Intra-urban Region-based Traffic Flow Prediction Based on Spatial-Temporal Graph Convolutional Network Enhanced by Spatial Context

Guikai Xi
Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences
Shenzhen, China
University of Chinese Academy of Sciences
Beijing, China
gk.xi@siat.ac.cn

Ling Yin*
Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences
Shenzhen, China
yinling@siat.ac.cn

Kang Liu
Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences
Shenzhen, China
kang.liu@siat.ac.cn

ABSTRACT

Short-term traffic flow prediction in intra-urban regions plays a significant role in traffic management and public safety. However, it is still a challenging task due to the complex spatial and temporal dependencies in traffic data. Existing studies usually modeled the spatial and temporal dependencies separately, and neglected to integrate the spatial context in the temporal dependency modeling. In addition, most of the studies either ignored semantic relationships among regions or built multiple graphs to model different relationships. In this study, we propose a spatial-temporal graph convolutional network with temporal dependency modeling enhanced by spatial context (ST-GCN-SC) for short-term region-based traffic flow prediction in the urban environment. First, a spatial-context enhanced long short-term memory network (SC-LSTM) is presented to utilize both the sequential information and spatial context for the temporal dependency modeling. Second, a graph construction method is proposed to construct a relation graph to model both the adjacent relationships and the semantic relationships (i.e., flow pattern similarities) among regions, which helps ignore false signals from adjacent regions and model the long-range spatial correlations from a globally spatial perspective. Our experiments on two public datasets show that the proposed model achieves an RMSE reduction of 4.8% and 5.7% on the BikeNYC and TaxiBJ datasets, respectively, compared with several state-of-the-art methods, proving that the proposed model can facilitate short-term region-based traffic flow prediction.

*The corresponding author.

UrbComp '21, November 1, 2021, Beijing, China

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/10.1145/1122445.1122456

CCS CONCEPTS

• **Applied computing** → *Transportation*; • **Information systems** → *Spatial-temporal systems*; *Data mining*.

KEYWORDS

traffic prediction, spatial-temporal prediction, urban computing, deep learning, graph convolutional network

ACM Reference Format:
Guikai Xi, Ling Yin, and Kang Liu. 2021. Intra-urban Region-based Traffic Flow Prediction Based on Spatial-Temporal Graph Convolutional Network Enhanced by Spatial Context. In *The 10th International Workshop on Urban Computing (UrbComp '21), November 1, 2021, Beijing, China.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

Short-term traffic flow prediction tasks can be divided into two types: network-based and region-based predictions [30]. The first type usually aims to predict the traffic flow collected by the sensors of the road network [5]. By dividing a city into a grid map (each grid is defined as a region), the second type is to predict the traffic inflow and outflow in each region of the city [6, 31, 32]. The region-based traffic flow can be calculated from the mobile phone data, floating car data, shared bicycle data, and etc.

In this study, we only study the second prediction task: short-term traffic flow prediction in intra-urban regions. Knowing the future region-based traffic flow accurately, governments can take better control measures to avoid large crowds of people, travelers can make better routing plans, and enterprises can also dynamically allocate transportation resources (e.g., shared bikes and taxis). To this end, accurate region-based traffic flow prediction is essential. However, predicting traffic flow in intra-urban regions is primarily challenging owing to the complex spatial and temporal dependencies in traffic data. On the one hand, the traffic inflow and outflow of a region are correlated with its historical inflows and outflows (i.e., temporal dependency). On the other hand, the traffic inflow and outflow of a region are also correlated with those of its adjacent regions (i.e., spatial dependency). In addition, the temporal dependency may exist at multiple different time scales (e.g., hours, days, and week) and environmental factors (e.g., temperature and rainfall) can affect the traffic inflows and outflows of the regions.

Early studies utilized two representative methods: classical statistical methods (e.g., ARIMA and its variants) [13, 16, 19] and machine learning methods (e.g., neural networks and support vector machines (SVM)) [23–25, 28] for traffic prediction. However, most of these methods can merely utilize the historical features of individual regions and do not capture the complex spatial-temporal dependency, which limits the prediction performance.

Recently, deep learning has achieved remarkable success across several domains, including computer vision and natural language processing [11, 12]. This development had inspired many attempts to use deep learning for short-term region-based traffic flow prediction. Typically, the traffic inflow and outflow of the gird map (i.e., the partitioned city) can be treated as an image with multiple channels. By given a set of historical images as the input, a feasible schema is using deep learning methods to predict the future image (i.e., the future traffic flows of the city). Several models using convolutional neural networks (CNNs) have been proposed to model the spatial dependency, and thus they can learn features from the spatially adjacent regions for better prediction [31, 32]. However, these methods are based on the spatial modeling method and failed to model the temporal sequential dependency. Therefore, to model both the spatial and temporal sequential dependencies, several models (e.g., ST-3DNet) are proposed to improve the region-based traffic prediction performance [6].

Although these studies have achieved reasonable performance, there is still scope to improve the short-term region-based traffic flow prediction accuracy from the following two perspectives.

First, population movement is always across regions, and changes of the traffic flows in one region will inevitably lead to the changes in another. Therefore, the spatial context can carry essential information about traffic flow changes in the adjacent regions and help improve the spatial-temporal prediction. In recent spatial-temporal prediction studies, the spatial and temporal dependencies were usually separately and sequentially modeled [4, 26, 27]. However, we argue that while modeling the temporal dependency, the spatial context of each region is not well-utilized in these studies. Geng et al. achieved taxi demand prediction improvement by utilizing the global contextual information of the city to re-weight observations in different time slices while modeling the temporal relations [4]. However, we find that the locally spatial context of each region can help improve prediction, and it is better than the globally spatial context in the temporal modeling process for short-term region-based traffic flow prediction.

