An Attention-Based Spatiotemporal GGNN for Next POI Recommendation

QUAN LI, XINHUA XU, XINGHONG LIU, AND Qi CHEN
Department of Computer and Information Engineering, Hubei Normal University, Huangshi Hubei 435002, China
Corresponding author: Quan LI (e-mail: cleverlq@hotmail.com).

This work was supported in part by the Science and Technology Research Project of Hubei Department of Education under Grant D20212503, and in part by the Research Project of Young Teachers in Hubei Normal University under Grant HS2020QN029.

ABSTRACT
The task of Point-of-Interest (POI) recommendation is to recommend the next interest locations for users. Gated Graph Neural Network (GGNN) has been proved to be effective on POI recommendation tasks. However, existing GGNN solutions rarely consider the spatiotemporal information between nodes in the sequence graph, which is essential for modeling user check-in behaviors in next POI recommendation. In this paper, we propose an attention-based spatiotemporal gated graph neural network model (ATST-GGNN) for next POI recommendation. Firstly, the user’s check-in sequence is represented as a graph structure. Secondly, we use spatiotemporal context information to dynamically update nodes in the sequence graph, and obtain the complex transfer relationships between the check-ins. Thirdly, each session is then represented as the composition of the long and short preference using an attention network. However, current short preference fails to model union-level sequential patterns, we improve the local embedding representation of graph nodes by window pooling method, as well as the global embedding representation of graph nodes by integrating it into attention mechanism. Finally, the objective function is constructed by cross entropy and the model parameters are learned. The experimental results show that the precision rate and mean reciprocal ranking of ATST-GGNN method are greatly improved compared with the state-of-art methods. It has good application prospect.

INDEX TERMS
POI recommendation; Gated graph neural network; Window pooling; Attention; Cross entropy.

I. INTRODUCTION

WITH the rapid development of smart phones and global positioning systems, the location positioning function provided by intelligent terminals is becoming more and more accurate. Under this background, the location-based social network (LBSN) service has been developed rapidly, and has been loved by lots of users, such as Gowalla, Yelp and Facebook Places and so on. Compared with traditional social networks, the advantage of LBSN is that users can publish their geographic check-in information in the form of check-ins, and share their experiences with friends, such as: gymnasium, coffee shop and restaurant, etc. Points of interest (POI) recommendation can help users find information of their own interest in the massive datas of location social network, and access new geographical locations and facilitate their life. So the task of POI recommendation is to recommend the next interest location for users [1], [2], [3], [4].

The traditional POI recommendation algorithm generally adopts matrix factorization. This method decomposes the user’s check-in datas, and extracts the latent feature matrices of the users and the POIs, and make the recommendation for user [5], [6], [7]. However, the user’s check-in datas are sequential. The current solution generally uses recurrent neural network (RNN) to analyze check-in sequence data [8], such as long-short term memory (LSTM) [9] and gated recurrent unit (GRU) [10]. Meanwhile, the current research shows that these methods can only model the one-way transitions of adjacent check-in points in sequences, while ignoring the other check-in ones. The current solution is to convert the user’s check-in sequence to a graph, update the nodes of sequence graph by the gated graph neural network (GGNN), so as to obtain the complex transitions between the different check-in points [11]. However, there are still some problems with the method. Firstly, GGNN did not consider the spatial and temporal relationships between nodes in the
sequence graph. Secondly, every POI session is represented as the composition of the long and short preference using an attention network. However, current short preference means the last location, and fail to model union-level sequential patterns.

To address the problem mentioned above, this paper proposed an attention-based spatiotemporal gated graph neural network (ATST-GGNN) for next POI recommendation. The model can further improve the accuracy of POI recommendation. The main contributions are as follows:

1) A spatiotemporal GGNN model is proposed. The method firstly convert the user’s check-in sequence to a graph. Secondly, three kinds of contextual informations are integrated into the GGNN model. Finally, every node of sequence graph is dynamically updated.

2) A window-based pooling method is proposed to obtain the short-term representation of user, and every POI sequence is represented as the composition of the long and short preference using an attention network.

3) We conduct empirical studies on two real-world datasets of Foursquare and Gowalla. Experimental results show that the proposed ATST-GGNN outperforms the state-of-art methods. For example, compared with next POI recommendation model, the P@5, P@10, P@20, MRR@5, MRR@10 and MRR@20 value of ATST-GGNN are increased.

The remainder of this paper is organized as follows. Section 2 reviews the works related to the topics of collaborative filtering recommendation, neural network recommendation, sequence recommendation and graph neural network recommendation. Section 3 introduces the preliminaries to our study. Section 4 introduces the details of the proposed attention-based spatiotemporal GGNN network. Section 5 describes the experimental environment and results. Finally, Section 6 concludes this paper and outlines our future work.

II. RELATED WORK
A. COLLABORATIVE FILTERING RECOMMENDATION
Collaborative filtering recommendation include user-based collaborative filtering [12], item-based collaborative filtering [13] and matrix factorization recommendation [14], [15] etc. Zhang et al. [16] proposed personalized and efficient geographical location recommendation framework (iGeoRec). They could predict the probability of a user visiting any new location using his personal distribution, and developed an efficient approximation method to compute the probability of any user to all new locations. Zhao et al. [17] proposed a POI mining method, and a personalized recommendation model by fusing sentimental spatial context. The sentimental–spatial POI mining method is utilized to mine the POIs by fusing the sentimental and geographical attributes of locations. The sentimental–spatial POI Recommendation model incorporates the factors of sentiment similarity and geographical distance. Lian et al. [18] proposed a scalable and flexible framework, dubbed GeoMF++, for joint geographical modeling and implicit feedback-based matrix factorization. They then developed an efficient optimization algorithm for parameter learning, which scales linearly with data size and the total number of neighbor grids of all locations. Gao et al. [19] proposed a POI recommendation algorithm by fusing time influence and matrix factorization. They introduced four time aggregation strategies, and integrated different time states into the user’s check-in preference.

