrix, which is trained and learned by the whole network, where $|V|$ is the number of candidate sites and $d_p$ is the dimension of the embedding feature vector.

Previous work [8] has shown that temporal and spatial features help capture user check-in activities. We extract the timestamp $t$ and latitude and longitude of the place where the user visits and use the location embedding vector as the input of the model. For a user trajectory $\mathrm{traj}^u = \mathrm{point}_1, \mathrm{point}_2, \ldots, \mathrm{point}_L$, we set $L^u = L$, and the input of the model is

$$X^u = x_1, x_2, \ldots, x_L, \tag{2}$$

where $x_i = \{\mathbf{p}_i, (\mathrm{latitude}, \mathrm{longitude})_i, t_i\}$ contains three parts: $\mathbf{p_i}$ is the embedding vector of the $i$-th location in the trajectory, $(\mathrm{latitude}, \mathrm{longitude})_i$ is the latitude and longitude for this location, and $t_i$ is the timestamp of the visited location.

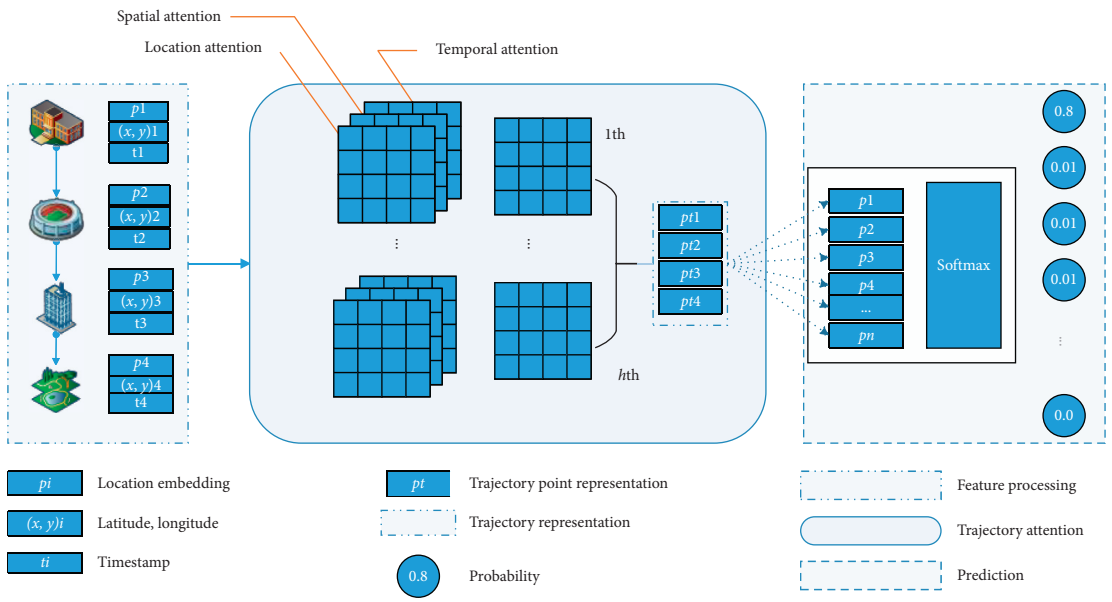



FIGURE 1: The overall architecture of STSAN. First the feature information is extracted from the entire history of a user by feature extraction module. Then, we apply trajectory model to obtain user trajectory representation. At the end of the sequence, the hidden state of trajectory attention is passed through a softmax layer for next location prediction.

*4.2. Trajectory Attention Module.* Figure 2 shows the architecture of trajectory attention module, which consists of three major components: (1) location attention, (2) temporal attention, and (3) spatial attention.

*4.2.1. Location Attention.* The self-attention is a special case of attention function that can be described as mapping a query and a set of key-value pairs to an output, where the query, keys, values, and output are all vectors. Based on the RNN sequence model, it is not easy to control the long-distance hidden state. The attention mechanism allocates weight to the hidden state and gives more weight to the more important state as much as possible. However, it is still limited by RNN recursive structure and cannot process longer sequences. The self-attention can capture the relationship between arbitrary elements by using sequence information regardless of the position in the sequence. The self-attention can process longer sequences, so we use self-attention to capture the relationship between location embedding vectors.

Suppose the length of the user's trajectory is $L$. The input of location attention is a matrix in shape of $L$ by $d_p$:

$$P^u = [\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_L]^T. \tag{3}$$

Note that the trajectory length is different for different users, and the model can accept trajectory sequence input of different lengths.

To get query, key, and value for user $u$, we project $P^u$ to three spaces through nonlinear transformation, as shown in the following formulas:

$$\text{Query} = \text{ReLU}\left(P^u W^Q\right), \tag{4}$$

$$\text{Key} = \text{ReLU}\left(P^u W^K\right), \tag{5}$$

$$\text{Value} = \text{ReLU}\left(P^u W^V\right), \tag{6}$$

where $W_Q \in \mathbb{R}^{d \times d_Q}$, $W_K \in \mathbb{R}^{d \times d_K}$, and $W_V \in \mathbb{R}^{d \times d_V}$ are weight matrices for Query, Key, and Value, respectively. ReLU is a nonlinear activation function, which makes the neural network have enough capacities to capture complex patterns. Note that Query, Key, and Value are projected into the same space; hence, $d_Q = d_K = d_V$ and $d = d_p$ can be known by the combination of matrix multiplication in mathematics.

Then, we can calculate the location's attention score using the following formula:

$$S_p^u = \text{soft max}\left(\frac{\text{Query} \cdot \text{Key}^T}{\sqrt{d_K}}\right). \tag{7}$$

The output is the location's attention matrix in shape of $L$ by $L$. We compute the dot products of the Query and Key, which aims to calculate the similarity or correlation between elements and then divide by $\sqrt{d_K}$ to prevent correlation from being weakened by soft max functions and apply softmax function to get normalized attention weights.

For this module, given a user's check-in location sequence embedding representation $P^u$, the module output is a weight matrix $S_p^u$ of the similarity or correlation of each element. By learning the sequence of check-in locations of all users, the module captures the common mobile pattern among users and the unique interest preferences of different users. This process is learned by training and adjusting the parameters of the model.