Second, traditional convolutions are usually utilized to model the spatial dependency for region-based traffic flow prediction, which cannot well model the non-Euclidean spatial relationships (e.g., nonadjacent relationships) among regions. In real world, the functionality of a location can affect human mobility. For example, people arrive at the office area in the morning and depart in the afternoon. In addition, commercial areas will attract more crowd on weekends. Therefore, locations with similar functionalities may have similar traffic flow patterns and the similarities between traffic flow patterns may exist persistently. If we model the spatial relationships of these semantic neighbors (i.e., regions with similar traffic flow patterns), the prediction model can have a long-range spatial perception capability to capture the traffic flow changes of the entire city by another dimension of the spatial dependencies, which can improve the prediction performance.

To address these challenges, we propose a **s**patial-**t**emporal **g**raph **c**onvolutional **n**etwork with temporal dependency modeling enhanced by **s**patial **c**ontext (ST-GCN-SC) to concurrently predict the traffic inflow and outflow in each region of a city. Our contributions are as follows:

- We propose a **s**patial-**c**ontext enhanced **l**ong **s**hort-**t**erm **m**emory network (SC-LSTM) to model the temporal dependency. In SC-LSTM, both the sequential information and spatial context of each region are utilized.
- We design a histogram-based similarity algorithm to measure the semantic similarities (i.e., flow pattern similarities) among regions and then model the semantic relationships. Based on this, we propose a graph construction method to construct a relation graph for depicting both the adjacent and semantic relationships among regions and utilize graph convolutional networks to model the spatial relationships.
- Comprehensive experiments are conducted on two public traffic flow datasets. It indicates that the proposed ST-GCN-SC outperforms several state-of-the-art methods, proving the effectiveness of our spatial-temporal dependency modeling strategy for short-term region-based traffic flow prediction.

The rest of this study is organized as follows. First, we review some related works in Section 2 and introduce some preliminaries in Section 3. Then, we detail the proposed model in Section 4. Extensive evaluation and comparisons are conducted in Section 5. Finally, this paper is concluded with a discussion of the contributions and future work in Section 6.

2 RELATED WORK

Deep-learning-based models have significantly improved traffic prediction performance. Thus, in this section, we mainly discuss the deep-learning-based methods for traffic prediction.

By treating the features across the regions in the city at every time slice as an image with multiple channels, Zhang et al. proposed a deep learning model called DeepST for spatial-temporal data prediction [32]. They added residual learning [7] and further proposed a new model named ST-Resnet for better traffic flow prediction [31]. However, although these studies can model the spatial dependency by taking historical city-level traffic flow images as inputs, they still failed to model the temporal sequential dependency.

To concurrently model both the spatial and temporal dependencies, one category of the studies is using different structures of convolutions. Guo et al. adopted 3D convolutions [22] to extract both the spatial and temporal features of traffic data and proposed a ST-3DNet for traffic prediction [6]. However, 3D convolutions are ineffective and consume substantial time. In addition, the long-term sequential dependency is still overlooked.

Another category is using CNNs and recurrent neural networks (RNNs) to model the spatial and temporal sequential dependencies, respectively. Yao et al. proposed a deep multi-view spatial-temporal network (DMVST-Net) by using a local CNN, an LSTM layer [9], and a graph embedding technique named LINE [20] to model the spatial, temporal, and semantic relationships, respectively, for taxi demand prediction [27]. By integrating a flow gating mechanism and a periodically shifted attention mechanism, a spatial-temporal dynamic network (STDN) was proposed to improve the traffic prediction performance [26]. However, none of these studies considered the semantic relationships (i.e., flow pattern similarities) among regions while modeling the spatial and temporal dependencies. These models can merely utilize local features from adjacent regions, and the potentially useful features from distant regions (e.g., distant regions with similar patterns) are hard to learn. Even in [27], the extracted graph embedding of the semantic graph is used as the external features and thus the semantic relationships are still ignored in the spatial-temporal modeling process.

Non-Euclidean data also exists in traffic prediction, especially in the network-based predictions. Several studies have extended the network structure from traditional convolutions and RNNs to graph convolutions and RNNs by merely updating the convolutions (e.g., DCRNN [14] and STGCN [29]). To consider multiple region-wise relationships, Geng et al. proposed a spatiotemporal multi-graph convolution network (ST-MGCN) for region-based taxi demand prediction [4]. They collected additional data (i.e., POI and road network data) to construct multiple graphs and proposed a multi-graph convolution method to model these spatial relationships. However, multi-graph convolutions increase the convolution computation several times, and thus increase the model complexity. Lu et al. built two relation graphs to consider both the spatial neighbors and semantic neighbors, and utilized two proposed modules to model both the spatial relationships for road-based traffic speed prediction, which also increases the model complexity [15]. In contrast to ST-MGCN, the proposed graph construction method builds a relation graph to describe both the adjacent and semantic relationships among different regions. This neither changes the model complexity and nor requires any additional data. In addition, we propose a different strategy to leverage the spatial context while modeling the temporal dependency, which is proved to be more effective than ST-MGCN in the following experiments.

In summary, the difference between the proposed model and the related works is that we propose a graph construction method to model both the adjacent and semantic relationships based on a relation graph. To well utilize the spatial context while modeling the temporal dependency, compared with ST-MGCN [4], we propose a different temporal dependency modeling method named SC-LSTM by concatenating the traffic flow and extracted spatial context of each region as the input of the LSTM unit.

3 PRELIMINARIES

In this section, we first introduce several basic concepts, and then we define the problem that we aim to address.

3.1 Basic Concepts

Definition 3.1 (Region). A city can be divided into an $I{\times}J$ grid map based on longitudes and latitudes. Then, we define every grid of the city as a region.

Definition 3.2 (Region-based Traffic Flow Matrix). For a given time slice t, we can obtain a traffic flow matrix $X_t \in \mathbb{R}^{I\times J\times p}$ to describe the inflow and outflow of the entire city, where $I{\times}J$ is the size of the grid map and p is the number of features. As shown in Fig. 1, p is equal to 2 because only the inflow and outflow are considered in this study.