B. NEURAL NETWORK RECOMMENDATION
Neural network includes fully connected neural network [20], convolutional neural network [21] and recurrent neural network [8], etc. Machine translation [22], image classification [23] and recommendation system [24], [25] can be realized through neural network. Yang et al. [26] bridged collaborative filtering and semi-supervised learning. The potential characteristics of users and POI are learnt in deep neural network, and the preferences of users for the next POI are predicted. Xing et al. [27] proposed a POI recommendation model based on convolution matrix factorization by convolution neural network and probability matrix factorization. The model combines user check-in information, social relationships and comment information etc. Ma et al. [28] proposed a POI recommendation algorithm combining self-attentive coding and neighbor-aware decoding. Self-attentive coding uses multi-dimensional attention mechanism to analyze user preferences from different aspects. Neighbor-aware decoding can take advantage of geographic context information.

C. SEQUENCE RECOMMENDATION
With the development of POI recommendation, the next POI recommendation based on sequence information has become a new hot topic. Cheng et al. [29] proposed a novel matrix factorization method, namely factoring personalized markov chain and region localization (FPMC-LR). It exploits tensor decomposition to recommend POIs being close to the user’s geographical position. Feng et al. [30] proposed a personalized ranking metric embedding method (PRME). It maps each location to a low-dimensional space to calculate the transfer probability between locations, and considers two influencing factors: user preference and timing transfer. With the development of natural language processing technology, recurrent neural network has been successfully applied in the fields of text classification and machine translation. Liu et al. [31] proposed a recommendation method based on spatial temporal recurrent neural network (ST-RNN). It combines time information and geographic information by time and distance transfer matrix. Zhao et al. [32] proposed a variant of LSTM, named a spatial temporal LSTM model (ST-LSTM), which implements time gates and distance gates into LSTM to capture the spatiotemporal relation between check-ins. Lian et al. [33] proposed a geography-aware sequential recommender based on the self-attention network (GeoSAN) for location recommendation. They represented the hierarchical gridding of each GPS point with a self-attention based geography encoder, and put forward geography-aware negative samplers to promote the informativeness of neg-
ative samples. Hao et al. [34] proposed an annular-graph attention based sequential recommendation model (AGSRR) by exploring user’s long-term and short-term preferences for the personalized sequential recommendation. Huang et al. [35] proposed attention-based spatiotemporal LSTM POI recommendation algorithm (ATST-LSTM). It combines time and space information on the basis of LSTM, and extracts effective information by soft-attention. Wu et al. [36] proposed a novel method named personalized long and short term preference learning (PLSPPL) to learn the specific preference for each user. They combined the long and short term preference via user-based linear combination unit to learn the personalized weights on different parts for different users.

D. GRAPH NEURAL NETWORKS RECOMMENDATION

With the development of neural network, graph neural network can process graph structure data, including social network, knowledge graph, and so on. A graph representation learning algorithm DeepWalk is designed to learn vector representations of graph nodes based on random walk [37]. Unsupervised network embedding algorithms LINE is designed to extract any information of graph structure, including undirected graph, directed graph and weight graph, etc [38]. In addition, graph neural network can also be applied to recommendation system. Wu et al. [11] proposed session-based recommendation with graph neural networks (SR-GNN). The session sequences are modeled as graph structure. The vector representation of nodes in the graph is obtained by gated graph neural network. Finally, the soft-attention mechanism is combined to realize the session recommendation. Xu et al. [39] proposed a graph contextualized self-attention model (GC-SAN), which utilizes both graph neural network and self-attention mechanism for session-based recommendation. However, there exist some challenges preventing graph neural networks from becoming the best solution for next POI recommendation.

First of all, graph neural networks fail to model the spatial and temporal relationships between nodes in the sequence graph. They can be very useful for learning the node representations in the sequence graph by using the context information [40], [41], [42]. Secondly, attention mechanism has been designed for long-term user’s preference. Zhao et al. [32] reported that users’ short-term and long-term preference are both significant on achieving the best performance. However the short-term preference means that recommended POIs should depend on the last visited POI, and it fail to model union-level sequential patterns. In fact, several last actions jointly influence the target action [43].

To this end, in this paper, we propose an attention-based spatiotemporal GGNN, named ATST-GGNN. The user’s check-in sequence graph based on GGNN combines the spatiotemporal contextual information, and the local embedding representation of graph nodes by window pooling method is improved , as well as the global embedding representation of graph nodes by integrating it into attention mechanism. The proposed method can better improve the accuracy of POI recommendation.

III. PRELIMINARIES TO THIS STUDY

A. NOTATIONS AND DEFINITIONS

Notations are summarized in Table 1. 

Definition 1 (POI). In LBSNs, a point-of-interest (POI) is a geographical location associated with spatial and temporal information.

Definition 2 (Check-in activity). A user’s check-in activity is represented as \( v^u_t \) which stands for visiting the location \( v \) for user \( u \) at time point \( t \).

Definition 3 (Check-in sequence). A check-in sequence of a user \( u \) is a set of check-in activities of the user, denoted by \( S^u = \{ v^u_t \}_{t=1}^{\infty} \).

Definition 4 (Check-in sequence graph). A check-in sequence graph of a sequence is sets of check-ins and edges, denoted by \( G^u = (Q^u, E^u) \), \( Q^u = \{ q^u_i \}_{i=1}^{|Q^u|} \) stands for set of check-ins. \( E^u \) stands for set of edges between two adjacent nodes in sequence graph, it means that the next location is \( v^u_{t+1} \) after visiting location \( v^u_t \).