*4.2.2. Spatial Attention.* We use spatial features of locations to enhance learning about user mobility patterns and user geographic preferences. Specifically, to reduce the repetition

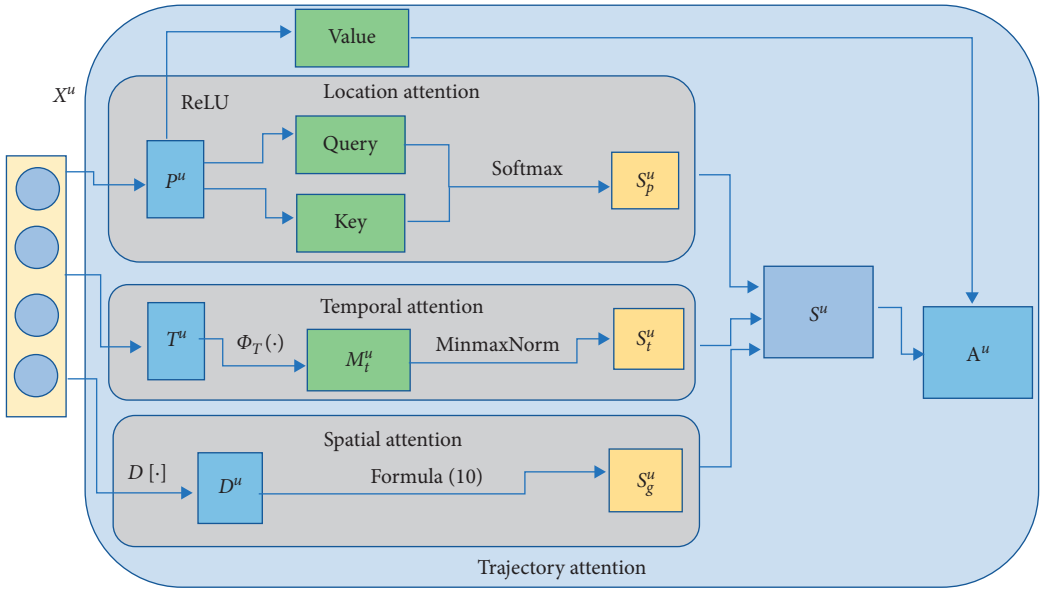



Figure 2: The overall architecture of trajectory attention module.

of calculation, we calculate the geographic distance between all POIs in advance and store it in matrix $D$ and get it by index when using. Given the longitude and latitude of any two POIs of $v_i$ and $v_j$, the spatial distance between the two POIs $d(v_i, v_j)$ is calculated by using the following formula:

$$\Psi\left(\frac{d(v_i, v_j)}{R}\right) = \Psi(\varphi_2 - \varphi_1) + \cos(\varphi_1)\cos(\varphi_2)\Psi(\Delta\lambda), \tag{8}$$

where $\Psi(\theta) = (1 - \cos(\theta))/2$; $R$ is the radius of the Earth, averaging 6371 km; $\varphi_1$ and $\varphi_2$ represent the latitude of the two locations; and $\Delta\lambda$ is the longitude difference between the two locations. We set an index operation $D[\cdot]$ that selects the corresponding values of $D$ to form an $L$ by $L$ matrix:

$$D^u = D[X^u]. \tag{9}$$

Note that the value $D_{ij}^u$ in row $i$ and column $j$ of output $D^u$ represents the actual distance between $v_i$ and $v_j$.

Previous studies [26] have shown that the user's check-in behavior has cluster properties in space. Rather than visiting the places far away, users are more likely to visit the surrounding places that they have gone to, which is congenial with reason and common sense—in our daily life, everyone has their scope of activities, rarely exceeded. To capture this pattern in space, we design spatial attention. To reduce the complexity of the model and facilitate the adjustment of parameters, we manually process the geographical features instead of embedding them into vectors. We set the spatial attention matrix $S_g^u$ in shape of $L$ by $L$ as follows:

$$S_g^u(i, j) = \begin{cases} 1, i \neq j, & D_{ij}^u < \varepsilon, \\ 0, & \text{otherwise.} \end{cases} \tag{10}$$

The purpose of our work is to increase the spatial relevance of locations among users' access areas and indirectly reduce the spatial relevance of locations far away from user preference areas. Note that the distance from a place to itself is zero, so we treat its spatial relevance as a special value of 0. If the value of $D_{ij}^u$ is less than the threshold value $\varepsilon$, the spatial correlation is 1; in other cases, the value is set to 0, and the value on the diagonal is also 0.

*4.2.3. Temporal Attention.* Different user's check-in behavior has its characteristics in time, such as the time interval. Therefore, we take the two check-in time intervals in the user trajectory as features to capture the time preferences between different users. Similar to spatial attention, we take the time interval between the trajectory points as the feature to calculate the attention in time. The temporal attention input is $T^u$, that is, a time sequence as follows:

$$T^u = [t_1, t_2, \ldots, t_L]^T. \tag{11}$$

We set the temporal attention matrix $S_t^u$ in shape of $L$ by $L$ as shown in the following formulas:

$$S_t^u = \text{MinMaxNorm}(M_t^u), \tag{12}$$

$$M_t^u = \Phi_T(T^u), \tag{13}$$

$$M_t^u(i, j) = t_i - t_j, \tag{14}$$

where $M_t^u$ is a matrix in shape of $L$ by $L$ and $M_t^u(i, j)$ is the time interval between the check-in of user $u$ at location $v_i$ and location $v_j$. MinMaxNorm $(\cdot)$ is min-max normalization function that maps the value of the feature to the interval $[0, 1]$. $\Phi_T(\cdot)$ is that we define an operation to map $T^u$ to $M_t^u$.

We use the parameters $\beta$ and $\gamma$ to connect location attention matrix and spatial matrix and temporal matrix linearly and get the total attention matrix formulate as

$$S^u = S_p^u + \beta S_g^u + \gamma S_t^u. \tag{15}$$

The output $S^u$ is a matrix in shape of $L$ by $L$, representing the location vector relevance affected by time and space features. Hyperparameters $\beta$ and $\gamma$ are the time and space influencing factors, indicating the influencing degree of the two factors.

Next, the soft max function is introduced to convert the score $S^u$ of total attention, which is the weight coefficient corresponding to the value, and then the weighted sum is performed:

$$A^u = \text{soft max}\left(S^u\right) \cdot \text{Value}. \tag{16}$$

The output $A^u$ is a matrix in shape of $L$ by $d_V$. On the one hand, the soft max function can be normalized to sort the original calculated score into a probability distribution with the sum of all element weights of 1; on the other hand, it can also highlight the weight of important elements through the internal mechanism of soft max.