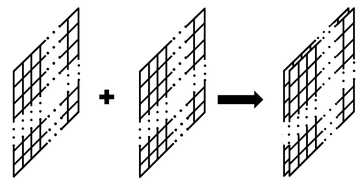


Figure 1: Construction of the traffic flow matrix.

Definition 3.3 (Flow Sequences at Different Time Scales). As the influence from historical flows exists across multiple time scales (e.g., hours, days, and weeks) [31], we construct the traffic flow matrix sequences of different time scales that are S_{t-1}^{hour}, S_{t-1}^{day}, and S_{t-1}^{week} as the model input. They are defined as follows.

$$S_{t-1}^{hour} : X_{t-l_h}, X_{t-l_h+1}, \cdots, X_{t-1},$$
$$S_{t-1}^{day} : X_{t-l_d*T_d}, X_{t-(l_d-1)*T_d}, \cdots, X_{t-T_d},$$
$$S_{t-1}^{week} : X_{t-l_w*T_w}, X_{t-(l_w-1)*T_w}, \cdots, X_{t-T_w},$$

where T_d and T_w are the lengths of the daily and weekly trend spans, respectively. And l_h, l_d, and l_w are the respective lengths of the sequences.

Definition 3.4 (Graph Convolution). Graph Laplacian is an essential operator for spectral graph analysis [1]. As a representation of a graph, the normalized Laplacian matrix L is denoted as $L = I_N - D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$, where I_N, A, and D are the identity, adjacency, and diagonal degree matrices, respectively, with $D_{ii} = \sum_j A_{ij}$. Because L is a real symmetric matrix, L can be further decomposed as $L = U \Lambda U^T$, where U is the matrix of eigenvectors and Λ is the diagonal matrix of eigenvalues of L. Graph convolutions [8] are convolution operations over graphs and can be defined as the multiplication of a signal x on a graph $\mathcal{G}$ with a filter g_θ, as follows:

$$g_\theta *_{\mathcal{G}} x = g_\theta(L)x = g_\theta(U\Lambda U^T)x = U g_\theta(\Lambda) U^T x,$$

where $*_{\mathcal{G}}$ denotes a graph convolution operation and $g_\theta(\Lambda)$ is a diaginal matrix.

However, multiplication with U is time-consuming as the time complexity is $O(n^2)$. Furthermore, computing the eigendecomposition of L is expensive, especially for large graphs. To localize the filters in space and reduce the time complexity, the filter g_θ can be approximated by Chebyshev polynomials $T_k(x)$ up to K-th order. Therefore, graph convolutions can be rewritten as follows [2]:

$$g_\theta *_{\mathcal{G}} x = \sum_{k=0}^{K} \theta_k T_k(\tilde{L}) x,$$

where $\tilde{L} = \frac{2}{\lambda_{max}} L - I_N$ and λ_{max} is the largest eigenvalue of L. The recursive definition of Chebyshev polynomials is $T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x)$ with $T_0(x) = 1$ and $T_1(x) = x$.

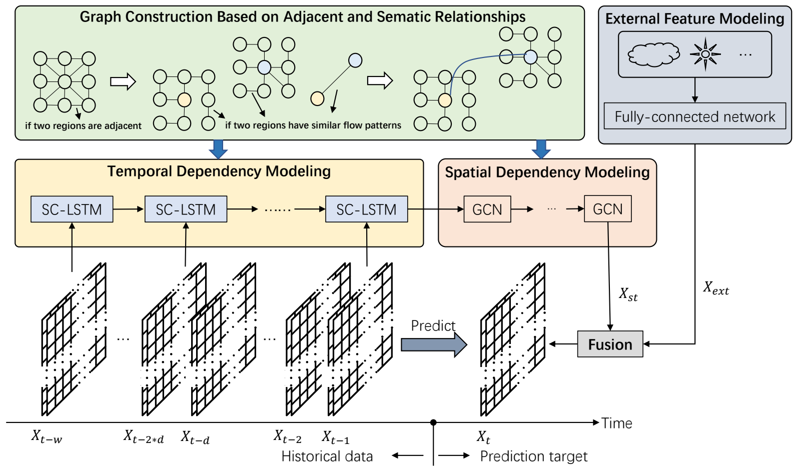


Figure 2: Model framework of the proposed model.

In this study, we define a graph convolutional layer as follows:

$$X_{l+1} = GCN(X_l) = g_\theta *_{\mathcal{G}} X_l,$$

where X_l denotes the input features of the l-th layer.

3.2 Problem Definition

Given a set of historical traffic flow matrices of the city, the aim of intra-urban region-based traffic flow prediction is to predict the future inflow and outflow in each region of the city. The formulation can be denoted as follows.

$$S_{t-1}^{week}, S_{t-1}^{day}, S_{t-1}^{hour} \xrightarrow{f(\cdot)} X_t,$$

where $f(\cdot)$ is the learned function.

4 METHODOLOGY

In this section, we describe the model architecture of the proposed ST-GCN-SC. As shown in Fig. 2, the proposed model comprises three parts: the temporal dependency modeling, spatial dependency modeling, and fusion.

4.1 Spatial Dependency Modeling

4.1.1 Graph Construction. To better model the complicated relationships (e.g., Euclidean and non-Euclidean relationships) among regions, we need to construct a relation graph for all the regions. Therefore, we propose a relation graph construction method by considering two types of relationships among regions: the adjacent and semantic relationships, which helps concurrently capture traffic flow changes from the adjacent neighbors and the semantic neighbors (i.e., regions with similar traffic flow patterns).