The primary goal of this study is to offer target user a list of interested locations that the user is likely to visit at the next time.

| Notation | Description |
|----------|-------------|
| \( U, V \) | the set of users and locations |
| \( C \) | the set of historical check-ins |
| \( S^u \) | check-in sequence set of user \( u \) |
| \( D^u \) | distance interval set of location \( u \) |
| \( T^u \) | time interval set of location \( u \) |
| \( G^u \) | check-in sequence graph of user \( u \) |
| \( Q^u \) | check-ins set, edges set of sequence graph of user \( u \) |
| \( E^u \) | incoming edge and outgoing edge |
| \( T^u_{A^u_i} \) | weight of edge based on time interval |
| \( A^u_i \) | weight of edge based on distance interval |
| \( \Delta t_{x,y} \) | time interval and distance interval between locations \( (x,y) \) |
| \( tt, dd \) | neighbor connection matrix, time connection matrix, spatial connection matrix |
| \( q^u_i \in \mathbb{R}^d \) | node embedding vector of sequence graph of user \( u \) |
| \( a^u_i, a^u_{D^u_i} \) | input vectors of confusing neighbor, time and spatial information |
| \( r^u_t, z^u_t \) | reset gate and update gate |
| \( \gamma_x \) | hyperparameter |
| \( s_l, s_d \) | local and global embedding representation |
| \{ \mathbf{W} \} | set of weight matrices for ATST-GGNN network |
| \{ \mathbf{b} \} | set of bias vectors for ATST-GGNN network |

B. GATED GRAPH NEURAL NETWORK

The primary challenge of the next POI recommendation problem is learning personalized user preference for POIs. As we know, a favorite choice is RNN, LSTM or GRU architecture. However, how to obtain transfer relationship between the check-ins in a check-in sequence and learn
the embedding representations of graph-structured data are important problems needed to be solved. As the standard GNN architecture can automatically extract features of sequence graph with considerations of rich node connections [44], in this study, we use it as a building block for next POI recommendation. Formally, the update functions of graph nodes are given as follows:

\[
\begin{align*}
\text{a}_i^t &= \text{Concat}\left(\text{A}_{\text{in}}^i[q_1^i, q_2^i, \ldots, q_n^i], \text{A}_{\text{out}}^i[q_1^i, q_2^i, \ldots, q_n^i]\right) \text{W}_{\text{in}},
\text{A}_{\text{out}}^i[q_1^i, q_2^i, \ldots, q_n^i] \text{W}_{\text{out}}\right) \\
\text{r}_i^t &= \sigma\left(\text{W}_{\text{ra}} \text{a}_i^t + \text{W}_{\text{rq}} q_{i-1}^t + b_r\right) \\
z_i^t &= \sigma\left(\text{W}_{\text{za}} \text{a}_i^t + \text{W}_{\text{zq}} q_{i-1}^t + b_z\right) \\
\hat{q}_i^t &= \tanh\left(\text{W}_{\text{hr}}(r_i^t \odot q_{i-1}^t) + \text{W}_{\text{hq}} a_i^t + b_q\right) \\
q_i^t &= z_i^t \odot \hat{q}_i^t + (1 - z_i^t) \odot q_{i-1}^t
\end{align*}
\]

Where the \( \text{W}_{\text{in}}, \text{W}_{\text{out}}, \text{W}_{\text{ra}}, \text{W}_{\text{rq}}, \text{W}_{\text{za}}, \text{W}_{\text{zq}}, \text{W}_{\text{hr}}, \) and \( \text{W}_{\text{hq}} \in \mathbb{R}^{d \times d} \) are parameter matrices. \([q_1^i, q_2^i, \ldots, q_n^i]\) is the list of node vectors in check-in sequence. \( \text{a}_i^t \) is the input vector of GGNN model. The vector \( r_i^t \) and \( z_i^t \) stand for reset gate and update gate, respectively. The \( q_{i-1}^t \) and \( q_i^t \) stand for input vector of previous time \( t - 1 \) and input vector of current time \( t \), respectively. \( \sigma(x) = \frac{1}{1 + e^{-x}} \), which controls the open and close of reset and update gate. The \( b_r, b_z \) and \( b_q \) are all the bias vectors. The symbol \( \odot \) is the hadamard operations based on matrix elements. The \( r_i^t \odot q_{i-1}^t \) controls the former information retained by the reset gate \( r_i^t \) in the formula (4). The left part controls the current information of input by the \( z_i^t \), the right part controls the former retained information by the \( 1 - z_i^t \) in the formula (5).

The \( \text{A}_{\text{in}} \) and \( \text{A}_{\text{out}} \) are defined as the \( i^{th} \) row of neighbor connection matrix \( \text{A} = [\text{A}_{\text{in}}, \text{A}_{\text{out}}] \), which represents weighted connections of sequence graph. For example, consider a sequence \( S_i = \{v_1, v_2, v_3, v_4, v_5, v_6\} \), the corresponding graph \( G_i = (Q_i, E_i) \) and the matrix \( \text{A} \) are shown in Figure 1.

There are incoming edge \( e_{ji}^\text{in} \) and outgoing edge \( e_{ji}^\text{out} \) for each node. The weight of incoming edge \( e_{ji}^\text{in} \) is represented by \( A_{ji}^\text{in} \), and the weight of outgoing edge \( e_{ji}^\text{out} \) is represented by \( A_{ji}^\text{out} \) [45].

\[
\begin{align*}
A_{ji}^\text{in} &= \frac{\text{count}(v_j, v_i)}{\sum_{k \in N_i(n_j)} \text{count}(v_k, v_i)} \\
A_{ji}^\text{out} &= \frac{\text{count}(v_i, v_j)}{\sum_{k \in N_{ji}(n_i)} \text{count}(v_i, v_k)}
\end{align*}
\]

Function \( \text{count}(x, y) \) means whether there is an edge between node \( x \) and \( y \). \( N_i(n_j) \) is the set of predecessor nodes \( v_i \). \( N_{ji}(n_i) \) is the set of successor node \( v_i \). The neighbor connection matrix element can be written as:

\[
\begin{align*}
\text{A}_{\text{in}}[i, j] &= \text{A}_{ji}^\text{in} \\
\text{A}_{\text{out}}[i, j] &= \text{A}_{ji}^\text{out}
\end{align*}
\]

**IV. OUR APPROACH**

In this section, we introduce how we build the proposed ATST-GGNN. The architecture of the ATST-GGNN is shown in Figure 2. Firstly, a user’s check-in sequence is converted to a sequence graph confusing spatiotemporal information. Secondly, three kinds of contextual informations are integrated into the ST-GGNN to realize the dynamic update of node vectors. Thirdly, the short-term representation of user is obtained by window-based pooling method. We utilize attention network to learn the long-term representation of user. Finally, the loss function is constructed by cross entropy, and the model parameters are optimized.