We design $h$-times attention in different $h$-spaces, which is similar to the convolution kernel of the convolutional neural network. This allows the model to learn relevant information in different representation subspaces. The results we obtained are as follows:

$$A^u_{K_1}, A^u_{K_2}, \ldots, A^u_{K_h}. \tag{17}$$

Finally, we concatenate the $h$-order attention results and use the value obtained by linear transformation as the result of multiple attentions:

$$\text{Traj}^u = \text{concat}\left(A^u_{K_1}, A^u_{K_2}, \ldots, A^u_{K_h}\right)W^A, \tag{18}$$

where $W_A \in \mathbb{R}^{hd_V \times d}$ is the projection parameter matrix, the concat $(\cdot)$ connection function concatenates the last dimension of tensors, and the output $\text{Traj}^u$ is a matrix in shape of $L$ by $d$, representing the embedded trajectory for user $u$. Note that $h$ is the number of parallel attention layers and $d_V = d/h$.

*4.3. Prediction.* Given the user $u_i$ trajectory sequence $\text{traj}^u$, we use the trajectory attention model to model the embedding vector of trajectory at a higher level to obtain $\text{Traj}^{u_i}$. Specifically, we first map the one-hot vector of locations to the vector space of location through the embedding matrix $W_p$. In the process of trajectory coding, the trajectory vector is mapped to locations vector space in time order, and then, the closest location to the projection vector is found in the location vector space as the prediction result. We formulate it as

$$p\left(v^{u_i}_{L^u+1} | \text{Traj}^{u_i}\right) = \text{soft max}\left(\text{Traj}^{u_i} W_p^T + b\right). \tag{19}$$

Here, the output $p\left(v^{u_i}_{L^u+1} | \text{Traj}^{u_i}\right)$ is a probability distribution, which represents the probability distribution of the next position under the condition of known $\text{Traj}^{u_i}$ and $b$ is the bias parameter of the current layer network.

Different from the previous work, we not only realize the weight sharing of $W_p$ but also use trajectory points to dynamically represent user trajectory. This method not only reduces the risk of overfitting but also avoids compressing the trajectory to be represented as a vector and keeps more trajectory information.

*4.4. Training Algorithm.* In order to model user spatial activity preference in a continuous manner, we propose trajectory attention model by considering the spatial and temporal features. In this paper, we consider the next location prediction problem as a multiclassification problem, so we use cross-entropy loss function combining regularization term as the objective function of model training. The objective function of the proposed model is shown as

$$\mathscr{L} = -\sum_{u_i \in U} \sum_{m=1}^{L^{u_i}-1} p^{u_i}_{m+1} \log\left(p\left(v^{u_i}_{m+1} | \text{Traj}^{u_i}_m\right)\right) + \lambda_1 \|\Theta\|_1 + \lambda_2 \|\Theta\|_2, \tag{20}$$

where $\text{Traj}^{u_i}_m$ is trajectory representation at the $m$-th moment; $\lambda_1, \lambda_2$ are the parameters of $L1$ regularization and $L2$ regularization, respectively; and $\Theta$ represents the parameters to be regularized in the model. The gradient descent and backpropagation algorithms are used to modify the network connection parameters, and then, the objective function is minimized. The source code of our model is available online (https://github.com/li-neu/SASAN).

## 5. Experiments

*5.1. Dataset.* We evaluate the proposed model with state-of-the-art methods on four check-ins datasets from the following two publicly available datasets:

(i) Foursquare dataset [27]. This dataset contains check-ins in NYC and Tokyo collected for about 10 months (from 12 April 2012 to 16 February 2013). It contains 227,428 check-ins in New York City and 573,703 check-ins in Tokyo. Each check-in is associated with its timestamp and its GPS coordinates.

(ii) Gowalla dataset [28]. This dataset is collected for about 18 months (from 1 February 2009 to 31 October 2010). It contains 6,442,890 check-ins with its timestamp and its GPS coordinates. We select the check-ins in Los Angeles and Houston as experiment dataset.

For each city dataset, we remove users with less than 10 check-in records. The user trajectory is divided into several subtrajectories according to the interval between two adjacent records, and the interval is set to 72 hours, as it was done in previous related works [6]. Then, we remove subtrajectories whose length is less than 2 and remove users whose number of subtrajectories is less than 5. Table 1 shows the basic information of data preprocessing. We calculated the average number of records per user per day on each dataset as the sparsity of the dataset, as shown in Table 1. The size of the time dimension of a single input sample is 5.17, 4.76, 23.33, and 18.94 days, respectively. The maximum length of the model input is set to 50 on all the four datasets. The actual input is not fixed and is determined by the length of the input trajectory.

Table 1: Statistics of datasets.

| | Foursquare | | Gowalla | |
|---|---|---|---|---|
| City | New York | Tokyo | Los Angeles | Houston |
| Users | 1,082 | 2,293 | 1,057 | 821 |
| Locations | 34,440 | 56,106 | 10,657 | 10,282 |
| Records | 184,509 | 449,640 | 46,397 | 45,553 |
| Loc./user | 32 | 24 | 10 | 12 |
| Sparsity | 0.5684 | 0.6536 | 0.0813 | 0.1027 |

To show the check-in situation of users in the dataset more intuitively, we draw the cumulative distribution of the check-in quantity over four datasets, which is shown in Figure 3. We can see that the cumulative distribution of the check-in frequency of four datasets is similar to power-law distribution [26, 29]. It can also be seen that the distribution of two cities collected on the same platform is closer, for example, New York and Tokyo collected from Foursquare and Los Angeles and Boston collected from Gowalla. The New York and Tokyo datasets from Foursquare and the Los Angeles and Houston datasets from Gowalla have more users with less than 100 check-ins. About 90% of users in four city datasets have less than 600 check-ins frequency.

*5.2. Baselines.* To demonstrate the effectiveness of our model, we compare to the following location prediction methods:

(i) STRNN [3]. It is an extended RNN method that can model local temporal and spatial contexts.

(ii) DeepMove [6]. It is a recent location prediction method to learn user periodical patterns with attention mechanism and the recurrent neural networks.

(iii) VANext [11]. It is a location prediction method for variational attention mechanism and leverage CNN to learn historical mobility RNN to learn recent check-in preference.

(iv) Flashback [30]. It is a general RNN architecture design for modeling sparse user mobility traces by doing flashbacks on hidden states in RNNs.

(v) STSAN is a model we proposed, which can integrate spatial-temporal information with the self-attention for location prediction.

*5.3. Evaluation Metric.* Given the trajectory of the user before the current time, location prediction aims at predicting the next location of a user. Intuitively, a good model is to be able to restore the actual ground records more realistically. Hence, we use the prediction accuracy [6, 31] to measure the performance.