Adjacent Relationship. According to Tobler's First Law of Geography [21], everything is related to everything else, but near things are more related than distant things. It is natural to consider the adjacent relationships among regions for the spatial dependency modeling. Based on the adjacent relationship, a graph edge between two regions is added when they are adjacent to each other. For example, given a 3×3 grid map, the middle region has connecting edges only with its eight surrounding regions. Thus, we can get the responding adjacent matrix A_{adj} for these regions.

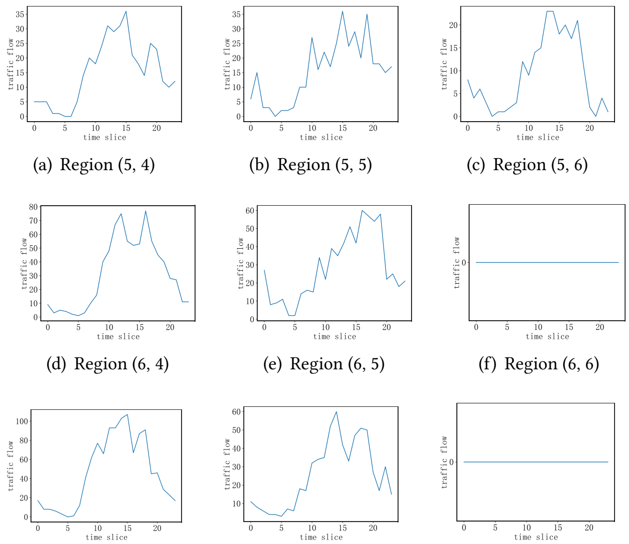
Semantic Relationship. However, the traffic flow patterns of the adjacent regions may be dissimilar. For example, we assume that a region i is with constant and zero traffic flow, and the adjacent region j is with changing and non-zero traffic flow. Although these two regions are adjacent, their flow patterns are entirely dissimilar. This situation has happened in the BikeNYC dataset, as shown in Fig. 3. Keeping the graph edges between the adjacent but dissimilar regions will transit false signals and may deteriorate performance. From another perspective, the distant regions (e.g., distant regions with similar flow patterns) may have close relations. Obviously, adding graph edges between these similar regions (i.e., regions with similar flow patterns), whether they are adjacent or not, can help predict the traffic flow trends from a globally spatial perspective. Therefore, designing a similarity algorithm to identify whether the regions are similar is necessary.

To identify the similar regions, a histogram-based similarity algorithm to compute the flow pattern similarity between two regions is designed. First, the histograms of traffic flow sequences are constructed, as shown in Algorithm 1. Subsequently, the cosine similarities of the traffic inflow and outflow histograms between two regions are computed and averaged, as shown in Algorithm 2. Finally, the flow pattern similarity (i.e., cosine similarity of histograms) between two regions is obtained. When the flow pattern similarity between two regions is much higher, we can think that the regions have very similar functionalities and strong correlations. Thus, a edge between these two regions can be added.

In essence, from the semantic perspective, the edge between two regions is defined as follows:

$$A_{se,\theta}^{i,j} = \begin{cases} 1, & w_{i,j} > \theta \\ 0, & otherwise \end{cases}, \tag{1}$$

where θ denotes the threshold. An edge between region i and region j is added when the flow pattern similarity value $w_{i,j}$ is higher than the threshold θ.



(a) Region (5, 4) (b) Region (5, 5) (c) Region (5, 6)

(d) Region (6, 4) (e) Region (6, 5) (f) Region (6, 6)

(g) Region (7, 4) (h) Region (7, 5) (i) Region (7, 6)

Figure 3: The traffic inflow sequences on 2014/9/20 in the BikeNYC dataset. Region (i, j) means the region in i-th row and j-th column of the grip map. And some regions have zero flows.

Algorithm 1 Histogram Construction Algorithm

Input: The given time series x and the expected number of bins N for the returned histogram.
Output: The generated histogram $hist$ of x.

function GET_HISTOGRAM(x, N)
  $max\_num = max(x)$
  $width = ceil(max\_num\ /\ N)$
  $hist = new\ list[N]$ // a list of length N
  for $i = 0 \to N - 1$ **do**
    $hist[i] = 0$
  end for
  if $max\_num \geq N$ **then**
    for i in x **do**
      $hist[i \mid width] = hist[i \mid width] + 1$
    end for
  end if
  return $hist$
end function

Combination of Two Types of Spatial Relationships. To model both the adjacent and semantic relationships, the graph adjacent matrix is finally defined as follows:

$$A_{comb} = A_{adj} \circ A_{se,\theta_1} + (1 - A_{adj}) \circ A_{se,\theta_2}, \tag{2}$$

where $\circ$ indicates the element-wise matrix multiplication. We use different thresholds (i.e., θ_1 and θ_2) for different parts of Equation 2 due to the inconsistent effects with respect to the distances between regions. When θ_1 is equal to θ_2, Equation 2 denotes a special case

Algorithm 2 Region Similarity Algorithm

Input: The inflow and outflow sequences of region i are S_i^{in} and S_i^{out}, respectively, at a specific day: $S_i^{in} = \{X_1^{i,in}, X_2^{i,in}, \cdots, X_{T_d}^{i,in}\}$, $S_i^{out} = \{X_1^{i,out}, X_2^{i,out}, \cdots, X_{T_d}^{i,out}\}$; the inflow and outflow sequences S_j^{in} and S_j^{out} of region j; the expected number of bins N for histograms.
Output: The similarity $w_{i,j}$ between regions i and j.

function CAL_SIMILARITY(S_i^{in}, S_i^{out}, S_j^{in}, S_j^{out})
  // N is the number of bins for histograms.
  $hist_i^{in}$ = GET_HISTOGRAM(S_i^{in}, N)
  $hist_i^{out}$ = GET_HISTOGRAM(S_i^{out}, N)
  $hist_j^{in}$ = GET_HISTOGRAM(S_j^{in}, N)
  $hist_j^{out}$ = GET_HISTOGRAM(S_j^{out}, N)
  $w_{i,j} = 0.5 * cosine\_similarity(hist_i^{in}, hist_j^{in}) + 0.5 * cosine\_similarity(hist_i^{out}, hist_j^{out})$
  return $w_{i,j}$
end function

of merely considering the semantic relationships among regions to construct the relation graph.