**A. SPATIOTEMPORAL CHECK-IN SEQUENCE GRAPH**

Considering the effect of spatiotemporal contextual information on human real-world check-in activities, modeling the geographical and temporal influence to dynamically update the nodes in the sequence graph is essential to predict the next location. To model such information more effectively in the sequence graph, a model called spatiotemporal GGNN (ST-GGNN) is proposed. Firstly, we employ the distance interval \( \Delta t_{x,y} \) and the time interval \( \Delta d_{x,y} \) to define the spatial weight and temporal weight between nodes in sequence graph, respectively. The graph structure of user’s check-in sequence is converted to the neighbor connection matrix, time connection matrix and space connection matrix. Secondly, we dynamically update node vectors by three kinds of contextual information, such as neighbor, time and space in ST-GGNN model. Finally, POI embedding representations of the graph nodes are obtained by iterative operation of ST-GGNN.

In sequence graph, the calculation method of incoming weight and outgoing weight based on time interval and distance interval is defined as \( T_{-A_{ji}^\text{in}}, T_{-A_{ji}^\text{out}}, D_{-A_{ji}^\text{in}} \) and \( D_{-A_{ji}^\text{out}} \) respectively.

\[
T_{-A_{ji}^\text{in}} = \frac{T_{-\text{weight}}(v_j, v_i)}{\sum_{k \in N_i(n_j)} T_{-\text{weight}}(v_k, v_i)} \\
T_{-A_{ji}^\text{out}} = \frac{T_{-\text{weight}}(v_i, v_j)}{\sum_{k \in N_{ji}(n_i)} T_{-\text{weight}}(v_i, v_k)}
\]
The haversine formula of the coordinates of longitude and latitude information can be converted into two kinds of connection matrices. The spatial connection matrix $A_T$ represents the spatial connection information, and the time connection information, respectively. The $T \_ A_{ij}^{in}$ and $T \_ A_{ij}^{out}$ extract the $i$th row of time connection matrix $A_T$. The $D \_ A_{ij}^{in}$ and $D \_ A_{ij}^{out}$ extract the $i$th row of spatial connection matrix $A_D$. The $W_T$, $W_T^{out}$, $W_D$, $W_D^{out}$, $W_{dis}$, and $W_{dis}^{out}$ are parameter matrices.

Finally, all node vectors of sequence graph are updated dynamically by ST-GGNN:

$$ r^1_i = \sigma (W_{ra}a^u_i + \gamma W_{dis}a^u_i + \gamma W_{dis}a^u_i + (1 - \gamma)W_{time}a^u_i + W_{qa}q_i^{l-1} + b_r) $$

$$ z^1_i = \sigma (W_{za}a^u_i + \gamma W_{dis}a^u_i + (1 - \gamma)W_{time}a^u_i + W_{qa}q_i^{l-1} + b_z) $$

$$ \hat{q}^1_i = tanh(W_{hr}r^1_i + q_i^{l-1}) + W_{qa}a^u_i + \gamma W_{dis}a^u_i + (1 - \gamma)W_{time}q_i^{l-1} + b_q) $$

Where Formula (18), (19) and (20) is an improvement on formula (2), (3) and (4). The $W_{dis}$, $W_{time}$, $W_{dis}$, $W_{time}$, $W_{dis}$, and $W_{time}$ are parameter matrices. Hyperparameters $\gamma$ is used to adjust the effect of time and distance on the recommended next POI.

**C. EMBEDDING REPRESENTATION OF SEQUENCE**

The check-in behaviors of user are periodic and regular, so the contribution of each location in the check-in sequence is different for next POI recommendation. The irrelevant check-ins can generate noise. Thus, attention mechanism is used in this paper, an attention-based spatiotemporal GGN model is proposed.

The POI embedding of check-in sequence is obtained by ST-GGNN, that is $h_{\text{hidden}} \in \mathbb{R}^{n \times d}$. In order to obtain the user’s short term representation $s_t$, a kind of window-based pooling method is proposed.

$$ s_t = \frac{1}{w} \sum_{j=n-w+1}^{n} h_{\text{hidden}} $$

$h_{\text{hidden}}$ is the embedded representation of $i$th node being closest to the next POI. $w$ is the size of window. The long
term representation $s_g$ of the user is obtained by the attention mechanism. The weight coefficient $\alpha_i$ is designed as follow.

$$\alpha_i = \text{hidden}_i \times W^\alpha + s_i \times W^l + b$$  \hspace{1cm} (22)

$W^\alpha, W^l \in \mathbb{R}^{k \times l}$ are the weight matrices of the embedded vectors. The long term representation $s_g$ is shown in following formula.

$$s_g = \sum_{i=1}^{n} \alpha_i \times \text{hidden}_i$$  \hspace{1cm} (23)

$\alpha_i$ is the corresponding weight coefficient for sequence graph node $q_i$. Finally, the long and short term are combined, and the embedded representation of sequence graph is obtained by linear transformation, that is $s_h$.

$$s_h = [s_g; s_l] \times W^h$$  \hspace{1cm} (24)

