Dynamic Spatial-Temporal Representation Learning for Traffic Flow Prediction

Lingbo Liu, *Graduate Student Member, IEEE*, Jiajie Zhen, Guanbin Li, *Member, IEEE*, Geng Zhan, Zhaocheng He, *Member, IEEE*, Bowen Du, *Member, IEEE*, and Liang Lin, *Senior Member, IEEE*

***Abstract*— As a crucial component in intelligent transportation systems, traffic flow prediction has recently attracted widespread research interest in the field of artificial intelligence (AI) with the increasing availability of massive traffic mobility data. Its key challenge lies in how to integrate diverse factors (such as temporal rules and spatial dependencies) to infer the evolution trend of traffic flow. To address this problem, we propose a unified neural network called Attentive Traffic Flow Machine (ATFM), which can effectively learn the spatial-temporal feature representations of traffic flow with an attention mechanism. In particular, our ATFM is composed of two progressive Convolutional Long Short-Term Memory (ConvLSTM [1]) units connected with a convolutional layer. Specifically, the first ConvLSTM unit takes normal traffic flow features as input and generates a hidden state at each time-step, which is further fed into the connected convolutional layer for spatial attention map inference. The second ConvLSTM unit aims at learning the dynamic spatial-temporal representations from the attentionally weighted traffic flow features. Further, we develop two deep learning frameworks based on ATFM to predict citywide short-term/long-term traffic flow by adaptively incorporating the sequential and periodic data as well as other external influences. Extensive experiments on two standard benchmarks well demonstrate the superiority of the proposed method for traffic flow prediction. Moreover, to verify the generalization of our method, we also apply the customized framework to forecast the passenger pickup/dropoff demands in traffic prediction and show its superior performance. Our code and data are available at https://github.com/liulingbo918/ATFM.**

***Index Terms*— Traffic flow prediction, mobility data, spatial-temporal modeling, attentional recurrent neural network.**

Manuscript received July 15, 2019; revised December 25, 2019 and April 15, 2020; accepted June 3, 2020. This work was supported in part by the National Natural Science Foundation of China under Grant U1811463, Grant 61976250, and Grant 61702565, in part by the Guangdong Basic and Applied Basic Research Foundation under Grant 2020B1515020048, in part by the National High Level Talents Special Support Plan (Ten Thousand Talents Program), and in part by the Fundamental Research Funds for the Central Universities under Grant 18lgpy63. The Associate Editor for this article was S. Hamdar. *(Corresponding author: Guanbin Li.)*

Lingbo Liu, Jiajie Zhen, Guanbin Li, and Liang Lin are with the School of Data and Computer Science, Sun Yat-sen University, Guangzhou 510000, China (e-mail: liulingb@mail2.sysu.edu.cn; zhenjj@mail2.sysu.edu.cn; liguanbin@mail.sysu.edu.cn; linliang@ieee.org).

Geng Zhan is with the School of Electrical and Information Engineering, The University of Sydney, Sydney, NSW 2000, Australia (e-mail: gengzh0308@gmail.com).

Zhaocheng He is with the School of Intelligent Systems Engineering, Sun Yat-sen University, Guangzhou 510000, China (e-mail: hezhch@mail.sysu.edu.cn).

Bowen Du is with the State Key Laboratory of Software Development Environment, Beihang University, Beijing 100191, China (e-mail: dubowen@buaa.edu.cn).

Digital Object Identifier 10.1109/TITS.2020.3002718

I. Introduction

CITY is the keystone of modern human living and individuals constantly migrate from rural areas to urban areas with urbanization. For instance, Delhi, the largest city in India, has a total of 29.4 million residents.[1] Such a huge population brings a great challenge to urban management, especially in traffic control [2]. To address this challenge, intelligent transportation systems (ITS) [3] have been exhaustively studied for decades and have emerged as an efficient way of improving the efficiency of urban transportation. As a crucial component in ITS, traffic flow prediction [4]–[6] has recently attracted widespread research interest in both academic and industry communities, due to its huge potentials in many real-world applications (e.g., intelligent traffic diversion and travel optimization).

In this paper, we aim to forecast the future traffic flow in a city with historical mobility data of taxicabs/bikes. Nowadays, we live in an era where ubiquitous digital devices are able to broadcast rich information about taxicabs/bikes mobility in real-time and at a high rate, which has rapidly increased the availability of large-scale mobility data (e.g., GPS signals or mobile phone signals). How to utilize these mobility data to predict traffic flow is still an open problem. In literature, numerous methods applied time series models (e.g., Auto-Regressive Integrated Moving Average (ARIMA) [8] and Kalman filtering [9]) to predict traffic flow at each individual location separately. Subsequently, some studies incorporated spatial information to conduct prediction [10], [11]. However, these traditional models can not well capture the complex spatial-temporal dependency of traffic flow and this task is still far from being well solved in complex traffic systems.

Recently, notable successes have been achieved for citywide traffic flow prediction based on deep neural networks coupled with certain spatial-temporal priors [7], [12]–[14]. In these works, the studied city is partitioned into a grid map based on the longitude and latitude, as shown in Fig. 1. The historical traffic flow maps/tensors generated from mobility data are fed into convolutional neural networks to forecast the future traffic flow. Nevertheless, there still exist several challenges limiting the performance of traffic flow analysis in complex scenarios. **First**, traffic flow data can vary greatly in temporal sequences and capturing such dynamic variations is non-trivial. However, previous methods [7], [14], [15] simply

[1] http://worldpopulationreview.com/world-cities/

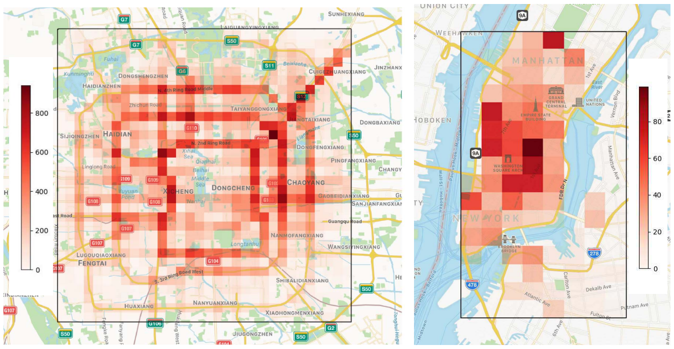


Fig. 1. Visualization of two traffic flow maps in Beijing and New York City. Following previous work [7], we partition a city into a grid map based on the longitude and latitude and generate the historical traffic flow maps by measuring the number of taxicabs/bikes in each region with mobility data. The weight of a specific grid indicates the traffic density of its corresponding region during a time interval. In this work, we take these historical maps as input to forecast the future traffic flow.

applied convolutional operations or LSTM units to directly map historical data to future flow prediction which are not able to well model the temporal patterns. **Second**, the spatial dependencies between locations are not strictly stationary and the relation significance of a specific region may change from time to time. Unfortunately, most of the existing methods do not consider such dynamic spatial dependence of traffic flow. **Third**, some internal periodic laws (e.g., traffic flow suddenly changing due to rush hours) and external factors (e.g., a precipitate rain) can greatly affect the situation of traffic flow, which increases the difficulty in learning traffic flow representations from data. Conventional works [7], [16] statically fuse these internal and external factors, which fail to flexibly generate effective representations to capture complex traffic flow patterns.

To solve all above issues, we propose a novel spatial-temporal neural network, called Attentive Traffic Flow Machine (ATFM), to adaptively exploit diverse factors that affect traffic flow evolution and at the same time produce the traffic flow estimation map in an end-to-end manner. The attention mechanism embedded in ATFM is designed to automatically discover the regions with primary impacts on the future flow prediction and simultaneously adjust the impacts of the different regions with different weights at each time-step. Specifically, our ATFM comprises two progressive ConvLSTM [1] units. The first one takes input from i) the original traffic flow features at each moment and ii) the memorized representations of previous moments, to compute the attentional weights. The second LSTM dynamically adjusts the spatial dependencies with the computed attentional map and generates superior spatial-temporal feature representation. The proposed ATFM has the three following appealing properties. **First**, it can effectively incorporate spatial-temporal information in feature representation and can flexibly compose solutions for traffic flow prediction with different types of input data. **Second**, by integrating the deep attention mechanism [17]–[20], ATFM adaptively learns to represent the weights of each spatial location at each time-step, which allows the model to dynamically perceive the impact of the given area at a given moment for the future traffic flow. **Third**, as a general and differentiable module, our ATFM can be effectively incorporated into various network architectures for end-to-end training.

Based on the proposed ATFM, we further develop a deep architecture for forecasting the citywide short-term traffic flow. Specifically, this customized framework consists of four components: **i)** a normal feature extraction module, **ii)** a sequential representation learning module, **iii)** a periodic representation learning module and **iv)** a temporally-varying fusion module. The middle two components are implemented by two parallel ATFMs for contextual dependencies modeling at different temporal scales, while the temporally-varying fusion module is proposed to adaptively merge the two separate temporal representations for traffic flow prediction. Finally, we extend and improve this framework to predict long-term traffic flow with an extra LSTM prediction network. Notice that our framework is general. Besides citywide traffic flow prediction, it can also be applied to extensive traffic tasks (e.g., citywide passenger demand prediction, crowd flow prediction), if the following preprocessing procedures are satisfied: **i)** the studied city is divided into a regular grid map and the raw traffic data is transformed into tensors, which is the most common form of structured data to fit the deep neural networks; **ii)** the sequential data and periodic data have been recorded; **iii)** the external factors (e.g., holiday information and meteorology information) are available, or else this submodule can be directly ignored.