By additionally considering the semantic relationships, the proposed graph construction method has two main advantages. First, removing the attention on the adjacent but dissimilar regions can help eliminate the transmission of uncorrelated signals. Second, considering the semantic relationships among distant regions can help model the long-range spatial correlations and capture the traffic flow patterns effectively from a globally spatial perspective instead of merely from a locally spatial perspective.

4.1.2 Modeling Network. To learn the features from both the adjacent and semantic neighbors on graphs, graph convolutions are utilized. We design a spatial dependency modeling module that is denoted as follows.

$$\tilde{H}_{t-1} = g(H_{t-1}) \in \mathbb{R}^{|V| \times h}, \tag{3}$$

where H_{t-1} is the current feature representation of all regions, $|V|$ ($|V| = I \times J$) is the number of the regions in the city, and h is the dimension of the new representation. g is the spatial dependency modeling function, and it primarily consists of several graph convolutional layers. The rectified linear unit (ReLU) is applied to the outputs of the inner layers.

4.2 Temporal Dependency Modeling

Spatial context is essential in the spatial-temporal dependency modeling. To utilize the spatial context, as shown in Fig. 4, we propose a network named SC-LSTM by incorporating the original traffic flow and the spatial context for the temporal sequential dependency modeling. With the use of a sequential modeling strategy (e.g., temporal-then-spatial) to model the spatial and temporal dependencies, the spatiotemporal distribution may be changed when the former temporal dependency modeling is completed. Thus, the following spatial dependency modeling module is facing the changed spatial dependency that may be different from the correlations of input features among regions, which makes modeling challenging. The proposed SC-LSTM ensures spatial perception of each region while modeling the temporal dependency. Therefore, the aforementioned issues can be mitigated.

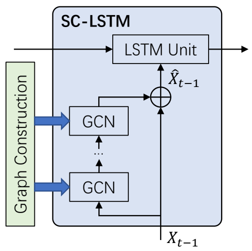


Figure 4: Overview of the SC-LSTM.

We can get a traffic flow matrix $X_{t-1} \in \mathbb{R}^{|V| \times p}$ at time slice $t-1$, where $|V|$ is the number of the regions. The number of the features p equals two since the inflow and outflow are considered.

First, two graph convolutional layers are designed to extract the spatial context for each region at each time slice as follows.

$$m_{t-1} = \delta(f_1(X_{t-1})) \in \mathbb{R}^{|V| \times k}, \tag{4}$$

$$\tilde{X}_{t-1} = f_2(m_{t-1}) \in \mathbb{R}^{|V| \times p}, \tag{5}$$

where both f_1 and f_2 are the graph convolutional layers and the activation function δ is the ReLU. The feature dimension of the inputs is increased to k and then decreased into p. Obviously, the extracted contextual information can carry essential information from their similar regions.

Secondly, the extracted spatial context is then concatenated with the original traffic flow matrix.

$$\hat{X}_{t-1} = [X_{t-1}, \tilde{X}_{t-1}] \in \mathbb{R}^{|V| \times 2p}, \tag{6}$$

where the dimension of the features is doubled.

Finally, as shown in Equation 7, a shared LSTM layer is utilized to model the temporal sequential dependency across all regions in the city. For each region, the LSTM layer aims to learn a feature representation of the input sequence that consists of the traffic flow features and the extracted context features.

$$H^i_{t-1} = LSTM(\hat{X}^i_{t-T}, \cdots, \hat{X}^i_{t-2}, \hat{X}^i_{t-1}), \tag{7}$$

where $i \in \{1, \cdots, |V|\}$ means the i-th region.

4.3 Fusion

4.3.1 Multi-time-scale Data Fusion. Traffic flow has periodic patterns at different time scales (e.g., hours, days, and weeks). Therefore, to enhance the prediction performance, we directly concatenate multiple traffic flow sequences of different time scales as the input of the proposed model. The new traffic flow sequence is composed of three parts: the hourly traffic flow sequence S^{hour}_{t-1}, daily traffic flow sequence S^{day}_{t-1}, and weekly traffic flow sequence S^{week}_{t-1}.

4.3.2 External Data Fusion. To consider the environmental factors, we follow the modeling method in [31]. For external features, one-hot encoding is applied to the category features (e.g., if the day is a weekend), and data normalization is performed to the continuous features (e.g., wind speed). Subsequently, different feature vectors are concentrated as an one-dimensional vector. A fully-connected network consisting two fully-connected layers is designed to predict the inflow and outflow for each region by only considering the one-dimensional vector as the input. Finally, the respective prediction results based on the historical traffic flow and external features are added together as the final result of the prediction model, as shown in Equation 8.

$$X_t = X_{st} + X_{ext}, \tag{8}$$

where X_{st} is the prediction result of the spatial-temporal model based on the historical traffic flow, and X_{ext} is the prediction result of the fully-connected network based on the external features. X_t is the final prediction result of the proposed model framework.

5 EXPERIMENTS

In this section, we first introduce two public datasets and the experimental settings. Then, we introduce the baseline methods and evaluation metrics. Finally, the experiments are conducted to prove the effectiveness of the proposed model.

5.1 Datesets

To evaluate the proposed model, we use two different public datasets [31]: BikeNYC and TaxiBJ, as described in Table 1. These two datasets are detailed as followed.

5.1.1 BikeNYC Dataset. This dataset is generated from the bike trajectory data in New York City from April 1st, 2014, to September 30th, 2014. In addition, it includes several external features such as the index of the day in a week, if this day is a weekday, and if this day is a holiday. The studied city is divided into 16×8 grids and the time interval is set as one hour. Thus, one day has 24 time intervals and totally we can get 4,392 available time intervals. Among this dataset, the last 10 days are chosen as the testing set and the rest are used for training.