Formally, given the user set $\mathbf{U}$ and the candidate locations set $\mathbf{C}$, for each user, and given the initial location $v_1$, we predicted the next continuous $L^u - 1$ locations. The prediction accuracy is calculated as follows:

$$\mathrm{Acc@}k = \frac{1}{|U|}\sum_{u_i \in U}\frac{1}{L^{u_i}-1}\sum_{t=2}^{L^{u_i}}\left(v_t \in S_t^{u_i}(k)\right), \tag{21}$$

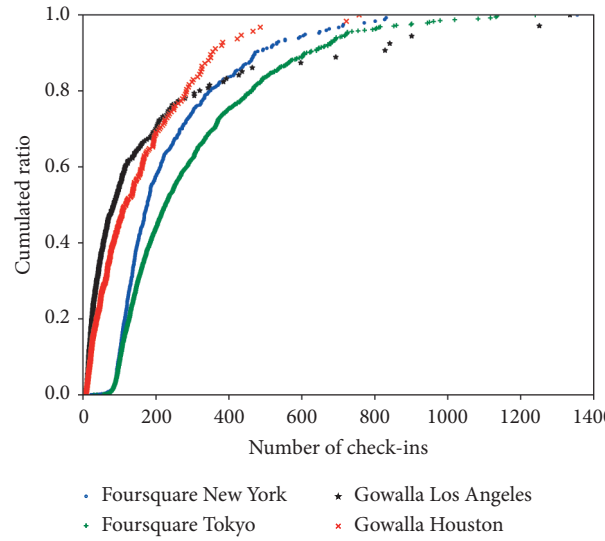



Figure 3: Cumulative distribution function of check-in number.

where $S_t^{u_i}(k)$ is a set of top-$k$ locations for user $u_i$ in time step $t$, $v_t$ is real visited location for user $u_i$ in time step $t$, and $L^{u_i}$ is the history trajectory length for user $u_i$. When $v_t$ in the set $S_t^{u_i}(k)$, the expression value is 1; otherwise, it is 0.

Moreover, similar to the previous work, we apply average percentage rank (APR) [27, 32] as another metric to measure the overall ranking performance of the model. The average percentage rank (APR) is calculated as follows:

$$\mathrm{APR} = \frac{1}{|U|}\sum_{u_i \in U}\frac{1}{L^{u_i}-1}\sum_{t=2}^{L^{u_i}}\frac{|V| - \mathrm{rank}^{u_i}(v_t) + 1}{|V|}, \tag{22}$$

where $\mathrm{rank}^{u_i}(v_t)$ denotes the rank of candidate location $v_t$ for user $u_i$ in timestamp $t$ after sorting $|V|$ locations in decreasing order.

*5.4. Implementation Details.* We partition each dataset into a training set and a test set. For each user, we use the first 80% subtrajectory as the training data and the remaining 20% as the test data. We implement our model with PyTorch. All experiments are conducted on an NVIDIA Tesla P4 GPU and a 64G CPU on the Ubuntu system. We use RMSProp adaptive learning rate optimization algorithm [33] and the Reduce LR On Plateau learning rate adaptation method. We clip the $L2$ norm of vector composed of several parameters of gradient to alleviate the problem of gradient explosion. To prevent overfitting, we use $L1$ regularization, weight decay, and dropout [34] to improve the performance of the neural network by preventing the interaction of feature detectors. For the objective function, we set the weighting factors $\lambda_1 = 1e-4$, $\lambda_2 = 1e-5$, dropout rate 0.5, the initial learning rate to $2e-5$, the embedding dimension of location $d_p = 512$, the hidden size $d = 512$, distance threshold $\varepsilon = 20$, spatial factor $\beta = 1.0$, temporal factor $\gamma = 1.0$, the decay of learning rate 0.1, gradient clip 1.0, and the max epoch 50.

*5.5. Performance Comparison.* We evaluate our model with the baseline methods on four city check-ins datasets to present the performance of our model. Different parameter settings do have an impact on the final performance of the model. Table 2 shows the basic parameter settings of the STSAN model on four datasets when compared with the baselines, and the performance comparison is shown in Figure 4.

Our proposed model STSAN achieves the best performance on four datasets with all evaluation metrics, which illustrates the superiority of our model. Take the New York City dataset; for example, STSAN achieves 39.8% on Acc@1, 51.6% on Acc@5, 53.3% on Acc@10, and 4.4% on APR over VANext [11], which performs the second best. STRNN is an RNN model which incorporates spatial-temporal context. However, these methods do not explicitly model user's historical visit patterns. STRNN has employed recurrent networks to capture long-term POI dependency, and it cannot deal with very long history trajectories well because of the inherent gradient disappearance problem of recurrent networks. Therefore, the main advantage of DeepMove is that it leverages attention network on historical trajectories. Previous techniques try to improve RNNs by considering spatial-temporal context. The context-parameterized transition matrices or gates are used to fuse the spatial-temporal context to simplify the temporal periodicity and spatial law. Flashback [30] is to model the sparse user movement trajectory by flashbacking the hidden state in RNN. VANext not only effectively captures short-term human mobility patterns but also leverages variable attention and CNN networks to generate better historical trajectory representation. It significantly outperforms these previous methods. Therefore, it achieves the best performance than all previous methods. From the results, the performance of the two more sparse datasets (Los Angeles and Houston) is significantly worse. The reason is the model may not be able to get enough data to train, which leads to overfitting problem. For our model STSAN, the performance on sparse datasets has achieved better results on Acc@1. From the latter analysis, we know that the spatial attention module of the model plays a great role.

*5.6. Impacts of Spatial and Temporal Attention.* To study the influence of temporal and spatial characteristics on the framework, we split the model into three variants: (1) spatial self-attention network (SSAN), which is the network structure without considering the temporal attention module; (2) temporal self-attention network (TSAN), which is the network structure without considering the spatial attention module; (3) self-attention network (SAN), which is the network structure without considering the spatial attention and temporal attention module.