In summary, the contributions of this work are three-fold:

- We propose a novel neural network module called Attentive Traffic Flow Machine (ATFM), which incorporates two ConvLSTM units with an attention mechanism to infer the evolution trend of traffic flow via dynamic spatial-temporal feature representations learning.
- We integrate the proposed ATFM in a customized deep framework for citywide traffic flow prediction, which effectively incorporates the sequential and periodic dependencies with a temporally-varying fusion module.
- Extensive experiments on two public benchmarks of traffic flow prediction demonstrate the superiority of the proposed method.

A preliminary version of this work is published in [21]. In this work, we inherit the idea of dynamically learning the spatial-temporal representations and provide more details of the proposed method. Moreover, we extend this customized framework to forecast long-term traffic flow. Further, we conduct a more comprehensive ablation study on our method and present more comparisons with state-of-the-art models under different settings (e.g., weekday, weekend, day and night). Finally, we apply the proposed method to forecast the passenger pickup/dropoff demands and show that our method is generalizable to various traffic prediction tasks.

The rest of this paper is organized as follows. First, we review some related works of traffic flow analysis in Section II and provide some preliminaries of this task in Section III. Then, we introduce the proposed ATFM in Section IV and develop two unified frameworks to forecast short-term/long-term traffic flow in Section V.

Extensive evaluation and comparisons are conducted in Section VI. Finally, we conclude this paper in Section VII.

II. Related Work

A. Traffic Flow Analysis

As a crucial task in ITS, traffic flow analysis has been studied for decades [22], [23] due to its wide applications in urban traffic management and public safety monitoring. Traditional approaches [8], [24], [25] usually used time series models (e.g., Vector Auto-Regression [26], ARIMA [27] and their variants [22]) for traffic flow prediction. However, most of these earlier methods modeled the evolution of traffic flow for each individual location separately and cannot well capture the complex spatial-temporal dependency.

Recently, deep learning based methods have been widely used in various traffic-related tasks [28]–[32]. Inspired by these works, many researchers have attempted to address traffic flow prediction with deep learning algorithms. For instance, an artificial neural network termed ST-ANN [7] was proposed to forecast traffic flow by extracting both the spatial (values of 8 regions in the neighborhood) and temporal (8 previous time intervals) features. In [12], a DNN-based model DeepST was proposed to capture various temporal properties (i.e. temporal closeness, period and trend). In [7], a deep ST-ResNet framework was developed with ResNet [33] to leverage the temporal closeness, period and trend information for citywide traffic flow prediction. Xu *et al.* [13] designed a cascade multiplicative unit to model the dependencies between multiple frames and applied it to forecast the future traffic flow. Zhao *et al.* [15] proposed a unified traffic forecast model based on long short-term memory network for short-term traffic flow forecast. Geng *et al.* [34] developed a multi-graph convolution network to encode the non-Euclidean pair-wise correlations among regions for spatiotemporal forecasting. Currently, to overcome the scarcity of traffic flow data, Wang *et al.* [35] proposed to learn the target city model from the source city model with a region based cross-city deep transfer learning algorithm. Yao *et al.* [36] incorporate the meta-learning paradigm into networks to tackle the problem of traffic flow prediction for the cities with only a short period of data collection. However, the above-mentioned algorithms have two major disadvantages. First, some of them [7], [12], [13] simply employed convolution operations to extract temporal features and could not fully explore the temporal patterns. Second, all of them neglected the dynamic dependencies of spatial regions and failed to capture complex spatial patterns. In contrast, our ATFM incorporates two progress ConvLSTM units with a spatial attention map to effectively learn dynamic spatial-temporal features.

B. Temporal Sequences Modeling

Recurrent neural network (RNN) is a special class of artificial neural network for temporal sequences modeling. As an advanced variation, Long Short-Term Memory Networks (LSTM) enables RNNs to store information over extended time intervals and exploit longer-term temporal dependencies. Recently, LSTM has been widely applied to various sequential prediction tasks, such as natural language processing [37] and speech recognition [38]. Many works in computer vision community [39]–[41] also combined CNN with LSTM to model the spatial-temporal information and achieved substantial progress in various tasks, such as video prediction. For instance, in [39], a Video Pixel Network (VPN) learned the temporal relationships of previous frames in video with ConvLSTM to forecast the content of the next several frames. A predictive neural network (PredNet [40]) used multiple LSTM-based layers to predict future frames in a video sequence, with each layer making local predictions and only forwarding deviations from those predictions to subsequent layers. PredRNN [41] utilized some stacked spatial-temporal LSTM layers to memorize both spatial and temporal variations of input frames. Without doubts, these models can be implemented and retained to forecast traffic flow, but they mainly focus on temporal modeling and are not aware of the dynamic spatial dependencies of traffic flow.

Inspired by the success of the aforementioned works, many researchers [42]–[44] have attempted to address traffic flow prediction with recurrent neural networks. However, existing works simply apply LSTM to extract feature and also cannot fully model the spatial-temporal evolution of traffic flow. Thus, a comprehensive module that can simultaneously learn the dynamic dependencies of both spatial view and temporal view is extremely desired for traffic flow prediction.

C. Attention Mechanism

Visual attention [17], [18] is a fundamental aspect of the human visual system, which refers to the process by which humans focus the computational resources of their brain's visual system to specific regions of the visual field while perceiving the surrounding world. It has been recently embedded in deep convolution networks or recurrent neural networks to adaptively attend on mission-related regions while processing feedforward operations [45], [46]. For instance, in the task of visual question answering, Xu and Saenko [47] chose some question-related regions dynamically with spatial attention to answer the questions about a given image. In crowd counting, Liu *et al.* [19] utilized an attention mechanism to select some local regions of the input image and then conducted local density map refinement. Tay *et al.* [20] integrated person attributes and attribute attention maps into a classification framework to solve the person re-identification problem. Inspired by these work, our ATFM computes the attention weights of spatial regions at each time intervals and incorporates two ConvLSTM units to dynamically learn the spatial-temporal representations. Thanks to this simple yet effective attention mechanism, our method can favorably model the dynamic spatial-temporal dependencies of traffic flow.

III. Preliminaries

In this section, we first introduce some basic elements of traffic flow and then elaborate the definition of the traffic flow prediction problem.

A. Region Partition

There are many ways to divide a city into multiple regions in terms of different granularities and semantic meanings, such as road network [11] and zip code tabular [48]. In this work, we follow the previous work [12] to partition a city into $h \times w$ non-overlapping grid map based on the longitude and latitude. Each rectangular grid represents a different geographical region in the city. All partitioned regions of Beijing and New York City are shown in Fig.1. With this simple partition strategy, the raw mobility data could be easily transformed into a matrix or tensor, which is the most common format of input data of the deep neural networks.

B. Traffic Flow Map

In some practical applications, we can extract a mass of taxicabs/bikes trajectories from GPS signals or mobile phone signals. With these trajectories, we measure the number of vehicles/bikes entering or leaving a given region at each time interval, which are called as inflow and outflow in our work. For convenience, we denote the traffic flow map at the t^{th} time interval of d^{th} day as a tensor $\boldsymbol{M_d^t} \in R^{2\times h\times w}$, in which the first channel is the inflow and the second channel denotes the outflow. Some examples of traffic flow maps are visualized in Fig. 11.

C. External Factors

As mentioned in [7], traffic flow can be affected by many complex external factors. For example, a sudden rain may seriously affect the traffic flow evolution and people would gather in some commercial areas for celebration on New Year's Eve. In this paper, we also consider the effect of meteorology information and holiday information, and their detail descriptions on different benchmarks can be found in Section VI-A.

i) Meteorological preprocessing: Some meteorology factors (e.g., weather condition, temperature and wind speed) can be collected from a public website Wunderground.[2] Specifically, the weather condition is categorized into multiple categories (e.g., sunny and rainy) and it is digitized with One-Hot Encoding [49], while temperature and wind speed are scaled into the range [0, 1] with a min-max linear normalization.

ii) Holiday preprocessing: Multiple categories of holiday (e.g., Chinese Spring Festival and Christmas) can be acquired from a calendar and encoded into a binary vector with One-Hot Encoding. We concatenate all data of external factors to a 1D tensor. The tensor of external factors at the t^{th} time interval of d^{th} day is represented as $\boldsymbol{E_d^t}$ in the following sections.

D. Traffic Flow Prediction

Given the historical traffic flow maps and data of external factors until the t^{th} time interval of d^{th} day, we aim to predict the traffic flow map M_d^{t+1}, which is called as short-term prediction in our work. Moreover, we also extend our model to conduct long-term prediction, in which we forecast the traffic flow at the next several time intervals.

IV. Attentive Traffic Flow Machine

In this section, we propose a unified neural network, named Attentive Traffic Flow Machine (ATFM), to learn the spatial-temporal representations of traffic flow. ATFM is designed to adequately capture various contextual dependencies of the traffic flow, e.g., the spatial consistency and the temporal dependency of long and short term. As shown in Fig. 2, the proposed ATFM consists of two progressive ConvLSTM units connected with a convolutional layer for attention weight prediction at each time step. Specifically, the first ConvLSTM unit learns temporal dependency from the normal traffic flow features, the extraction process of which is described in Section V-A.1. The output hidden state encodes the historical evolution information and it is concatenated with the current traffic flow feature for spatial weight map inference. The second ConvLSTM unit takes the re-weighted traffic flow features as input at each time-step and is trained to recurrently learn the spatial-temporal representations for further traffic flow prediction.