5.1.2 TaxiBJ Dataset. This dataset is collected from the taxi trajectory data and meteorology data in Beijing City during 4 intervals: (1) July 1st, 2013 - October 30th, 2013, (2) March 1st, 2014 - June 30th, 2014, (3) March 1st, 2015 - June 30th, 2015, and (4) November 1st, 2015 - April 10th, 2016. The studied city is divided into 32×32 grids and the time interval is set to 0.5 hour. Therefore, we can get 22,459 available time intervals. In this dataset, the last 28 days are selected as the testing set and the rest are used for training.

5.2 Preprocessing and Experimental Settings

Min-max normalization is applied to scale the traffic flow data into the range [-1, 1]. For external features, we apply one-hot encoding to the categorical features and apply min-max normalization to scale the continuous external features into the range [0, 1].

In the following experiments, following the setting of ST-Resnet [31], the lengths of the hourly trend sequences, the daily trend sequences, and the weekly trend sequences (i.e., l_h, l_d, and l_w) are set as 3, 4, and 4 for the BikeNYC dataset, respectively. Similarly, for the TaxiBJ dataset, they are set as 3, 1, and 1, respectively.

In the proposed model, the number of LSTM layers is set to one and the number of LSTM units is set to 32. The number of GCN layers is set to two and the hidden units of all GCN layers are set to 32 for both the temporal and spatial dependency modeling. The

Table 1: Details of BikeNYC and TaxiBJ datasets.

Dataset	BikeNYC	TaxiBJ
Location	New York	Beijing
Date range	2014/4/1-2014/9/30	2013/7/1-2013/10/30 2014/3/1-2014/6/30 2015/3/1-2015/6/30 2015/11/1-2016/4/10
Time interval	1 hour	0.5 hour
Grid map size	16x8	32x32
# Time interval	4,392	22,459
# Holidays	20	41
Weather conditions	-	16 types
Temperature/$^{\circ}C$	-	-24.6~41.0
Wind speed/mph	-	0~48.6

graph convolution degree K equals one. For the graph construction, θ_1 and θ_2 are selected in the range [0.0,1.0] with an interval of 0.1.

We select 90% of the training set for training the models and the remaining 10% as the validation set to early-stop our training algorithm based on the best validation score. Then, we continue to train the models on the entire training set for a fixed number of epochs (e.g., 100). The learning rate and batch size are set to 0.0002 and 32, respectively. The proposed model is trained by Adam optimizer [10] and implemented using PyTorch [17].

To be consistent with [6, 31], we evaluate the baseline models and the proposed model by two popular regression metrics: root mean square error (RMSE) and mean absolute error (MAE).

5.3 Baselines

We compare the proposed model with the following methods for short-term region-based traffic flow prediction:

- **Historical Average (HA)**: We predict the inflow and outflow by averaging all historical inflow and outflow of the responding time intervals. For example, to predict the inflow between 09:00 and 09:30 on this Friday, we averaged the inflows between 09:00 and 09:30 of all historical Fridays.
- **LASSO**: LASSO takes historical data as the input features for linear regression with L1 regularization. In this study, we implement LASSO by using sklearn [18] and set the learning rate to 0.01.
- **GBDT** [3]: We use the gradient-boosting-decision-tree-based regression implemented by sklearn [18] for comparison. The default parameters are selected: the number of trees is 100, the maximum depth is 3, the learning rate is 0.1, etc.
- **ST-Resnet** [31]: ST-Resnet uses multiple CNN modules with residual connections to model temporal influences of three different time scales (i.e., hours, days, and weeks). In addition, it uses a fully connected network to model the influence from external features. For each CNN module, 4 and 12 residual units are used for the BikeNYC and TaxiBJ datasets, respectively.
- **ST-3DNet** [6]: ST-3DNet uses 3D convolutions to extract features from both the spatial and temporal dimensions. This model considers two temporal properties of traffic data: hourly and weekly properties.
- **STGCN** [29]: STGCN utilizes a fully convolutional network structure for traffic speed prediction. In this model, traffic flow information of past six time intervals are used.
- **ST-MGCN** [4]: ST-MGCN considers constructing multiple graphs based on different relationships between regions, namely neighborhood, functional similarity, and transportation connectivity. Then, the model uses (1) the proposed CGRNN to model the temporal relations and (2) multi-graph convolutions to model the spatial dependency. Without additionally data sources, we use the single-graph (i.e., the neighborhood graph) version of the model for comparison.

For ST-Resnet and ST-3DNet, the experimental results are averaged based on five runs of the official implementations[1,2] as the seeds are not set. For STGCN and ST-MGCN, the external features are modeled by using the same network architecture of ST-Resnet [31]. These two models are re-implemented by using Pytorch [17].

5.4 Experimental Results

5.4.1 Method Comparison. We evaluate the proposed model with several baseline models on the BikeNYC and TaxiBJ datasets. The results are shown in Table 2. The HA has the worst prediction metrics across all the baseline methods because it directly computes the average of the historical features as a result for each region. Although machine learning methods (e.g., LASSO and GBDT) merely consider the historical features of a region, they still perform better than the Historical Average. This is because machine learning methods can model the non-linear dependency from the given data.

Table 2: Performance comparison of different methods on the BikeNYC and TaxiBJ datasets.