Both Tokyo and New York from foursquare perform similarly, and Houston and Los Angeles from Gowalla perform similarly. Therefore, limited to the length of the article, the article only gives two cities as representatives. Figure 5 shows the performance of STSAN and three variants on the New York and Los Angeles datasets. For the New York dataset, we can see that, with the increase in the epoch, the performance of the four models keeps improving, and after about 10 epochs, they tend to be stable. The performance of TSAN leads SSAN and SAN, so it can be seen that time information is more important than space information on the New York dataset. What is more, we are surprised that STSAN and SSAN both add spatial attention based on the original model, but STSAN brings more performance improvement. When time information and space information are added to the model at the same time, the result of one plus one is greater than two. Finally, we can get the exact conclusion that the performance is STSAN > TSAN > SSAN > SAN on the New York dataset. For the Los Angeles dataset, SSAN leads TSAN and SAN in performance, but TSAN converges faster. Finally, the performance is STSAN > SSAN > TSAN > SAN on the Los Angeles dataset. For different city datasets, spatial and temporal information is not the same for the performance of the model. The reason may be that the spatial information is too complex and not captured well for the New York dataset with more candidate locations.

Table 2: Parameters setting on four datasets.

| | $\gamma$ | $\beta$ | $\varepsilon$ | $H$ | $d$ |
|---|---|---|---|---|---|
| New York | 1.0 | 1.0 | 40 | 8 | 512 |
| Tokyo | 1.0 | 1.0 | 20 | 32 | 768 |
| Los Angeles | 0.8 | 0.8 | 50 | 2 | 704 |
| Houston | 0.8 | 0.9 | 60 | 1 | 640 |

The STSAN has achieved the best performance on all datasets, which proves the importance of using spatial-temporal information modeling. On the one hand, in sparse datasets, spatial factors are more important for model performance improvement. On the other hand, the time factor can help the model converge quickly.

*5.7. The Effect of Hyperparameter Settings*

*5.7.1. Effect of $\varepsilon$ and $\beta$.* The distance threshold $\varepsilon$ controls the influence of geographic distance on our model. In general, Figure 6(a) shows that the performance of the model on the four datasets has improved with the increase in distance and the performance improvement on sparse datasets (Los Angeles and Houston) is more significant. We can see that users in different cities have different preferences for distance. Under the default parameter settings, it is more suitable to set $\varepsilon$ to 30 and 60 on Los Angeles and Houston datasets, respectively. The spatial factor $\beta$ controls the influence degree of geographic information on our model. We can see that the accuracy increases in different degrees with the growth of $\beta$ in the four datasets. The spatial geographic location information is very helpful to improve the accuracy of the model. We also successfully capture it with spatial attention.

*5.7.2. Effect of $\gamma$.* Figure 6(b) shows the APR and Acc@10 for various $\gamma$ that control the influence of degree of temporal information while keeping other optimal hyperparameters

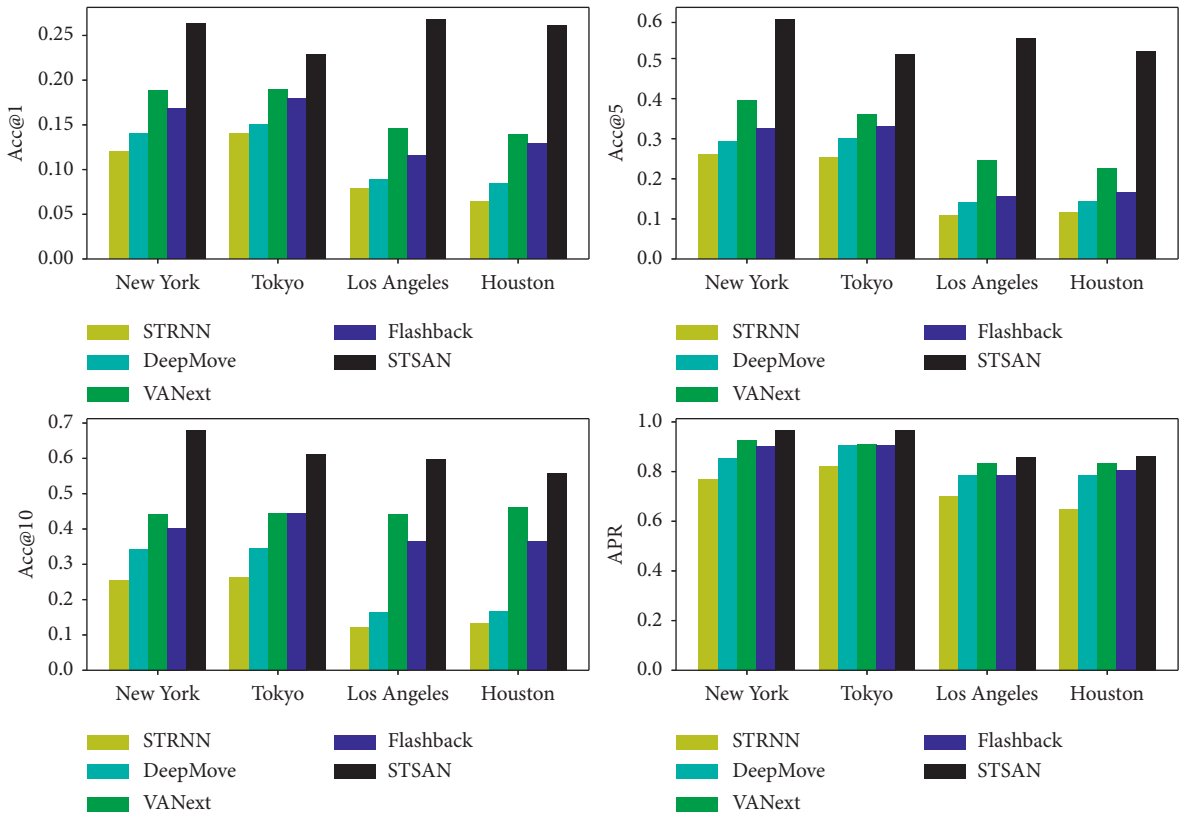



Figure 4: Performance compared with the baselines on the four datasets.