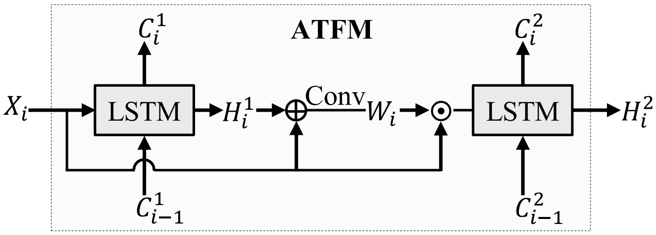


Fig. 2. Overview of the proposed Attentive Traffic Flow Machine (ATFM). X_i is the normal traffic flow feature of the i^{th} iteration. "$\bigoplus$" denotes a feature concatenation operation and "$\odot$" refers to an element-wise multiplication operation. The first ConvLSTM unit takes X_i as input and incorporates the historical information to dynamically generate a spatial attention map W_i. The second ConvLSTM unit learns a more effective spatial-temporal feature representation from the attentionally weighted traffic flow features.

Let us denote the input feature of traffic map at the i^{th} iteration as $X_i \in R^{c\times h\times w}$, with h, w and c representing the height, width and the number of channels. At each iteration, the first ConvLSTM unit takes X_i as input and updates its memorized cell state C_i^1 with an input gate $\mathcal{I}_i^1$ and a forget gate $\mathcal{F}_i^1$. Meanwhile, it updates its new hidden state H_i^1 with an output gate $\mathcal{O}_i^1$. The computation process of our first ConvLSTM unit is formulated as:

$$\begin{aligned}
\mathcal{I}_i^1 &= \sigma\left(w_{xi} * X_i + w_{hi} * H_{i-1}^1 + w_{ci} \odot C_{i-1}^1 + b_i\right)\\
\mathcal{F}_i^1 &= \sigma\left(w_{xf} * X_i + w_{hf} * H_{i-1}^1 + w_{cf} \odot C_{i-1}^1 + b_f\right)\\
C_i^1 &= \mathcal{F}_i^1 \odot C_{i-1}^1 + \mathcal{I}_i^1 \odot tanh\left(w_{xc} * X_i + w_{hc} * H_{i-1}^1 + b_c\right)\\
\mathcal{O}_i^1 &= \sigma\left(w_{xo} * X_i + w_{ho} * H_{i-1}^1 + w_{co} \odot C_i^1 + b_0\right)\\
H_i^1 &= \mathcal{O}_i^1 \odot tanh\left(C_i^1\right)
\end{aligned} \tag{1}$$

where $w_{\alpha\beta}$ $(\alpha \in \{x, h, c\}, \beta \in \{i, f, o, c\})$ are the parameters of convolutional layers in ConvLSTM. σ denotes the logistic sigmoid function and $\odot$ is an element-wise multiplication operation. For notation simplification, we denote Eq.(1) as:

$$H_i^1, C_i^1 = \mathbf{ConvLSTM}(H_{i-1}^1, C_{i-1}^1, X_i). \tag{2}$$

Generated from the memorized cell state C_i^1, the new hidden state H_i^1 encodes the dynamic evolution of historical traffic flow in temporal view.

[2] https://www.wunderground.com/

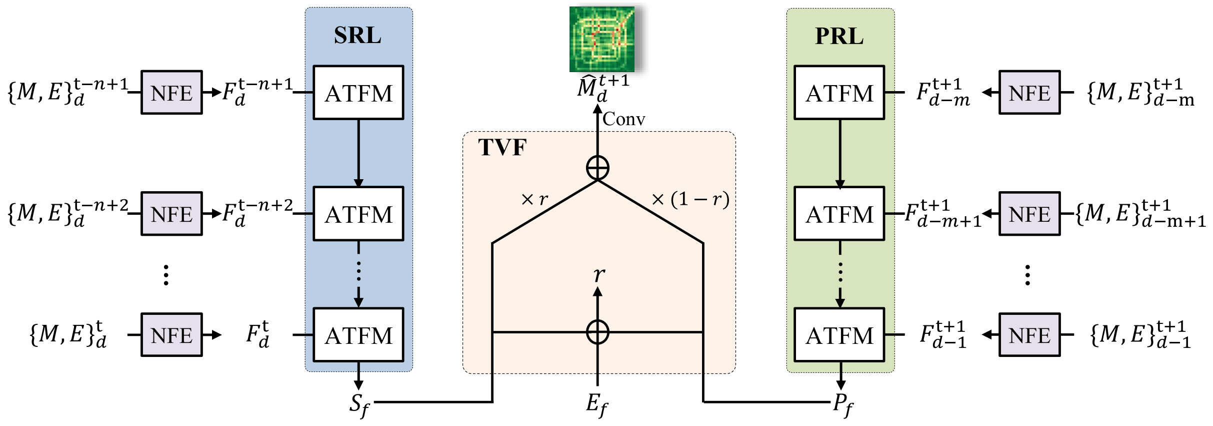


Fig. 3. The architecture of SPN based on ATFM for citywide short-term traffic flow prediction. It consists of four components: (1) a normal feature extraction (**NFE**) module, (2) a sequential representation learning (**SRL**) module, (3) a periodic representation learning (**PRL**) module and (4) a temporally-varying fusion (**TVF**) module. $\{M, E\}_j^i$ denotes the traffic flow map M_j^i and external factors tensor E_j^i at the i^{th} time interval of the j^{th} day. F_j^i is the embedded feature of M_j^i and E_j^i. S_f and P_f are sequential representation and periodic representation, while external factors integrative feature E_f is the element-wise addition of external factors features of all relative time intervals. "$\bigoplus$" refers to feature concatenation. The symbols r and $(1-r)$ reflect the importance of S_f and P_f respectively. $\widehat{M}_d^{t+1}$ is the predicted traffic flow map.

We then integrate a deep attention mechanism to dynamically model the spatial dependencies of traffic flow. Specifically, we incorporate the historical state H_i^1 and current state X_i to infer an attention map W_i, which is implemented by:

$$W_i = \mathbf{Conv}_{1\times 1}(H_i^1 \oplus X_i, w_a), \tag{3}$$

where $\oplus$ denotes a feature concatenation operation and w_a is the parameters of a convolutional layer with a kernel size of 1×1. The attention map W_i is learned to discover the weights of each spatial location on the input feature map X_i.

Finally, we learn a more effective spatial-temporal representation with the guidance of attention map. After reweighing the normal traffic flow feature map by multiplying X_i and W_i element by element, we feed it into the second ConvLSTM unit and generate a new hidden state $H_i^2 \in R^{c\times h\times w}$, which is expressed as:

$$H_i^2, C_i^2 = \mathbf{ConvLSTM}(H_{i-1}^2, C_{i-1}^2, X_i \odot W_i), \tag{4}$$

where H_i^2 encodes the attention-aware content of current input as well as memorizes the contextual knowledge of previous moments. When the elements in a sequence of traffic flow maps are recurrently fed into ATFM, the last hidden state encodes the information of the whole sequence and it can be used as the spatial-temporal representation for evolution analysis of future flow map.

V. Citywide Traffic Flow Prediction

In this section, we first develop a deep neural network framework which incorporates the proposed ATFM for citywide short-term traffic flow prediction. We then extend this framework to predict long-term traffic flow with an extra LSTM prediction network. Notice that our framework is general and can be applied for other traffic prediction tasks, such as the citywide passenger demand prediction described in Section VI-E.

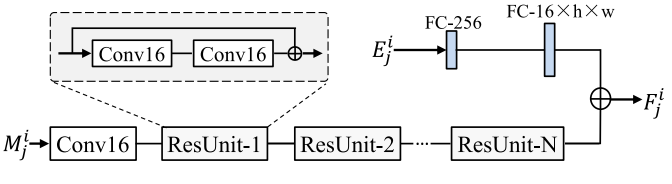


Fig. 4. The architecture of the subnetwork for normal feature extraction. It is designed as a concatenation of the embedded traffic flow feature and the external factor feature. Conv16 is a convolutional layer with 16 channels and FC-k denotes a fully-connected layer with k output neurons.

A. Short-Term Prediction

As illustrated in Fig. 3, our short-term prediction framework consists of four components: (1) a normal feature extraction (**NFE**) module, (2) a sequential representation learning (**SRL**) module, (3) a periodic representation learning (**PRL**) module and (4) a temporally-varying fusion (**TVF**) module. First, the NFE module is used to extract the normal features of traffic flow map and external factors tensor at each time interval. Second, the SRL and PRL modules are employed to model the contextual dependencies of traffic flow at two different temporal scales. Third, the TVF module adaptively merges the feature representations of SRL and PRL with the fused weight learned from the comprehensive features of various factors. Finally, the fused feature map is fed to one additional convolution layer for traffic flow map inference. For convenience, this framework is denoted as **S**equential-**P**eriodic **N**etwork (**SPN**) in following sections.

1) **Normal Feature Extraction:** We first describe how to extract the normal features of traffic flow and external factors, which will be further fed into the SRL and PRL modules for dynamic spatial-temporal representation learning.

As shown in Fig. 4, we utilize a customized ResNet [33] to learn feature embedding from the given traffic flow map M_j^i. Specifically, our ResNet consists of N residual units, each of which has two convolutional layers with channel number of 16 and kernel size of 3×3. To maintain the resolution

$h \times w$, we set the stride of all convolutional layers to 1 and do not adopt any pooling layers in ResNet. Following [7], we first scale M_j^i into the range $[-1, 1]$ with a min-max linear normalization and then feed it into the ResNet to generate the traffic flow feature, which is denoted as $F_j^i(M) \in R^{16\times h\times w}$.