\section*{D. MODEL TRAINING}

The score of each candidate check-in can be calculated by embedding representation $s_h$ and check-in feature matrix $L$. Then the probability distribution of each candidate check-in is calculated by softmax function, which can be written as follows:

$$\hat{y} = \text{softmax} (s_h \times L^T)$$  \hspace{1cm} (25)

$y$ is supposed to the true value of next location in POI sequence. The loss function is defined as the cross-entropy of the prediction $\hat{y}$ and the ground truth $y$.

$$\text{loss} = -\sum_{i=1}^{N} y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)$$  \hspace{1cm} (26)

All parameter are updated by stochastic gradient descent method to minimize the objective function until the convergence of the objective function is obtained. An attention-based spatiotemporal GGNN POI recommendation algorithm steps are as follows:

\begin{algorithm}
\caption{Training of ATST-GGNN}
\begin{algorithmic}
\State Input: set of check-in sequences $C$, time scaling factor $tt$, distance scaling factor $dd$, weight coefficient $\gamma$.
\State Output: ATST-GGNN model parameters $\Theta$.
\State //construct training instances
\For {each user $u \in U$}
\For {each location $v_i \in S_u$}
\State $\Delta t_t = t^u_{t+1} - t^u_t$;
\State $\Delta d_d = Haversine(v^u_{t+1}, v^u_t)$;
\State $\text{train\_time}_i = f(\Delta t_t, tt)$;
\State $\text{train\_dist}_i = f(\Delta d_d, dd)$;
\State Add $(\text{train\_time}_i, \text{train\_dist}_i)$ to $D^u$;
\EndFor
\EndFor
\State //train the model
\State Initialize the parameter set $\Theta$;
\State While(exceed(maximum number of iterations)==False) do
\State Randomly get batch of training list $D_b$ from $D$;
\State $A, b, \text{Dis}_b, \text{Time}_b = \text{get\_slice}(D_b)$;
\State $\text{hidden}_b = \text{ST-GGNN}(A, \text{Dis}_b, \text{Time}_b, \text{hidden}_b, \gamma)$;
\State $\text{score}_b = \text{Attention}(\text{hidden}_b)$;
\State Minimizing the objective (26);
\State End while
\State Saving the learned parameter set $\Theta$.
\end{algorithmic}
\end{algorithm}

\section*{V. EXPERIMENTS}

In this section, we firstly present the experimental datasets, evaluation metrics and parameter settings. Then, we assess the proposed method with compared methods. Besides, we analyze the performance results with different component, different vector dimension, size of window($w$), as well as effect of time and distance($\gamma$). Finally, the influence of different layers of ATST-GGNN are analyzed for two datasets. In this section, we intend to answer the following questions through experiments.

\begin{itemize}
\item \textbf{RQ1:} How does our model ATST-GGNN perform compared with other state-of-the-art models ?
\item \textbf{RQ2:} How effective are our designed ST-GGNN and proposed attention module?
\item \textbf{RQ3:} What are the influence of different parameter settings (vector dimension, $w$, and $\gamma$) for our model ATST-GGNN ?
\item \textbf{RQ4:} What are the effect of different layers in ATST-GGNN from different datasets ?
\end{itemize}

\section*{A. DATASETS}

This paper chooses two kinds of open check-in datasets for algorithm testing, such as Foursquare and Gowalla. The two datasets are from two location-based social network sites. Both sites all offer check-in services that allow users to share information about their current location, which contains user information, check-in information, location information and time information, and so on [33]. Firstly, two kinds of datasets are preprocessed. Following (Zhang et al.2019), the users whose check-in number is less than 10 will be deleted. The accessed check-ins being less than 10 will be deleted. The 70% of two datasets are used as the training set. The remaining 20% are used as the test set. The remaining 10% are used as the validation set to tune parameters. Statistical characteristics of the two datasets are shown in Table 2.

\begin{table}[h]
\centering
\caption{Statistical characteristics of datasets}
\begin{tabular}{|c|c|c|}
\hline
\textbf{Datasets} & \textbf{Foursquare} & \textbf{Gowalla} \\
\hline
Users & 2009 & 7747 \\
Locations & 4326 & 12948 \\
Check-ins & 75162 & 280263 \\
Density & 8.65E-3 & 2.79E-3 \\
\hline
\end{tabular}
\end{table}

\section*{B. EVALUATION METRICS}

The Precision@K and MRR@K (mean reciprocal ranking) will be used as evaluation indicators. The $K$ represents the number of POI recommendation. The Precision represents the proportion of correctly recommended locations amongst the top-K locations. The MRR is the average of reciprocal ranks of the correctly recommended items. The MRR measure considers the order of recommendation ranking, where larger MRR indicates that correct recommendations items are in the top of the ranking list.
C. PARAMETER SETTING

The experimental platform is composed of Intel i5 CPU, 8G RAM, Windows 7 operating system, etc. The Pycharm is used as the development tool. Python 3.5 is used as the program language. Pytorch is used as the neural network learning framework. In experiment, hyperparameter learning rate lr is 0.001, L2 penalty λ is 0.003, the size of batch batch_size is 20, and the number of training epoch_num is 32. The distance scale factor is 200. The time scale factor is 30. The recommended length K is 5, 10, and 20, respectively.

D. PERFORMANCE COMPARISON (RQ1)

First, for question RQ1, to demonstrate the overall performance of the proposed model, we compare it with 7 kinds of state-of-the-art POI recommendation methods.

- BPR-MF [46] optimizes the matrix factorization model on implicit feedback data using a pairwise ranking loss.
- FPMC-LR [29] is a POI recommendation method based on three-order tensor decomposition.
- LSTM [9] applies recurrent neural network to learn users’ sequential behaviors based on the check-in location sequences.
- ST-RNN [31] combines spatial and temporal information based on RNN for POI recommendation.
- PLSPL [36] consider the spatial and temporal features of POIs in user history records and leverage attention mechanism to capture user preference for POI recommendation.
- SR-GNN [11] is a session recommendation method based on graph neural network. It is applied to the POI recommendation.
- GC-SAN [39] utilizes both graph neural network and self-attention mechanism for session-based recommendation.