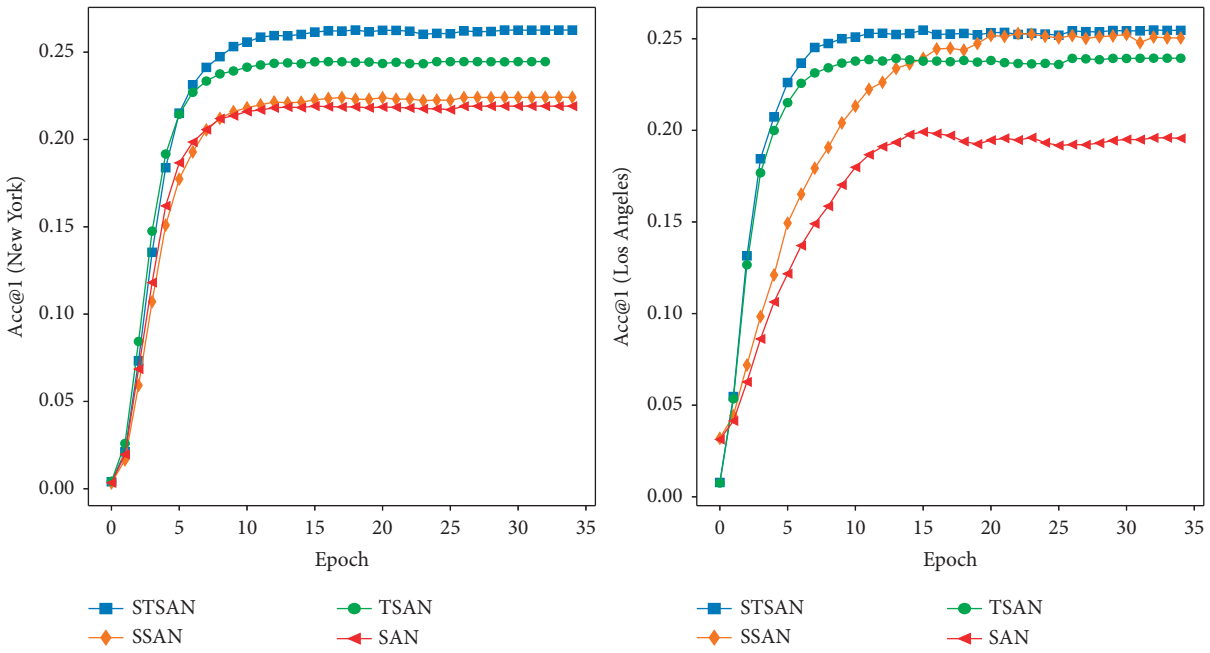
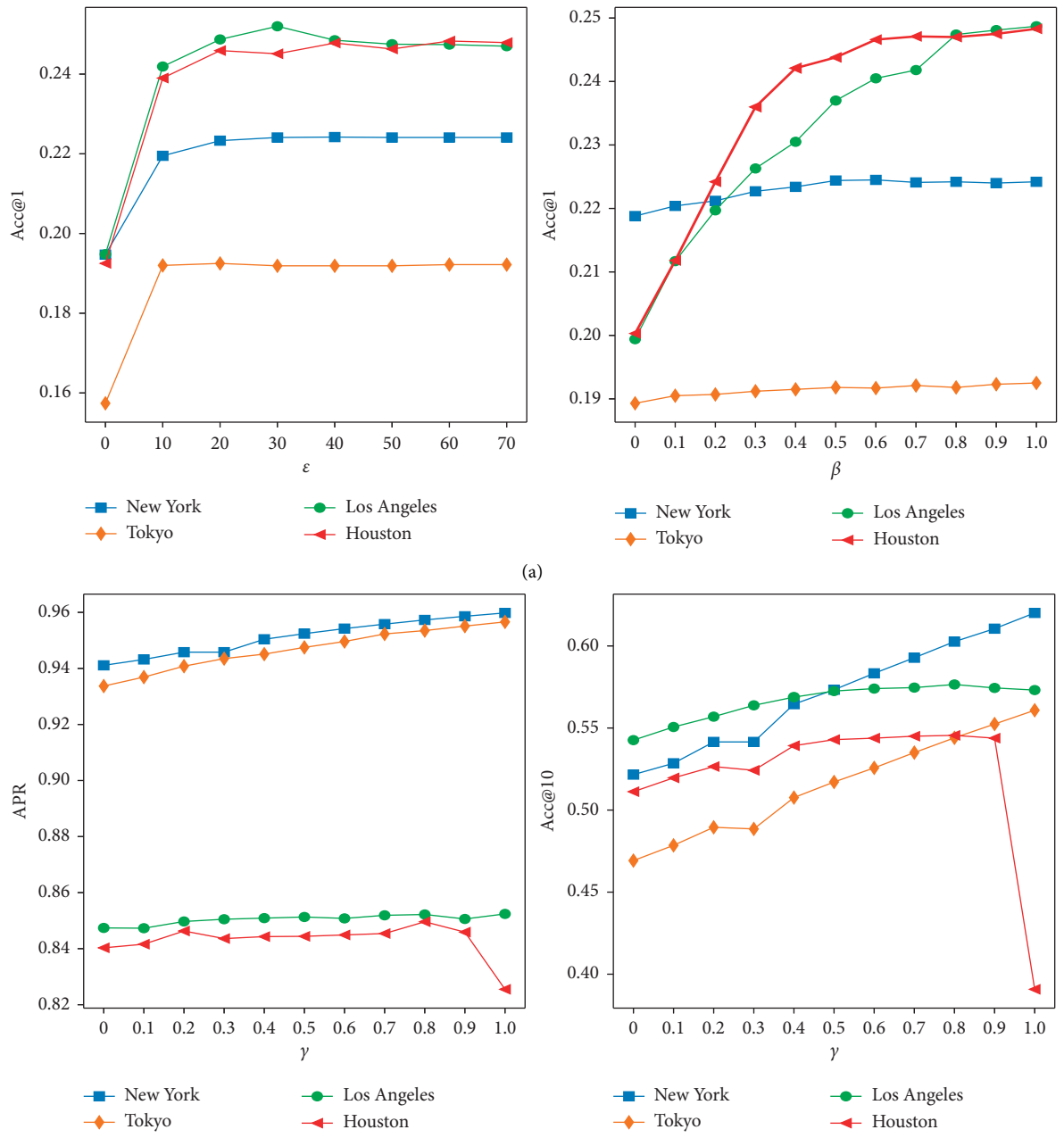



Figure 5: Impacts of spatial and temporal attention.

unchanged. As the factor $\gamma$ grows, our model performance grows on the Foursquare datasets. However, it can be seen that, in the Gowalla datasets, the upper limit of model performance appears with the increase in factor $\gamma$, and when $\gamma$ takes 1.0 in the Houston dataset, there is a significant decline, which shows that $\gamma$ has an optimal value 0.9. The reason for this phenomenon may be that the Gowalla datasets are more sparse and the time span is larger than that