Then, we extract the feature of the given external factors tensor E_j^i with a Multilayer Perceptron. We implement it with two fully-connected layers. The first layer has 40 output neurons and the second one has $16 \times h \times w$ output neurons. We reshape the output of the last layer to form the 3D external factor feature $F_j^i(E) \in R^{16\times h\times w}$. Finally, we fuse $F_j^i(M)$ and $F_j^i(E)$ to generate an embedded feature F_j^i, which is expressed as:

$$F_j^i = F_j^i(M) \oplus F_j^i(E), \tag{5}$$

where $\oplus$ denotes feature concatenation. F_j^i is the normal feature at a specific time interval and it is unaware of the dynamic spatial dependencies of traffic flow. Thus, the following two modules are proposed to dynamically learn the spatial-temporal representation.

2) **Sequential Representation Learning:** The evolution of citywide traffic flow is usually affected by the recent traffic states. For instance, a traffic accident occurring on a main road of the studied city during morning rush hours may seriously affect the traffic flow of nearby regions in subsequent time intervals. In this subsection, we develop a sequential representation learning (SRL) module based on the proposed ATFM to fully model the evolution trend of traffic flow.

First, we take the normal traffic flow features of several recent time intervals to form a group of sequential temporal features, which is denoted as:

$$S_{in} = \{F_d^{t-k} \big| k = n-1, n-2, \ldots, 0\}, \tag{6}$$

where n is the length of the sequentially related time intervals. We then apply the proposed ATFM to learn sequential representation from the temporal features S_{in}. As shown on the left of Fig. 3, at each iteration, ATFM takes one element in S_{in} as input and learns to selectively memorize the spatial-temporal context of the sequential traffic flow. Finally, we get the sequential representation $S_f \in R^{16\times h\times w}$ by feeding the last hidden state of ATFM into a 1×1 convolution layer. S_f encodes the sequential evolution trend of traffic flow.

3) **Periodic Representation Learning:** In urban transportation systems, there exist some periodicities which make a significant impact on the changes of traffic flow. For example, the traffic conditions are very similar during morning rush hours of consecutive workdays, repeating every 24 hours. Thus, in this subsection, we propose a periodic representation learning (PRL) module that fully captures the periodic dependencies of traffic flow with the proposed ATFM.

Similar to the sequential representation learning, we first construct a group of periodic temporal features

$$P_{in} = \{F_{d-k}^{t+1} \big| k = m, m-1, \ldots, 1\}, \tag{7}$$

where n is the length of the periodic days. At each iteration, we feed one element in P_{in} into ATFM to dynamically learn the periodic dependencies, as shown on the right of Fig. 3. After the last iteration, we feed the hidden state of ATFM into a 1×1 convolutional layer to generate the final periodic representation $P_f \in R^{16\times h\times w}$. Encoding the periodic evolution trend of traffic flow, P_f is proved to be effective for traffic prediction in our experiments.

4) **Temporally-Varying Fusion:** As described in the two previous modules, the future traffic flow is affected by the sequential representation S_f and the periodic representation P_f simultaneously. We find that the relative importance of these two representations is temporally dynamic and it is suboptimal to directly concatenate them without any specific preprocessing. To address this issue, we propose a novel temporally-varying fusion (TVF) module to adaptively fuse the representations S_f and P_f with different weights learned from the comprehensive features of various internal and external factors.

In TVF module, we take the sequential representation S_f, the periodic representation P_f and the external factors integrative feature E_f to determine the fusion weight. Specifically, E_f is the element-wise addition of the external factors features $\{F(E)_d^{t-k} \big| k = n-1, n-2, \ldots, 0\}$ and $\{F(E)_{d-k}^{t+1} \big| k = m, m-1, \ldots, 1\}$. As shown in Fig. 3, we first feed the concatenation of S_f, P_f and E_f into two fully-connected layers for fusion weight inference. The first layer has 32 output neurons and the second one has only one neuron. We then obtain the fusion weight of S_f by applying a sigmoid function on the output of the second FC layer. The weight of P_f is automatically set to $1 - r$. We then fuse these two temporal representations on the basis of the learned weights and compute a comprehensive spatial-temporal representation SP_f as:

$$\mathrm{SP}_f = r \times S_f \oplus (1 - r) \times P_f, \tag{8}$$

where SP_f contains the sequential and periodic dependencies of traffic flow.

Finally, we feed SP_f into a 1×1 convolutional layer with two filters to predict the future traffic flow map $\widehat{M}_d^t \in R^{2\times h\times w}$ with following formula:

$$\widehat{M}_d^t = tanh(\mathrm{SP}_f * w_p). \tag{9}$$

where w_p is the parameters of the predictive convolutional layer and the hyperbolic tangent $tanh$ ensures the output values are within the range $[-1, 1]$. Further, the predicted map $\widehat{M}_d^t$ is re-scaled back to normal value with an inverted min-max linear normalization.

B. Long-Term Prediction

In this subsection, we extend our method to predict the longer-term traffic flow. With a similar setting of short-term prediction, we incorporate the sequential data and periodic data at previous time intervals to forecast the traffic flow at the next four time intervals. For convenience, we denote this model as SPN-LONG in the following sections.

The architecture of our SPN-LONG is shown in Fig 5. For each previous time interval, we first extract its normal features F_j^i with the proposed NFE module. Then, the features in $\{F_d^{t-k} \big| k = n-1, n-2, \ldots, 0\}$ are recurrently fed into ATFM to learn the sequential representation. The output

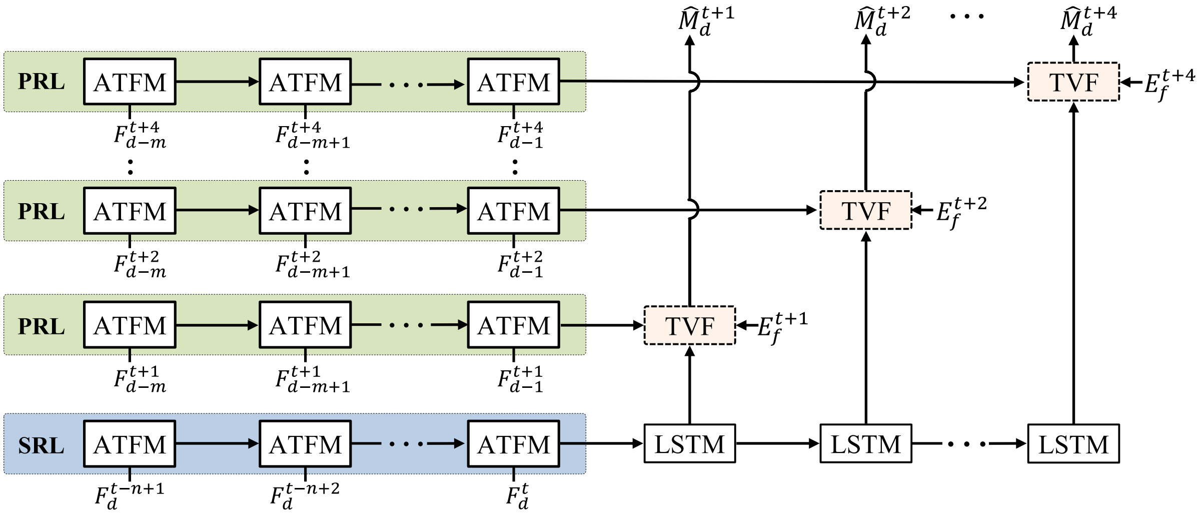


Fig. 5. The architecture of the SPN-LONG for Long-term Traffic Flow Prediction. F_j^i is the normal traffic flow feature described in Section V-A.1. E_f^{t+i} is the element-wise addition of external factors features $\{E_d^{t-k}|k = n-1, \ldots, 0\}$ and $\{E_{d-k}^{t+i}|k = m, \ldots, 1\}$.

TABLE I

THE OVERVIEW OF TAXIBJ AND BIKENYC DATASETS. "# TAXIS/BIKES" DENOTES THE NUMBER OF TAXIS OR BIKES IN THE DATASETS. OTHER TEXTS WITH "#" HAVE SIMILAR MEANINGS

Dataset		**TaxiBJ**	**BikeNYC**
Traffic Flow	City	Beijing	New York
	Gird Map Size	(32, 32)	(16, 8)
	Data Type	Taxi GPS	Bike Rent
	Time Span	7/1/2013 - 10/30/2013 3/1/2014 - 6/30/2014 3/1/2015 - 6/30/2015 11/1/2015 - 4/10/2016	4/1/2014 - 9/30/2014
	# Taxis/Bikes	34,000+	6,800+
	Time Interval	0.5 hour	1 hour
	# Available Time Interval	22,459	4,392
External Factors	# Holidays	41	20
	Weather Conditions	16 types (e.g., Sunny, Rainy)	\
	Temperature / °C	$[-24.6, 41.0]$	\
	Wind Speed / mph	$[0, 48.6]$	\

sequential representation is then fed into a LSTM prediction network. With four ConvLSTM units, this prediction network is designed to forecast the traffic flow at the next four time intervals. Specifically, at i^{th} LSTM, we use a TVF module to adaptively fuse its hidden state and the periodic representation learned from $\{F_{d-k}^{t+i}|k = m, \ldots, 1\}$. The external factors integrative feature E_f^{t+i} is the element-wise addition of $\{E_d^{t-k}|k = n-1, \ldots, 0\}$ and $\{E_{d-k}^{t+i}|k = m, \ldots, 1\}$. Finally, we take the output of i^{th} TVF module to predict $\widehat{M}_d^{t+i}$ with a convolutional layer.