The overall performance in terms of Precision@K and MRR@K is shown in Table 3.

For both the two datasets, BPR-MF and FPMC-LR are the worst performer. Because BPR-MF and FPMC-LR don’t analyze the check-ins’ sequence features. LSTM algorithm is superior to BPR-MF and FPMC-LR in Precision and MRR. Because it analyzes the sequence characteristics of check-ins by long-short term memory network.

Compared with the LSTM, P@5, P@10, P@20, MRR@5, MRR@10 and MRR@20 of ST-RNN are increased, on average, by 18.42, 15.35, 10.03, 21.59, 15.59, and 19.89 percent, respectively. The performance improvement is due to the advantage of confusing spatial and temporal information.

For the Foursquare, compared with the ST-RNN, P@5, P@10, P@20, MRR@5, MRR@10 and MRR@20 of PLSPL are increased, on average, by 27.59, 34.15, 22.67, 30, 31.82, and 28.57 percent, respectively. For the Gowalla, compared with the ST-RNN, P@5, P@10, P@20, MRR@5, MRR@10 and MRR@20 of PLSPL are increased, on average, by 41.38, 39.39, 28.89, 38.46, 33.33, 38.89 percent, respectively. The experimental results show that the attention mechanism can improve the performance of POI recommendation. Besides, the performance of PLSPL can be improved better when the Gowalla is sparser.

Compared with the PLSPL, SR-GNN and GC-SAN recommendation algorithms have greatly improved in Precision and MRR. It shows that the gated graph neural network has obvious advantages in improving the performance of POI recommendation.

Finally, our method ATST-GGNN always achieves the best performance regardless of datasets and evaluation metrics, and gains 11.39% Precision and 13.79% MRR improvements on average against the strongest baseline GC-SAN. In ATST-GGNN, we use spatiotemporal context information to dynamically update nodes in the sequence graph, and obtain the complex transfer relationships between the check-ins, and improve the local embedding representation of graph nodes by window pooling method, as well as the global embedding representation of graph nodes by integrating it into attention mechanism. The results mentioned above indicate that ATST-GGNN can indeed enhance the performance of next POI recommendation.

E. ABLATION STUDY (RQ2)

Next, we turn to RQ2. To verify the importance of designed ST-GGNN and proposed attention module, the different components of model will be compared. GGNN indicates that spatiotemporal information and attention module are not used, and only the GGNN model is used. ST-GGNN indicates that attention module is not used, and only the spatiotemporal information and GGNN model are used. The experimental results are observed from the Foursquare.

Table 4. Experimental results of ablation model

| Dataset | P@5 | P@10 | P@20 | MRR@5 | MRR@10 | MRR@20 |
|---------|-----|------|------|-------|--------|--------|
| GGNN    | 0.036 | 0.062 | 0.115 | 0.035 | 0.038 | 0.041 |
| ST-GGNN | 0.130 | 0.212 | 0.236 | 0.122 | 0.133 | 0.142 |
| ATST-GGNN | 0.134 | 0.237 | 0.262 | 0.134 | 0.144 | 0.154 |

We have the following conclusions from Table 4. GGNN model performs the worst performance in terms of Recall and MRR, compared to ST-GGNN, and ATST-GGNN. ATST-GGNN model improves the Precision and MRR by 3%-11% and 8%-10% compared with ST-GGNN model, respectively. It shows that the effectiveness of spatiotemporal information and attention module is very important for POI recommendation.

F. PARAMETER STUDY (RQ3)

To answer RQ3, different vector dimension, hyperparameter ω and γ, will have different effects on the recommended results for the POI recommendation based on ATST-GGNN model.
| Datasets | Foursquare | Gowalla |
|----------|------------|---------|
|          | Precision | MRR     | Precision | MRR     |
|          | @5        | @10     | @20       | @5        | @10     | @20       |
| BPR-MF   | 0.014     | 0.017   | 0.026     | 0.007     | 0.008   | 0.011     |
| FPMC-LR  | 0.019     | 0.031   | 0.033     | 0.012     | 0.015   | 0.017     |
| LSTM     | 0.024     | 0.033   | 0.068     | 0.016     | 0.019   | 0.022     |
| ST-RNN   | 0.029     | 0.041   | 0.075     | 0.020     | 0.022   | 0.028     |
| PLSPL    | 0.037     | 0.055   | 0.092     | 0.026     | 0.029   | 0.036     |
| SR-GNN   | 0.125     | 0.18    | 0.225     | 0.055     | 0.062   | 0.055     |
| GC-SAN   | 0.145     | 0.2     | 0.245     | 0.075     | 0.082   | 0.095     |

ATST - GGNN: 0.167, 0.227, 0.261, 0.092, 0.089, 0.112, 0.065, 0.082, 0.1, 0.041, 0.045, 0.043

Improv. 15.17%, 13.50%, 6.53%, 22.67%, 8.54%, 17.89%, 10.17%, 9.33%, 13.64%, 10.81%, 15.38%, 7.50%

This paper compares the effects of three parameters on both Precision@k and MRR@k.

1) Analysis with Vector Dimension

The vector dimensions are very significant in the model, and have an important impact on the recommended results. The effects on Precision and MRR are compared by setting different vector dimensions, such as 50, 100, 150, 200, and 250 from the Foursquare datasets.

![FIGURE 3. Experiment of vector dimensions from Foursquare](image)

Fig.3 shows the experiment results of different vector dimensions. When $50 \leq \text{hiddden size} < 150$, Precision and MRR gradually increased, as the vector dimension increases. When $150 < \text{hiddden size} \leq 250$, Precision and MRR gradually reduced. When $\text{hiddden size} = 150$, Precision and MRR become the maximum. The result means that the increase of $\text{hiddden size}$ does not necessarily lead to better results in model performance.