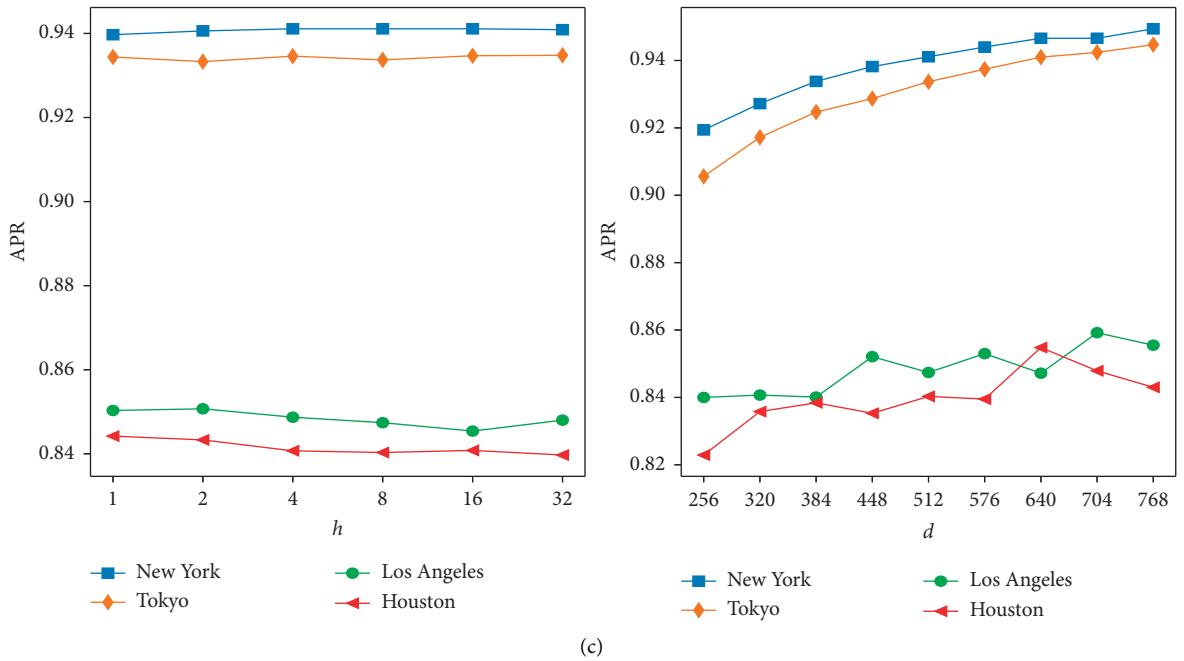



(a)




(b)

Figure 6: Continued.




(c)

Figure 6: Effect of hyperparameter settings: (a) effect of $\varepsilon$ and $\beta$; (b) effect of $\gamma$; (c) effect of $h$ and $d$.

of Foursquare, which makes the mining of time information more difficult.

*5.7.3. Effect of h and d.* Figure 6(c) shows the APR performance on four datasets when we vary parameters $h$ and $d$, respectively. The projection of multiple subspaces can improve the performance, but not obvious, and the optimal $h$ is different in different datasets. On the Foursquare dataset, with the increase in the hidden layer dimension, the accuracy of the model shows an upward trend. However, on the Gowalla dataset, the larger dimension of the model does not bring better accuracy. We analyze that the reason for this may be that more parameters are added to the dimension, which makes the model appear as overfitting problem earlier on the more sparse datasets.

## 6. Conclusions

In this paper, we proposed a novel network named STSAN (spatial-temporal self-attention network), which can integrate spatial and temporal information with the self-attention for location prediction. Our model can learn the dynamic representation of the user's trajectory by capturing the sequential transitions pattern of the user's trajectory and integrating the user's geographical preference and time correlation. It makes better use of spatial and temporal information to mine the user's trajectory, which is helpful to improve the accuracy of location prediction on sparse data and alleviates the problem of data sparsity. We experimentally evaluate our STSAN model using four datasets collected from Foursquare and Gowalla. The experiment results show the superiority of our approach over various baseline approaches, and it is more prominent in sparse datasets. In addition, we verified the influence of various parameters in STSAN on experimental performance through a large number of experimental results. STSAN introduces the self-attention mechanism and integrates spatiotemporal information, but it introduces a lot of relevant parameters to increase the difficulty in parameter adjustment. In the future, we plan to explore the application of federated machine learning on location prediction and extend our STSAN model to a privacy-aware version. In addition, more contextual information such as knowledge graph is also worth being incorporated into our model in order to enhance the interpretability of the model.

## Data Availability

Foursquare dataset contains check-ins in NYC and Tokyo collected for about 10 months (from 12 April 2012 to 16 February 2013). It contains 227,428 check-ins in New York City and 573,703 check-ins in Tokyo. Each check-in is associated with its timestamp, its GPS coordinates, and (https://www.kaggle.com/chetanism/foursquare-nyc-and-tokyo-checkin-dataset) Gowalla dataset. This dataset is collected for about 18 months (from 1 February 2009 to 31 October 2010). It contains 6,442,890 check-ins with its timestamp and its GPS coordinates. We select the check-ins in Los Angeles and Houston as experiment dataset (http://snap.stanford.edu/data/loc-gowalla.html).

## Conflicts of Interest

The authors declare that there are no conflicts of interest.

## Acknowledgments

This research was partially funded by the Natural Science Foundation of Liaoning Province (Grant no. 364 2019-MS-

111) and the National Natural Science Foundation of China (Grant no. 61872069).

## References


[1] A. Asahara, K. Maruyama, A. Sato, and K. Seto, "Pedestrian-movement prediction based on mixed markov-chain model," in *Proceedings of the 19th ACM SIGSPATIAL International Symposium on Advances in Geographic Information Systems, ACM-GIS 2011*, pp. 25–33, ACM, Chicago, IL, USA, November 2011.

[2] S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme, "Factorizing personalized Markov chains for next-basket recommendation," in *Proceedings of the 19th International Conference on World Wide Web, WWW 2010*, pp. 811–820, Raleigh, NC, USA, April 2010.

[3] Q. Liu, S. Wu, L. Wang, and T. Tan, "Predicting the next location: a recurrent model with spatial and temporal contexts," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pp. 194–200, AAAI Press, Phoenix, AZ, USA, February 2016.

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

[5] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," 2014, https://arxiv.org/abs/1412.3555.

[6] J. Feng, Y. Li, C. Zhang et al., "Deepmove: predicting human mobility with attentional recurrent networks," in *Proceedings of the 2018 World Wide Web Conference on World Wide Web, WWW 2018*, pp. 1459–1468, Lyon, France, April 2018.