VI. EXPERIMENTS

In this section, we first introduce the commonly-used benchmarks and evaluation metrics of citywide traffic flow prediction. Then, we compare the proposed approach with several state-of-the-art methods under different settings. Furthermore, we conduct extensive component analysis to demonstrate the effectiveness of each part in our model. Finally, we apply the proposed method to passenger pickup/dropoff demands forecasting and show its generalization for other traffic prediction tasks.

A. Experimental Setting

*1) **Dataset:*** In this work, we forecast the inflow and outflow of citywide transportation entities on two representative benchmarks, including the TaxiBJ dataset [7] for taxicab flow prediction and the BikeNYC dataset [12] for bike flow prediction. These two datasets can be accessed publicly and various comparison algorithms can be evaluated on the same testing sets for fair comparisons. The summaries of TaxiBJ and BikeNYC are shown in Table I[3].

TaxiBJ Dataset: In this dataset, a mass of taxi GPS trajectories are collected from 34 thousand taxicabs in Beijing for over 16 months. The time interval is half an hour and 22,459 traffic flow maps with size 32 × 32 are generated from these trajectory data. The external factors contain weather conditions, temperature, wind speed and 41 categories of

[3] The details of TaxiBJ and BikeNYC dataset are from quoted from [7].

holidays. This dataset is officially divided into a training set and a testing set. The number of testing data is around 6% of that of training data. Specifically, the data in the last four weeks are used for evaluation and the rest data are used for training.

BikeNYC Dataset: Generated from the NYC bike trajectory data for 182 days, this dataset contains 4,392 traffic flow maps with a time interval of one hour and the size of these maps is 16 × 8. As for external factors, 20 holiday categories are recorded. This dataset has the similar training-testing ratio of TaxiBJ. Specifically, the data of the first 172 days are used for training and the data of the last ten days are chosen to be the testing set.

*2) **Implementation Details:*** We adopt the PyTorch [50] toolbox to implement our traffic flow prediction network. The sequential length n and the periodic length m are set to 4 and 2, respectively. For fair comparison with ST-ResNet [7], we develop the customized ResNet in Section V-A.1 with 12 residual units on the TaxiBJ dataset and 4 residual units on the BikeNYC dataset. The filter weights of all convolutional layers and fully-connected layers are initialized by Xavier [51]. The size of a minibatch is set to 64 and the learning rate is 10^{-4}. We optimize the parameters of our network in an end-to-end manner via Adam optimization [52] by minimizing the Euclidean loss.

*3) **Evaluation Metric:*** In traffic flow prediction, Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) are two popular evaluation metrics used to measure the performances of related methods. Specifically, they are defined as:

$$\text{RMSE} = \sqrt{\frac{1}{z}\sum_{i=1}^{z}(\widehat{Y_i} - Y_i)^2}, \quad \text{MAE} = \frac{1}{z}\sum_{i=1}^{z}|\widehat{Y_i} - Y_i| \quad (10)$$

where $\widehat{Y_i}$ and Y_i represent the predicted flow map and its ground truth, respectively. z indicates the number of samples used for validation. Noted that some partitioned regions in New York City are water areas and their flow are always zero, which may decrease the mean error and affect the evaluation of algorithm performance. To correctly reflect the performance of different methods on BikeNYC dataset, we re-scale their mean errors with a ratio (1.58) provided by ST-ResNet.

B. Comparison for Short-Term Prediction

In this subsection, we compare the proposed method with ten typical methods for short-term traffic flow prediction. These compared methods can be divided into three categories, including: **(i)** traditional models for time series forecasting, **(ii)** deep learning networks particularly designed for traffic flow prediction and **(iii)** the state-of-the-art approaches originally designed for some related tasks. The details of the compared methods are described as follows.

- **HA:** Historical Average (HA) is a simple model that directly predicts the future traffic flow by averaging the historical flow in the corresponding periods. For example, the predicted flow at 7:00 am to 7:30 am on a specific Tuesday is the average flow from 7:00 am to 7: 30 am on all historical Tuesdays.

TABLE II

QUANTITATIVE COMPARISONS ON TAXIBJ AND BIKENYC. OUR METHOD OUTPERFORMS THE EXISTING METHODS ON BOTH DATASETS

Method	TaxiBJ		BikeNYC	
	RMSE	MAE	RMSE	MAE
HA	57.79	-	21.57	-
SARIMA	26.88	-	10.56	-
VAR	22.88	-	9.92	-
ARIMA	22.78	-	10.07	-
ST-ANN	19.57	-	-	-
DeepST	18.18	-	7.43	-
VPN	16.75	9.62	6.17	3.68
ST-ResNet	16.69	9.52	6.37	2.95
PredNet	16.68	9.67	7.45	3.71
PredRNN	16.34	9.62	5.99	4.89
SPN (Ours)	**15.31**	**9.14**	**5.59**	**2.74**

- **ARIMA [27]:** Auto-Regressive Integrated Moving Average (ARIMA) is a famous statistical analysis model that uses time series data to predict future trends.
- **SARIMA [22]:** Seasonal ARIMA (SARIMA) is an advanced variant of ARIMA that considers the seasonal terms.
- **VAR [26]:** Vector Auto-Regression (VAR) is a well-known stochastic process model and it can capture the linear interdependencies among multiple time series.
- **DeepST [12]:** This is a DNN-based model and it utilizes various temporal properties to conduct prediction.
- **ST-ANN [7]:** As an artificial neural network, this model extracts spatial (8 nearby region values) and temporal (8 previous time intervals) features for future traffic flow prediction.
- **ST-ResNet [7]:** As an advanced version of DeepST, this model incorporates the closeness, period, trend data as well as external factors to predict traffic flow with residual networks.
- **VPN [39]:** Video Pixel Networks (VPN) is a probabilistic video model designed for multi-frames prediction. A variant of VPN based on RMBs is implemented for traffic flow prediction.
- **PredNet [40]:** As a predictive neural network, this model is originally developed to predict the content of subsequent frame in a video sequence. We apply this method to traffic flow prediction.
- **PredRNN [41]:** This method is also originally designed for video generation and it is implemented by memorizing both spatial and temporal variations of input frames with a predictive recurrent neural network for future frames generation. In this work, it is re-implemented to forecast traffic flow.

1) Comparison on All Time Intervals: The performance of the proposed method and the other ten compared methods are summarized in Table II. Among these methods, the baseline model is HA that obtains a RMSE of 57.79 on the TaxiBJ dataset and 21.57 on the BikeNYC dataset. Although having some progress, the traditional time series algorithms (e.g., VAR, ARIMA, and SARIMA) still show inferior performance on both datasets, since these shallow models rely on hand-crafted features and have weak capacity to model complex patterns. Thanks to the deep representation learning, the recent CNN-based methods ST-ANN, DeepST, and ST-ResNet can decrease the errors to some extent. For instance, ST-ResNet reduces the RMSE to 16.59 on TaxiBJ and to 6.37 on BikeNYC. However, only with CNN features, these models fail to fully capture the temporal patterns. When applying recurrent neural networks to model the temporal evolution of traffic flow, the RNN-based methods VPN, PredNet and PredRNN can defeat the aforementioned CNN-based models. Nevertheless, the dynamic spatial dependencies of traffic flow are neglected in these methods and this task still cannot be solved perfectly. In contrast, our method can further improve the performance by explicitly learning the spatial-temporal feature and dynamically modeling the spatial attention weighting of each spatial influence. Specifically, our method achieves a RMSE of 15.31 on the TaxiBJ dataset, outperforming the previous best approach PredRNN by 6.3% relatively. On the BikeNYC dataset, our method also boosts the highest prediction accuracy, i.e., decreases the RMSE from 5.99 to 5.59, and outperforms other methods.

Notice that the official BikeNYC dataset does not contain meteorological information. To enrich the external factors of BikeNYC, we collect the information of weather conditions (31 types), temperature ($[-1.1, 33.9]$) and wind speed ($[0, 33]$) from the popular meteorological website Wunderground. That meteorological information is processed with the same technique described in Section III. After combining these factors, our method further decreases the RMSE and MAE to 5.50 and 2.71, respectively. For fair comparison with other methods, we mainly report the performance trained with the official BikeNYC dataset in the following sections.

2) Comparison on Different Time Intervals: As previously described, traffic flow is time-varying and its temporal patterns are very complex. To explore the model's stability, we compare the performance of five deep learning-based methods at different time intervals, such as weekday (from Monday to Friday), weekend (Saturday and Sunday), day (from 6:00 to 18:00) and night (from 18:00 to 6:00). As shown in Fig. 6 and Fig. 7, our method outperforms other compared methods under various settings on both TaxiBJ and BikeNYC, since our ATFM can effectively learn the temporal patterns of traffic flow and the Temporally-Varying Fusion module can flexibly combine the information of different temporal sequences. These experiments well demonstrate the robustness of our method.