2) Analysis with $w$

In this subsection, we investigate the impact of the hyperparameter $w$. It is set to 1, 2, 3, 4, 5, 6, 7, 8 and 9, respectively.

![FIGURE 4. Comparison of $w$](image)

As shown in Fig.4, we can observe that the performance of our model is firstly improved, then decreases with a larger window size. When $w = 5$ on Foursquare or $w = 7$ on Gowalla, MRR becomes the maximum. This indicates that window-based pooling method is helpful to improve the
accuracy of recommendations. For training efficiency, we choose $w = 5$ on Foursquare or $w = 7$ on Gowalla for our task.

3) Analysis with $\gamma$

Hyperparameter $\gamma$ reflects the effect of time and distance on POI recommendation in the experiment. The experimental results are observed by adjusting the $\gamma$ value from the Foursquare datasets. The Hyperparameter $\gamma$ is set to 0.2, 0.4, 0.6 and 0.8, respectively.

![Figure 5](image1.png)

**FIGURE 5.** Comparsion of $\gamma$ from Foursquare

![Figure 6](image2.png)

**FIGURE 6.** Comparsion of layer

Fig. 5 shows that when the hyperparameter $\gamma$ is different, the effect of distance factor and time factor is different in POI recommendation. When $0.2 \leq \gamma \leq 0.4$, that is, the weight of time factor is larger than that of distance factor, the performance of POI recommendation can be improved when increasing the $\gamma$. When $0.4 < \gamma \leq 0.8$, that is, the weight of time factor is less than that of distance factor, the performance of POI recommendation decreases with the increase of $\gamma$. Experimental results show that time factors can better determine the next POI that the user will access. For example, if users go to the restaurant at noon, go to the coffee shop in the afternoon, and probably go to the bar in the evening in their life circle.

G. MODEL LAYER STUDY (RQ4)

The layers of ATST-GGNN reflects the order of each node obtaining the adjacent relationship in the check-in sequence graph. To answer RQ4, we will compares the effects of different layers from the Foursquare and Gowalla.

![Figure 6](image3.png)

**FIGURE 6.** Comparsion of layer

![Figure 6](image4.png)

**FIGURE 6.** Comparsion of layer

Fig. 6 shows that in the Foursquare, when $layer = 1$, the Precision and MRR of the POI recommendation become the largest. When $layer > 1$, the Precision and MRR decrease gradually. It indicates that the first order neighbor feature of nodes extracted can obtain better performance. In the Gowalla, when $1 < layer \leq 4$, the Precision and MRR of POI recommendation increase gradually. When $layer = 4$, the Precision and MRR of POI recommendation become maximum. When $4 < layer \leq 6$, The Precision and MRR of POI recommendation decrease gradually. It shows that the fourth order neighbor feature of nodes extracted can obtain better performance. The experimental results show that the lower density dataset needs extract higher order neighbor feature of nodes.

VI. CONCLUSIONS

In this paper, we have proposed an attention-based spatiotemporal GGNN model for next POI recommendation. Firstly, the user’s check-in sequence graph based on GGNN combines the spatiotemporal contextual information. Secondly, the local embedding representation of graph nodes by window pooling method is improved, as well as the global embedding representation of graph nodes by integrating it into attention mechanism. Finally, we test the algorithm in two open datasets. The experimental results show that Precision and MRR of ATST-GGNN are greatly improved compared with other recommended methods. For future work, we will further improve performance of POI recommendation by incorporating social networks, and more effective attention mechanisms.
REFERENCES

[1] Z. Ding, X. Li, C. Jiang, and M. Zhou, “Objectives and State-of-the-art of Location-based Social Network Recommender Systems,” ACM Computing Surveys, vol. 51, no. 1, pp. 1-28, 2018, doi:10.1145/3154526.

[2] S. Liu, “User Modeling for Point-of-Interest Recommendations in Location-Based Social Networks: The State of the Art,” Mobile Information Systems, vol. 4, pp. 1-13, 2018, https://doi.org/10.1155/2018/8707461.

[3] X. Li, G. Cong, X.-L. Li, T.-A. N. Pham, and S. Krishnaswamy, “Rank–GeoFM: A Ranking based Geographical Factorization Method for Point of Interest Recommendation,” in Proc. 38th ACM SIGIR Conf. Information Retrieval, 2015, pp. 433-442, [Online]. Available: https://doi.org/10.1145/2766462.2767722.

[4] W. Zhao, N. Zhou, A. Sun, J.-R. Wen, J. Han, and E. Y. Chang, “A Time-Aware Trajectory Embedding Model for Next-Location Recommendation,” Knowledge and Information Systems, vol. 56, no. 3, pp. 559-579, 2018, doi:10.1007/s10478-017-1107-4.

[5] C. Cheng, H. Yang, I. King, and M. R. Lyu, “Fused matrix factorization with geographical and social influence in location-based social networks,” in Proc. 26th AAAI Conf. Artif. Intell., 2012, pp. 27-31, [Online]. Available: https://www.aaai.org/ocs/index.php/AAAI/AAAI12/paper/view/4748.

[6] R. Salakhutdinov and A. Mnih, “Probabilistic Matrix Factorization,” in Proc. 20th Int. Conf. Neural Information Processing, 2007, pp. 1257-1264, [Online]. Available: https://doi.org/10.1155/S0258126915812624.

[7] Y. Koren, R. Bell, and C. Volinsky, “Matrix Factorization Techniques for Recommender Systems,” Computer, vol. 42, no. 8, pp. 30-37, 2009, doi:10.1109/MC.2009.218.