[7] Z. Chang, J. Bai, J. Song et al., "An attention-based user behavior modeling framework for recommendation," in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18)*, pp. 4564–4571, AAAI Press, New Orleans, LO, USA, February 2018.

[8] D. Kong, F. Wu, and HST-LSTM, "A hierarchical spatial-temporal long-short term memory network for location prediction," in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018*, pp. 2341–2347, Stockholm, Sweden, July 2018.

[9] Di Yao, C. Zhang, J.-H. Huang, and S. E. R. M. Jingping Bi., "A recurrent model for next location prediction in semantic trajectories," in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, CIKM 2017*, pp. 2411–2414, ACM, Singapore, November 2017.

[10] A. Vaswani, N. Shazeer, N. Parmar et al., "Attention is all you need," in *Proceedings of the Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017*, pp. 5998–6008, Long Beach, CA, USA, December 2017.

[11] Q. Gao, Z. Fan, G. Trajcevski, K. Zhang, T. Zhong, and F. Zhang, "Predicting human mobility via variational attention," in *Proceedings of the The World Wide Web Conference, WWW 2019*, pp. 2750–2756, ACM, San Francisco, CA, USA, May 2019.

[12] S. Feng, X. Li, Y. Zeng, C. Gao, Y. Meng Chee, and Y. Quan, "Personalized ranking metric embedding for next new POI recommendation," in *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015*, pp. 2069–2075, AAAI Press, Buenos Aires, Argentina, July 2015.

[13] A. Al-Molegi, M. Jabreel, B. Ghaleb, and STF-RNN, "Space time features-based recurrent neural network for predicting people next location," in *Proceedings of the 2016 IEEE Symposium Series on Computational Intelligence, SSCI 2016*, pp. 1–7, IEEE, Athens, Greece, December 2016.

[14] A. Li, S. Wang, W. Li, S. Liu, and S. Zhang, "Predicting human mobility with federated learning," in *Proceedings of the SIGSPATIAL '20: 28th International Conference on Advances in Geographic Information Systems*, pp. 441–444, Seattle, WA, USA, November 2020.

[15] V. Mnih, N. Heess, A. Graves, and K. Kavukcuoglu, "Recurrent models of visual attention," *Proceedings of the Advances in Neural Information Processing Systems*, vol. 27, pp. 2204–2212, 2014.

[16] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," 2014, https://arxiv.org/abs/1409.0473.

[17] M.-T. Luong, H. Pham, and D. Christopher, "Effective approaches to attention-based neural machine translation," 2015, https://arxiv.org/abs/1508.04025.

[18] W. Yin, H. Schütze, B. Xiang, and B. Zhou, "ABCNN: attention-based convolutional neural network for modeling sentence pairs," 2015, https://arxiv.org/abs/1512.05193.

[19] P. Zhou, W. Shi, J. Tian et al., "Attention-based bidirectional long short-term memory networks for relation classification," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, Berlin, Germany, August 2016.

[20] S. Zhang, Yi Tay, L. Yao, and A. Sun, "Dynamic intention-aware recommendation with self-attention," 2018, https://arxiv.org/abs/1808.06414v1.

[21] W.-C. Kang and J. J. McAuley, "Self-attentive sequential recommendation," in *Proceedings of the IEEE International Conference on Data Mining, ICDM 2018*, pp. 197–206, IEEE Computer Society, Singapore, November 2018.

[22] M. Xia, Y. Cui, Y. Zhang, Y. Xu, J. Liu, and Y. Xu, "Dau-net: a novel water areas segmentation structure for remote sensing image," *International Journal of Remote Sensing*, vol. 42, no. 7, pp. 2594–2621, 2021.

[23] M. Xia, T. Wang, Y. Zhang, J. Liu, and Y. Xu, "Cloud/shadow segmentation based on global attention feature fusion residual network for remote sensing imagery," *International Journal of Remote Sensing*, vol. 42, no. 6, pp. 2022–2045, 2021.

[24] B. Chen, M. Xia, and J. Huang, "Mfanet: a multi-level feature aggregation network for semantic segmentation of land cover," *Remote Sensing*, vol. 13, no. 4, p. 2021.

[25] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proceedings of the Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013*, pp. 3111–3119, Lake Tahoe, NA, USA, December 2013.

[26] J. Cao, S. Xu, X. Zhu, R. Lv, and Bo Liu, "Efficient fine-grained location prediction based on user mobility pattern in lbsns," in *Proceedings of the Fifth International Conference on Advanced Cloud and Big Data, CBD*, pp. 238–243, IEEE Computer Society, Shanghai, China, August 2017.

[27] D. Yang, D. Zhang, V. W. Zheng, and Z. Yu, "Modeling user activity preference by leveraging user spatial temporal characteristics in LBSNS," *IEEE Transactions on Systems, Man, and Cybernetics Systems*, vol. 45, no. 1, pp. 129–142, 2015.

[28] E. Cho, S. A. Myers, and J. Leskovec, "Friendship and mobility: user movement in location-based social networks," in *Proceedings of the 17th ACM SIGKDD International*

*Conference on Knowledge Discovery and Data Mining*, pp. 1082–1090, ACM, San Diego, CA, USA, August 2011.

[29] H. Kwak, C. Lee, H. Park, B. Sue, and Moon, "What is twitter, a social network or a news media?" in *Proceedings of the 19th International Conference on World Wide Web, WWW 2010*, pp. 591–600, Raleigh, North Carolina, USA, April 2010.

[30] D. Yang, B. Fankhauser, P. Rosso, and P. Cudré-Mauroux, "Location prediction over sparse user mobility traces using rnns: Flashback in hidden states! in Christian Bessiere," in *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI*, pp. 2184–2190, New York, NY, USA, July 2020.

[31] C. Zhang, K. Zhang, Y. Quan, L. Zhang, Tim Hanratty, and J. Han, "Gmove: group-level mobility modeling using geo-tagged social media," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1305–1314, ACM, San Francisco, CA, USA, August 2016.

[32] A. Noulas, S. Scellato, L. Neal, and C. Mascolo, "Mining user mobility features for next place prediction in location-based services," in *Proceedings of the 12th IEEE International Conference on Data Mining, ICDM 2012*, pp. 1038–1043, IEEE Computer Society, Brussels, Belgium, December 2012.

[33] G. Hinton, N. Srivastava, and S. Kevin, "Neural networks for machine learning lecture 6a overview of mini-batch gradient descent," *Overview of Mini-Batch Gradient Descent*, vol. 14, no. 8, 2012.

[34] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," 2012, https://arxiv.org/abs/1207.0580.