3) Comparison on High-Flow Regions: Since traffic flow is not uniformly distributed in space, some specific applications are more concerned about the predicted results on congested regions. In this section, we further measure the RMSE on some regions with high traffic flow. We first rank all regions of Beijing on the basis of the average traffic flow on the training set and then choose the top-p regions (p is a percentage) to conduct the evaluation. As shown in Fig. 8, on the TaxiBJ dataset, the RMSE of five deep learning-based methods are much larger on the top-10% regions and our method obtains a RMSE of 32.11, which shows that this task still has a lot of room for improvement. As the percentage p increases, the RMSE of all methods gradually decrease. As shown in Fig. 9, all methods perform poorly on top-10% regions of BikeNYC, ranging in RMSE from 8.81 to 11.31. As p increases from 10% to 70%, their errors gradually decline and no-longer become smaller, since the traffic flow of the remaining 30% regions is very low. In summary, our method consistently outperforms other methods under different flow density range p on both TaxiBJ and BikeNYC.

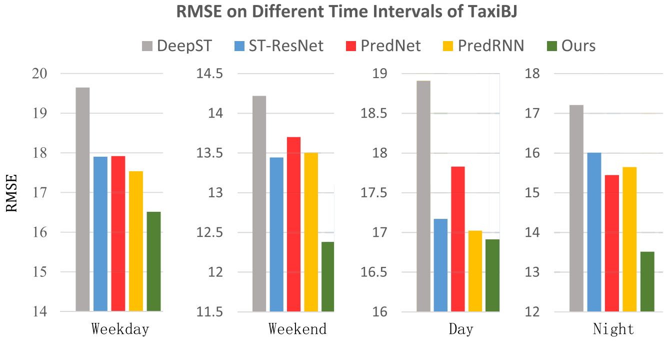


Fig. 6. The RMSE of weekday, weekend, day and night on the TaxiBJ dataset. The weekday RMSE is the average result from Monday to Friday, while the weekend RMSE is the average result of Saturday and Sunday. The day RMSE and the night RMSE are the average result from 6:00 to 18:00 and from 18:00 to 6:00, respectively. Best view in color.

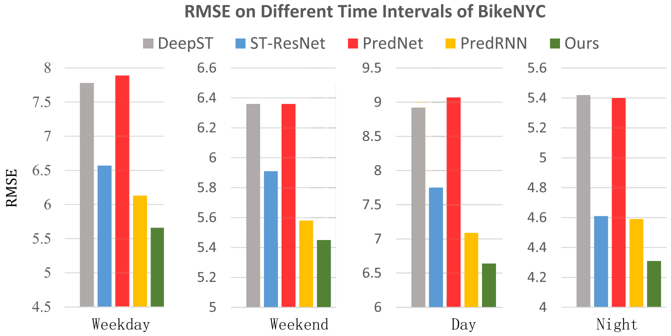


Fig. 7. The RMSE of weekday, weekend, day and night on the BikeNYC dataset.

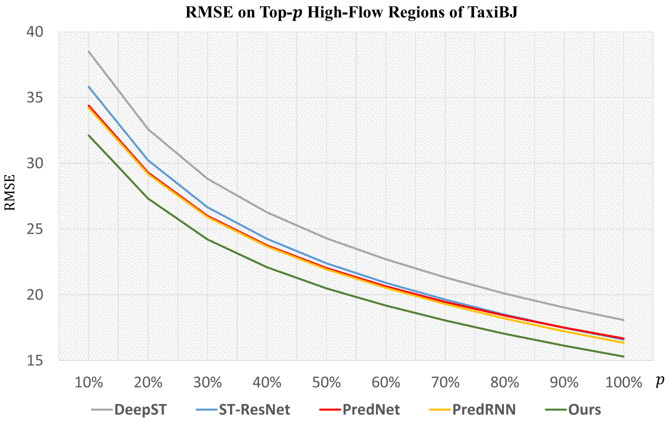


Fig. 8. The RMSE of five deep learning based methods on top-p regions with high traffic flow on the TaxiBJ dataset. p is a percentage. Specifically, we first rank all regions of Beijing on the basis of the average traffic flow and then conduct evaluations on the top-p regions. Best view in color.

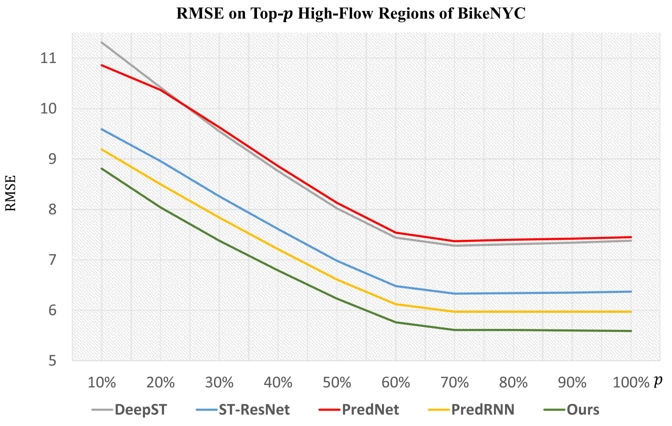


Fig. 9. The RMSE of five deep learning based methods on top-p regions with high traffic flow on the BikeNYC dataset.

TABLE III

RUNNING TIMES OF DIFFERENT METHOD ON BIKENYC DATASET

Model	Time (ms)
DeepST	0.18
ST-ResNet	2.08
PredNet	2.71
PredRNN	4.94
VPN	12.33
SPN (Ours)	7.17

These comparisons well demonstrate the superiority of our method.

4) Efficiency Comparison: Finally, we investigate the efficiency of different methods on the TaxiBJ dataset. The running times of six deep learning-based methods are measured with an NVIDIA 1060 GPU. As shown in Table III, DeepST costs 0.18 ms for each inference, while ST-ResNet, PredNet and PredRNN conduct a prediction within 5 ms. Only requiring 7.17 ms, our method is much faster than VPN. In summary, all methods can achieve practical efficiencies. Therefore, the running efficiency is not the bottleneck of this task and we should focus more on the improvement of the performance.

C. Comparison for Long-Term Prediction

In this subsection, we apply the customized SPN-LONG to predict long-term traffic flow and compare it with four deep learning based methods.[4] These compared methods have been finetuned for long-term prediction. As shown in Table IV, the RMSE of all methods gradually increases on the TaxiBJ dataset when attempting to forecast the longer-term flow. It can be observed that PredNet performs dreadfully in this scenario, since it was originally designed for single frame prediction and has a low capacity for long-term prediction. By contrast, our method has minor performance degradation and outperforms other methods at each time interval. Specifically, our method achieves the lowest RMSE 20.83 at the fourth time interval and has a relative improvement of 8.2%, compared with the previous best-performing method PredRNN. Moreover, we also evaluate the original SPN for long-term prediction and it is used to forecast traffic flow in a rolling style. As shown in the penultimate row of Table IV, it performs worse than SPN-LONG, thus we can conclude that it's essential to adapt and retrain SPN for long-term prediction. We also conduct long-term prediction on BikeNYC dataset. As shown in Table V, our SPN-LONG consistently outperforms other compared methods with the best RMSE (e.g., 5.81, 6.80, 7.54 and 7.90 for the 1^{st}-4^{th} time intervals, respectively). When combining our collected meteorological information, SPN-LONG further reduces the RMSE to 5.72, 6.24, 6.74 and 7.37 for the 1^{st}-4^{th} intervals, respectively. These experiments well demonstrate the effectiveness of the customized SPN-LONG for long-term traffic flow prediction.

TABLE IV

QUANTITATIVE COMPARISONS (RMSE) FOR LONG-TERM TRAFFIC FLOW PREDICTION ON TAXIBJ. ALL COMPARED METHODS HAVE BEEN FINETUNED FOR LONG-TERM PREDICTION. EACH TIME INTERVAL IS HALF AN HOUR (0.5 H) IN THIS DATASET

Method	Time Interval			
	1 (0.5 h)	2 (1.0 h)	3 (1.5 h)	4 (2.0 h)
ST-ResNet	16.75	19.56	21.46	22.91
VPN	17.42	20.50	22.58	24.26
PredNet	27.55	254.68	255.54	255.47
PredRNN	16.08	19.51	20.66	22.69
SPN (Ours)	15.31	19.59	23.70	28.61
SPN-LONG (Ours)	**15.42**	**17.63**	**19.08**	**20.83**

TABLE V

QUANTITATIVE COMPARISONS (RMSE) FOR LONG-TERM TRAFFIC FLOW PREDICTION ON BIKENYC. EACH TIME INTERVAL IS AN HOUR (1.0 H) IN THIS DATASET

Method	Time Interval			
	1 (1.0 h)	2 (2.0 h)	3 (3.0 h)	4 (4.0 h)
ST-ResNet	6.45	7.47	8.77	10.28
VPN	6.55	8.01	8.86	9.41
PredNet	7.46	8.95	10.08	10.93
PredRNN	5.97	7.37	8.61	9.40
SPN (Ours)	5.59	7.81	11.96	15.74
SPN-LONG (Ours)	**5.81**	**6.80**	**7.54**	**7.90**

D. Component Analysis

As described in Section V, our full model consists of four components: normal feature extraction, sequential representation learning, periodic representation learning and temporally-varying fusion module. In this section, we implement eight variants of our framework in order to verify the effectiveness of each component:

- **PCNN:** directly concatenates the periodic features P_{in} and feeds them to a convolutional layer with two filters followed by $tanh$ for future traffic flow prediction;
- **SCNN:** directly concatenates the sequential features S_{in} and feeds them to a convolutional layer followed by $tanh$ for future traffic flow prediction;

[4] On the TaxiBJ dataset, the performances of all compared methods for long-term prediction are directly quoted from [13]. On the BikeNYC dataset, there is not existing comparison for long-term prediction, thus we implement all compared methods and evaluate their performances.