[8] M. Quadrana, A. Karatzoglou, B. Hidasi, and P. Cremonesi, “Evaluation Methods for Point-of-Interest Recommendations,” ACM Transactions on Intelligent Systems and Technology, vol. 3, no. 2, pp. 1-41, 2012, doi:10.1145/2247649.2247650.

[9] F. Yu, Q. Liu, S. Wu, L. Wang, and T. Tan, “A Deep Learning Model for Next-Location Recommendation,” in Proc. 39th Int. ACM SIGIR Conf. Research and Development in Information Retrieval, 2016, pp. 729-732, [Online]. Available: https://doi.org/10.1145/2911451.2914683.

[10] M. Chen and X. Zhou, “DeepRank: Learning to rank with neural networks for recommendation,” Knowledge-Based Systems, vol. 209, pp. 106478, 2020, doi:10.1016/j.knosys.2020.106478.

[11] C. Yang, L. Bai, C. Zhang, Q. Yuan, and J. Han, “Collaborative Filtering and Semi-Supervised Learning: A Neural Approach for POI Recommendation,” in Proc. 23rd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining, 2017, pp. 13-17, [Online]. Available: https://doi.org/10.1145/3097893.3098094.

[12] X. Chen, F. Liu, X. Zhao, and T. Li, “Points-of-interest Recommendation Based on Convolution Matrix Factorization,” Applied Intell., vol. 48, pp. 2458-2469, 2017, doi:10.1007/s10489-017-1103-0.

[13] C. Ma, Y. Zhang, Q. Wang, and X. Liu, “Point-of-Interest Recommendation: Exploiting Self-Attentive Autoencoders with Neighbor-Aware Influence,” in Proc. 27th ACM Int. Conf. Information and Knowledge Management, 2018, pp. 697-706, [Online]. Available: https://doi.org/10.1145/3269206.3271733.

[14] C. Cheng, H. Yang, M. R. Lyu, and I. King, “Where You Like to Go Next: A Spatial-temporal LSTM Model for Next POI Recommendation,” in Proc. 15th Int. Conf. Artif. Intell. Inf. Sys., 2013, pp. 2605-2611, [Online]. Available: https://doi.org/10.1145/2490649.2490665.

[15] S. Feng, X. Li, Y. Zeng, G. Cong, Y. M. Chee, and Q. Yuan, “Personalized Ranking Metric Embedding for Next-Location POI Recommendation,” in Proc. 15th Int. Conf. Artif. Intell. Inf. Sys., 2015, pp. 2069-2075, [Online]. Available: https://doi.org/10.1145/2832415.2832536.

[16] Q. Liu, S. Wu, L. Wang, and T. Tan, “Predicting the Next Location: A Recurrent Model with Spatial and Temporal Contexts,” in Proc. 30th AAAI Conf. Artif. Intell., 2016, pp. 194-200, [Online]. Available: https://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/11900.

[17] P. Zhao, H. Zhu, Y. Liu, Z. Li, J. Xu, and V. S. Sheng, “Where to Go Next: A Spatio-temporal LSTM Model for Next POI Recommendation,” arXiv, 2018, arXiv:1806.06671, [Online]. Available: http://export.arxiv.org/pdf/1806.06671.

[18] D. Lian, Y. Wu, Y. Ge, X. Xie, E. Cheng, and G. Cong, “Geography-Aware Spatial-Layered Location Recommendation,” in Proc. 26th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining, 2020, pp. 2009-2019, [Online]. Available: http://staff.ustc.edu.cn/~liandefu/paper/locpred.pdf.

[19] J. Hao, Y. Dun, G. Zhao, Y. Wu, and X. Qian, “Anutter-Graph Attention Model for Personalized Sequential Recommendation,” IEEE Transactions on Multimedia, 2021, doi:10.1109/TMM.2021.3097186.

[20] L. Huang, Y. Ma, S. Wang, and Y. Liu, “An Attention-based Spatio-temporal LSTM Network for Next POI Recommendation,” IEEE Transactions on Service Computing, vol. 12, pp. 1-13, 2019, doi:10.1109/TSC.2019.2918310.

[21] Y. Wu, K. Li, G. Zhao, and X. Qian, “Personalized Long- and Short-term Preference Learning for Next POI Recommendation,” IEEE Trans. Knowledge and Data Engineering, 2020, doi:10.1109/TKDE.2020.3002531.

[22] P. Perozzi, R. A. Rfoè, and S. Skiena, “Deepwalk: Online Learning of Social Representations,” in Proc. 26th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining, 2014, pp. 701-710, [Online]. Available: https://doi.org/10.1145/2623330.2623732.

[23] J. Tang, M. Qu, M. Wang, Z. Zhang, J. Yan, and Q. Mei, “Line: Large-scale network embedding,” in Proc. 24th Int. Conf. World Wide Web, 2015, pp. 1067-1077, [Online]. Available: https://doi.org/10.1145/2661823.2661855.

[24] C. Xu, P. Zhao, Y. Liu, and V. S. Sheng, J. Xu, F. Zhuang, J. Fang, and X. Zhou, “Graph Contextualized Self-Attention Network for Session-based Recommendation,” in Proc. 28th Int. Joint
QUAN LI was born in 1982. He received the M.S. degree in Computer Application from Nanchang Hangkong University, China, in 2009. He is working in Hubei Normal University. His research interests include recommender system, machine learning, and neural network.

XINHUA XU was born in 1968. He received the M.S. degree from Huazhong University of Science and Technology, China, in 2006. He is currently an Professor with Hubei Normal University, China. His research interests include database technology, and neural network.

XINGHONG LIU was born in 1969. She received the M.S. degree from Central China Normal University, China, in 2002. She is currently an Professor with Hubei Normal University, China. Her research interests include information technology, and neural network.

QI CHEN was born in 1980. She received the Ph.D. degree from Central China Normal University, China, in 2019. She is currently an Associate Professor with Hubei Normal University, China. Her research interests include pattern recognition, and neural network.