Method	BikeNYC		TaxiBJ	
	RMSE	MAE	RMSE	MAE
HA	8.14	3.53	41.08	22.22
LASSO	7.83	3.51	21.65	12.54
GBDT	6.95	3.28	20.81	11.95
ST-Resnet	6.28	2.97	16.92	9.52
ST-3DNet	6.14	2.90	17.50	9.88
STGCN	6.01	3.01	18.79	10.41
ST-MGCN	6.18	3.23	18.24	10.77
ST-GCN-SC (using A_{adj})	5.77	**2.75**	16.30	9.21
ST-GCN-SC (using A_{comb})	**5.72**	2.77	**15.95**	**9.13**

By introducing deep learning to learn the spatial-temporal features, ST-Resnet experiences a significant improvement in evaluation metrics (i.e., RMSE and MAE) on both the dataset, further reducing RMSE from 6.95 to 6.28 and from 20.81 to 16.92 on the BikeNYC and TaxiBJ datasets, respectively. ST-3DNet yields lower evaluation metrics on the BikeNYC dataset, whereas it achieves worse prediction performance on the TaxiBJ dataset, compared with ST-Resnet. We find that this conclusion is different from that of

[1] https://github.com/lucktroy/DeepST.
[2] https://github.com/guoshnBJTU/ST3DNet.

the original study. STGCN that is designed for traffic speed prediction achieves worse prediction performance than ST-Resnet on the TaxiBJ dataset. This is understandable since the spatial-temporal properties of traffic speed data and region-based traffic flow data are not the same, and the modeling strategy for traffic speed prediction may not be suitable for region-based traffic flow prediction. It is the same for ST-MGCN that is designed for taxi demand prediction.

As described in Table 2, the proposed ST-GCN-SC performs well on both the BikeNYC and TaxiBJ datasets and significantly improves the evaluation metrics compared with the other state-of-the-art models. By additionally considering the semantic relationships (i.e., flow pattern similarities), the proposed ST-GCN-SC using A_{comb} further reduces the prediction error. On the BikeNYC dataset, the proposed model using A_{comb} improves RMSE; however, it yields a slightly worse MAE. This is because the proposed model seems to reduce errors by avoiding relatively large differences between the true and prediction values in the regions. On the TaxiBJ dataset, the proposed model reduces RMSE from 16.30 to 15.95 and MAE from 9.21 to 9.13. This proves that the semantic relationships among regions are significant, and modeling the semantic relationships helps improve performance.

Overall, compared with other baselines, the proposed model ST-GCN-SC with A_{comb} achieves an RMSE reduction of 4.8% and 5.7% on the BikeNYC and TaxiBJ datasets, respectively, validating that the proposed model is more efficient than the other state-of-the-art methods for short-term region-based traffic flow prediction.

5.4.2 Ablation Analysis. To prove the effectiveness of our spatial-context integration strategy, we evaluate SC-LSTM (compared with LSTM) for time series prediction and the ST-GCN for spatial-temporal prediction. Details of these variants are as followed.

- **LSTM**: This model only considers the temporal sequential dependency. A fully-connected layer is applied to the output of the last LSTM unit for time series prediction.
- **SC-LSTM**: This variant replaces the spatial dependency modeling module of ST-GCN-SC with a fully-connected layer and thus models the temporal sequential dependency only. Compared with LSTM, this variant additionally utilizes the spatial context.
- **ST-GCN**: Based on ST-GCN-SC, this variant removes the spatial context by replacing the SC-LSTM with the LSTM layer in the temporal dependency modeling module.

All these variants that utilize graph convolutions simply construct the relation graph structure based on the adjacent relationships.

As shown in Table 3, the LSTM model achieves RMSE of 6.76 and 19.85 on the BikeNYC and TaxiBJ datasets, respectively. By combining the LSTM and GCN models to model the temporal and spatial dependencies sequentially, ST-GCN outperforms the LSTM model, proving that the spatial information is significant for spatial-temporal prediction and the sequential modeling strategy is feasible to extract the spatial and temporal sequential features. SC-LSTM is proposed to utilize both the sequential information and the spatial context while modeling the temporal dependency. As shown in Table 3, the independently used SC-LSTM for time series prediction still outperforms the LSTM model, proving the significant role of the spatial context in the temporal dependency modeling. By

Table 3: Performance comparison among several variants of the proposed model on the BikeNYC and TaxiBJ datasets.

Method	BikeNYC		TaxiBJ	
	RMSE	MAE	RMSE	MAE
LSTM	6.76	3.22	19.85	11.71
ST-GCN	6.03	2.93	16.99	9.37
SC-LSTM	5.94	2.87	17.06	9.51
ST-GCN-SC	**5.77**	**2.75**	**16.30**	**9.21**

further learning the extracted temporal features (i.e., results of SC-LSTM) in the spatial dimension, the proposed ST-GCN-SC using A_{adj} achieves RMSE reductions on both the BikeNYC and TaxiBJ datasets when compared with SC-LSTM. This implies that the spatial dependency modeling is still indispensable for short-term region-based traffic flow prediction. In addition, the spatial context is proved to be effective again when we compare the prediction performance of ST-GCN-SC with that of ST-GCN.

6 CONCLUSION

In this study, we propose a deep learning model named ST-GCN-SC to learn the spatial and temporal sequential features for short-term region-based traffic flow prediction. In the proposed model, a sequential modeling strategy is utilized to model the spatial and temporal dependencies. To well-utilize the spatial context while modeling the temporal dependency, the SC-LSTM is proposed. A histogram-based similarity algorithm is designed to compute the flow pattern similarities among regions in order to identify the similar regions (i.e., model the semantic relationships). Therefore, a graph construction method is proposed to construct a relation graph for modeling both the adjacent and semantic relationships among regions, which does not increase the model complexity. The experiments with two public datasets show that the proposed model ST-GCN-SC outperforms other baseline methods with RMSE reductions of 4.8% and 5.7% on the BikeNYC and TaxiBJ datasets, respectively, validating the effectiveness of the proposed spatial-temporal prediction model.