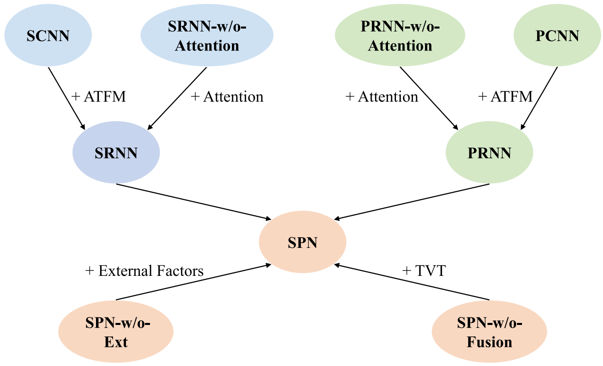


Fig. 10. Overview of the differences between all variants of our framework.

- **PRNN-w/o-Attention:** takes periodic features P_{in} as input and learns periodic representation with a LSTM layer to predict future traffic flow;
- **PRNN:** takes periodic features P_{in} as input and learns periodic representation with the proposed ATFM to predict future traffic flow;
- **SRNN-w/o-Attention:** takes sequential features S_{in} as input and learns sequential representation with a LSTM layer for traffic flow estimation;
- **SRNN:** takes sequential features S_{in} as input and learns sequential representation with the proposed ATFM to predict future traffic flow;
- **SPN-w/o-Ext:** does not consider the effect of external factors and directly trains the model with traffic flow maps;
- **SPN-w/o-Fusion:** directly merges sequential representation and periodic representation with equal weight (0.5) to predict future traffic flow.

The overview of all variants is shown in Fig. 10. First, we use "SCNN vs. SRNN" and "PCNN vs. PRNN" to verify the effectiveness of ATFM for sequential and periodic representation learning. Then, "SRNN-w/o-Attention vs SRNN" and "PRNN-w/o-Attention vs PRNN" are conducted to explain the effectiveness of spatial attention. Finally, "SPN-w/o-Ext vs. SPN" is utilized to illustrate the influence of external factors and "SPN-w/o-Fusion vs. SPN" is utilized to show the effectiveness of Temporally-Varying Fusion (TVF) module.

1) Effectiveness of ATFM for Sequential Representation Learning: As shown in Table VI, directly concatenating the sequential features S for prediction, the baseline variant SCNN gets an RMSE of 17.15. When explicitly modeling the sequential contextual dependencies of traffic flow using the proposed ATFM, SRNN decreases RMSE to 15.82, with 7.75% relative performance improvement compared to the baseline SCNN, which indicates the effectiveness of the sequential representation learning.

2) Effectiveness of ATFM for Periodic Representation Learning: We also explore different network architectures to learn the periodic representation. As shown in Table VI, the PCNN, which learns to estimate the flow map by simply concatenating all of the periodic features P, only achieves RMSE of 33.91. In contrast, when introducing ATFM to learn the periodic representation, the RMSE drops to 32.89. This experiment also well demonstrates the effectiveness of the proposed ATFM for spatial-temporal modeling.

TABLE VI

QUANTITATIVE COMPARISONS OF DIFFERENT VARIANTS OF OUR MODEL ON THE TAXIBJ DATASET FOR COMPONENT ANALYSIS

Model	RMSE	MAE
PCNN	33.91	17.16
PRNN-w/o-Attention	33.51	16.70
PRNN	32.89	16.64
SCNN	17.15	9.56
SRNN-w/o-Attention	16.20	9.43
SRNN	15.82	9.34
SPN-w/o-Ext	16.84	9.83
SPN-w/o-Fusion	15.67	9.40
SPN	15.31	9.14

3) Effectiveness of Spatial Attention: As shown in Table VI, adopting spatial attention, PRNN decreases the RMSE by 0.62, compared to PRNN-w/o-Attention. For another pair of variants, SRNN with spatial attention has similar performance improvement, compared to SRNN-w/o-Attention. Fig. 11 and Fig. 12 show some attentional maps generated by our method as well as the residual maps between the input traffic flow maps and their corresponding ground truth. We can observe that there is a negative correlation between the attentional maps and the residual maps to some extent. It indicates that our ATFM is able to capture informative regions at each time step and make better predictions by inferring the trend of evolution. Roughly, the greater difference a region has, the smaller its weight, and vice versa. We can inhibit the impacts of the regions with great differences by multiplying the small weights on their corresponding location features. With the visualization of attentional maps, we can also get to know which regions have the primary positive impacts for the future flow prediction. According to the experiment, we can see that the proposed model can not only effectively improve the prediction accuracy, but also enhance the interpretability of the model to a certain extent.

4) Necessity of External Factors: Without modeling the effect of external factors, the variant SPN-w/o-Ext obtains a RMSE of 16.84 on the TaxiBJ dataset and has a performance degradation of 10%, compared to SPN. The main reason of degradation lies in that some notable meteorological conditions (e.g., rain and snow) or holidays would seriously affect the traffic flow. Thus, it's necessary to incorporate the external factors to model the traffic flow evolution.

5) Effectiveness of Temporally-Varying Fusion: When directly merging the two temporal representations with an equal contribution (0.5), SPN-w/o-fusion achieves a negligible improvement, compared to SRNN. In contrast, after using our proposed fusion strategy, the full model SPN decreases the RMSE from 15.82 to 15.31, with a relative improvement of 3.2% compared with SRNN. The results show that the contributions of these two representations are not equal and are influenced by various factors. The proposed fusion strategy is effective to adaptively merge the different temporal representations and further improve the performance of traffic flow prediction.

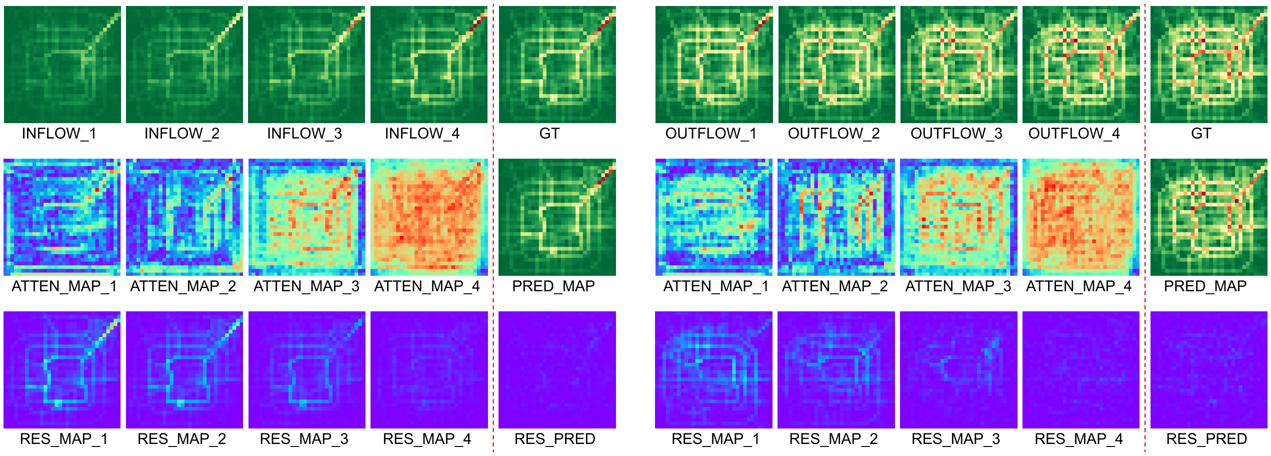


Fig. 11. Illustration of the generated attention maps of the traffic flow in sequential representation learning with n set as 4. Every five columns form one group. In each group: i) on the first row, the first four images are the input sequential inflow/outflow maps and the last one is the ground truth inflow/outflow map of next time interval; ii) on the second row, the first four images are the attentional maps generated by our ATFM, while the last one is our predicted inflow/outflow map; iii) on the third row, the first four images are the residual maps between the input flow maps and the ground truth, while the last one is the residual map between our predicted flow map and the ground truth. We can observe that there is a negative correlation between the attentional maps and the residual maps to some extent.

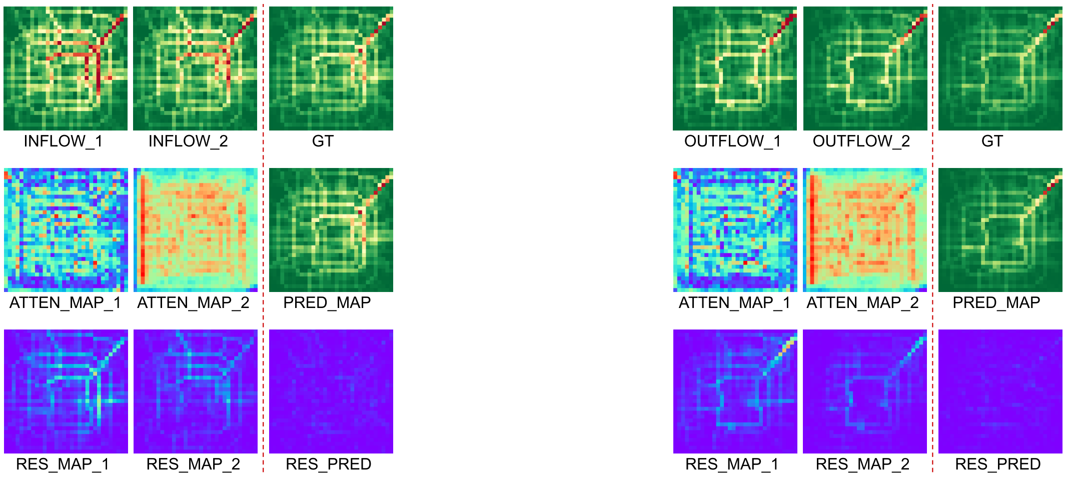


Fig. 12. Illustration of the generated attentional maps of the traffic flow in periodic representation learning with m set as 2. Every three columns form one group. In each group: i) on the first row, the first two images are the input periodic inflow/outflow maps and the last one is the ground truth inflow/outflow map of next time interval; ii) on the second row, the first two images are the attentional maps generated by our ATFM, while the last one is our predicted inflow/outflow map; iii) on the third row, the first two images are the residual maps between the input flow maps and the ground truth, while the last one is the residual map between our predicted flow map and the ground truth. We can observe that there is a negative correlation between the attentional maps and the residual maps to some extent.