Dynamic Time Warping (DTW) algorithm is commonly used to calculate the similarity between two time series. However, in this study, based on some experiments, we find that the DTW algorithm does not perform well when applied to the ST-GCN-SC. In the future, we will continue to evaluate other similarity algorithms for the graph construction and consider to automatically construct the relation graph. In addition, we plan to apply the proposed model to other domains (e.g., infectious disease prediction).

ACKNOWLEDGMENTS

This work is supported by the National Natural Science Foundation of China (Grant No.: 41771441 and 41901391), the Natural Science Foundation of Guangdong Province, China (Grant No. 2021A1515011191), the Major Science and Technology Projects in the Autonomous Region (Grant No. 020A03004-4), the Shenzhen Basic Research Program (Grant No. JCYJ20190807163001783), and Bill & Melinda Gates Foundation (Grant No. INV-005834).

REFERENCES

[1] Fan RK Chung and Fan Chung Graham. 1997. *Spectral graph theory*. Number 92. American Mathematical Soc.

[2] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. 2016. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in neural information processing systems*. 3844–3852.

[3] Jerome H Friedman. 2001. Greedy function approximation: a gradient boosting machine. *Annals of statistics* (2001), 1189–1232.

[4] Xu Geng, Yaguang Li, Leye Wang, Lingyu Zhang, Qiang Yang, Jieping Ye, and Yan Liu. 2019. Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 3656–3663.

[5] Shengnan Guo, Youfang Lin, Ning Feng, Chao Song, and Huaiyu Wan. 2019. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 922–929.

[6] Shengnan Guo, Youfang Lin, Shijie Li, Zhaoming Chen, and Huaiyu Wan. 2019. Deep spatial–temporal 3D convolutional neural networks for traffic data forecasting. *IEEE Transactions on Intelligent Transportation Systems* 20, 10 (2019), 3913–3926.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Identity mappings in deep residual networks. In *European conference on computer vision*. Springer, 630–645.

[8] Mikael Henaff, Joan Bruna, and Yann LeCun. 2015. Deep convolutional networks on graph-structured data. *arXiv preprint arXiv:1506.05163* (2015).

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

[10] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[11] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*. 1097–1105.

[12] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *nature* 521, 7553 (2015), 436–444.

[13] Xiaolong Li, Gang Pan, Zhaohui Wu, Guande Qi, Shijian Li, Daqing Zhang, Wangsheng Zhang, and Zonghui Wang. 2012. Prediction of urban human mobility using large-scale taxi traces and its applications. *Frontiers of Computer Science* 6, 1 (2012), 111–121.

[14] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. 2017. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. *arXiv preprint arXiv:1707.01926* (2017).

[15] Bin Lu, Xiaoying Gan, Haiming Jin, Luoyi Fu, and Haisong Zhang. 2020. Spatiotemporal Adaptive Gated Graph Convolution Network for Urban Traffic Flow Forecasting. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 1025–1034.

[16] Luis Moreira-Matias, Joao Gama, Michel Ferreira, Joao Mendes-Moreira, and Luis Damas. 2013. Predicting taxi–passenger demand using streaming data. *IEEE Transactions on Intelligent Transportation Systems* 14, 3 (2013), 1393–1402.

[17] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch. (2017).

[18] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.

[19] Shashank Shekhar and Billy M Williams. 2007. Adaptive seasonal time series models for forecasting short-term traffic flow. *Transportation Research Record* 2024, 1 (2007), 116–125.

[20] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. 2015. Line: Large-scale information network embedding. In *Proceedings of the 24th international conference on world wide web*. 1067–1077.

[21] Waldo R Tobler. 1970. A computer movie simulating urban growth in the Detroit region. *Economic geography* 46, sup1 (1970), 234–240.

[22] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. 2015. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*. 4489–4497.

[23] Tsung-Hsien Tsai, Chi-Kang Lee, and Chien-Hung Wei. 2009. Neural network based temporal feature models for short-term railway passenger demand forecasting. *Expert Systems with Applications* 36, 2 (2009), 3728–3736.

[24] Lelitha Vanajakshi and Laurence R Rilett. 2004. A comparison of the performance of artificial neural networks and support vector machines for the prediction of traffic speed. In *IEEE Intelligent Vehicles Symposium, 2004*. IEEE, 194–199.

[25] Senyan Yang, Jianping Wu, Yiman Du, Yingqi He, and Xu Chen. 2017. Ensemble learning for short-term traffic prediction based on gradient boosting machine. *Journal of Sensors* 2017 (2017).

[26] Huaxiu Yao, Xianfeng Tang, Hua Wei, Guanjie Zheng, and Zhenhui Li. 2019. Revisiting spatial-temporal similarity: A deep learning framework for traffic prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 5668–5675.

[27] Huaxiu Yao, Fei Wu, Jintao Ke, Xianfeng Tang, Yitian Jia, Siyu Lu, Pinghua Gong, Jieping Ye, and Zhenhui Li. 2018. Deep multi-view spatial-temporal network for taxi demand prediction. *arXiv preprint arXiv:1802.08714* (2018).

[28] Ramin Yasdi. 1999. Prediction of road traffic using a neural network approach. *Neural computing & applications* 8, 2 (1999), 135–142.

[29] Bing Yu, Haoteng Yin, and Zhanxing Zhu. 2017. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. *arXiv preprint arXiv:1709.04875* (2017).

[30] Haitao Yuan and Guoliang Li. 2021. A Survey of Traffic Prediction: from Spatio-Temporal Data to Intelligent Transportation. *Data Science and Engineering* 6, 1 (2021), 63–85.

[31] Junbo Zhang, Yu Zheng, and Dekang Qi. 2016. Deep spatio-temporal residual networks for citywide crowd flows prediction. *arXiv preprint arXiv:1610.00081* (2016).

[32] Junbo Zhang, Yu Zheng, Dekang Qi, Ruiyuan Li, and Xiuwen Yi. 2016. DNN-based prediction model for spatio-temporal data. In *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. 1–4.