6) Further Discussion: To analyze how each temporal representation contributes to the performance of traffic flow prediction, we measure the average fusion weights of two temporal representations at each time interval on the testing set. As shown in the left of Fig. 13, the fusion weights of sequential representation are greater than that of the periodic representation. To explain this phenomenon, we further measure **i)** the RMSE of traffic flow between two consecutive time intervals, denoted as “Pre-Hour”, and **ii)** the RMSE of traffic flow between two adjacent days at the same time interval, denoted as “Pre-Day”. As shown on the right of Fig. 13, the RMSE of “Pre-Day” is much higher than that of “Pre-Hour” at most time except for the wee hours. Based on this observation, we can conclude that the sequential representation is more essential for the traffic flow prediction, since the sequential data is more regular. Although the weight is low, the periodic representation still helps to improve the performance of traffic flow prediction qualitatively and quantitatively. For example, we can decrease the RMSE of SRNN by 3.2% after incorporating the periodic representation.

E. Extension to Citywide Passenger Demand Prediction

Our ATFM is a general model for urban mobility modeling. Apart from the traffic flow prediction, it can also be applied to other related traffic tasks, such as citywide passenger demand prediction. In this subsection, we extend the proposed method to forecast the passenger pickup/dropoff demands at the next time interval (half an hour) with historical mobility trips.

We conduct experiments with taxi trips in New York City. Since most taxi transactions were made in the Manhattan borough, we choose it as the studied area and divide it into a $h \times w$ grid map. We collect 132 million taxicab trip records

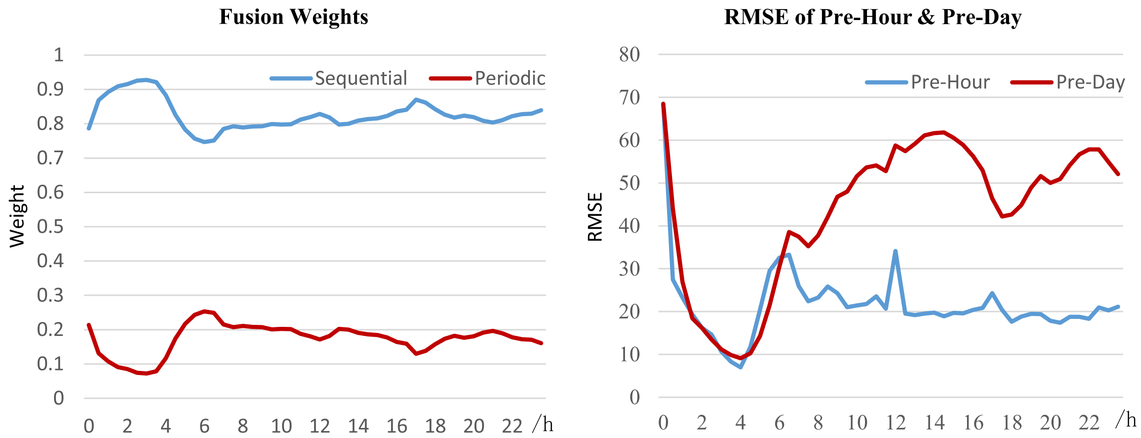


Fig. 13. **Left:** The average fusion weights of two types of temporal representation on the testing set of TaxiBJ dataset. **Right:** The RMSE of traffic flow between two consecutive time intervals (denoted as "Pre-Hour") and the RMSE of traffic flow between two adjacent days at the same time interval (denoted as "Pre-Day"). We can find that the weights of sequential representation are greater than that of the periodic representation, which indicates that the sequential trend is more essential for traffic flow prediction.

TABLE VII

EFFECTIVENESS OF DIFFERENT SPATIAL RESOLUTIONS FOR SHORT-TERM DEMAND PREDICTION

Spatial Resolution	RMSE	MAE
10 × 3	32.28	19.24
12 × 4	23.46	13.63
15 × 5	17.29	9.91
20 × 7	11.80	6.50
30 × 10	7.38	3.96

TABLE VIII

QUANTITATIVE COMPARISONS FOR CITYWIDE PASSENGER SHORT-TERM DEMAND PREDICTION

Spatial Resolution	RMSE	MAE
10 × 3	32.28	19.24
12 × 4	23.46	13.63
15 × 5	17.29	9.91
20 × 7	11.80	6.50
30 × 10	7.38	3.96

TABLE IX

QUANTITATIVE COMPARISONS (RMSE) FOR CITYWIDE PASSENGER LONG-TERM DEMAND PREDICTION. ALL COMPARED METHODS HAVE BEEN FINETUNED FOR LONG-TERM PREDICTION. EACH TIME INTERVAL IS HALF AN HOUR (0.5 H) IN THIS DATASET

Method	**Time Interval**			
	1 (0.5 h)	2 (1.0 h)	3 (1.5 h)	4 (2.0 h)
ST-ResNet	18.11	22.87	28.21	34.51
VPN	19.74	22.63	25.36	28.19
PredNet	18.44	22.44	25.97	29.34
PredRNN	17.75	21.62	25.41	29.31
SPN-LONG (Ours)	**17.41**	**20.08**	**22.19**	**24.45**

during 2014 from New York City Taxi and Limousine Commission (NYCTLC[5]). Each record contains the timestamp and the geo-coordinates of pickup and dropoff locations. For each region, we measure the passenger pickup/dropoff demands every half an hour, thus the dimensionality of passenger demand maps is $2 \times h \times w$. We collect external meteorological factors (e.g., temperature, wind speed and weather conditions) from Wunderground and the holidays are also marked. Finally, we train our model with the historical demand of the first 337 days and conduct evaluation with the data in the last four weeks.

We first explore the effectiveness of different spatial resolutions ($h \times w$). As shown in Table VII, the RMSE and MAE of our method gradually decrease as the resolution increases. However, this performance improvement may come from the corresponding reduction in demand as the unit area becomes smaller. Moreover, too high resolution may result in over-divided regions (e.g., a stadium may be divided into multi regions) and it is unnecessary to forecast taxi demand in a very small region. In the previous work [53], Didi Chuxing, a famous taxi requesting company in China, predicted taxi demand in each $0.7km \times 0.7km$ region. Following this setting, we divide the Manhattan borough into a 15×7 grid map and each grid represents a geographical region with a size of about $0.75km \times 0.75km$.

We then compare our method with HA and five deep learning based methods. As shown in Table VIII, the baseline method HA obtains a RMSE of 39.02 and a MAE of 20.24, which is impractical in the taxi industry. By contrast, our method dramatically decreases the RMSE to 17.29 and outperforms other compared methods for short-term prediction. Moreover, we adapt and retrain these deep learning based methods to forecast the long-term demand and summarize their RMSE in Table IX. It can be observed that our SPN-LONG model achieves the best performance at every time interval. In particular, our method has a performance improvement of 16.58% compared with PredRNN at the fourth time interval. These experiments show that the proposed method is also effective for passenger demand prediction.

VII. CONCLUSION

In this work, we utilize massive human trajectory data collected from mobility digital devices to study the traffic flow prediction problem. Its key challenge lies in how to adaptively integrate various factors that affect the flow changes, such

[5] https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page

as sequential trends, periodic laws and spatial dependencies. To address these issues, we propose a novel Attentive Traffic Flow Machine (ATFM), which explicitly learns dynamic spatial-temporal representations from historical traffic flow maps with an attention mechanism. Based on the proposed ATFM, we develop a unified framework to adaptively merge the sequential and periodic representations with the aid of a temporally-varying fusion module for citywide traffic flow prediction. By conducting extensive experiments on two public benchmarks, we have verified the effectiveness of our method for traffic flow prediction. Moreover, to verify the generalization of ATFM, we apply the customized framework to forecast the passenger pickup/dropoff demand and it can also achieve practical performance on this traffic prediction task.

However, there is still much room for improvement. First, it may be suboptimal to divide the studied cities into regular grid maps. In future work, we would divide them into traffic analysis zones with irregular shapes on the basis of the functionalities of regions. We would model such traffic systems as graphs and adapt Graph Convolutional Network (GCN [54], [55]) to learn spatial-temporal features. Second, the functionality information of zones has not been fully explored in most previous works. Intuitively, the zones with the same functionalities usually have similar traffic flow patterns. For instance, most residential regions have high outflow during morning rush hours and have high inflow during evening rush hours. Base on this consideration, we plan to incorporate the prior knowledge of functionality information of zones (e.g., the Point of Interest (POI) data, land-use data and socio-demographic data) into GCN to further improve the prediction accuracy.

References

[1] S. Xingjian, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-C. Woo, “Convolutional LSTM network: A machine learning approach for precipitation nowcasting,” in *Proc. NIPS*, 2015, pp. 802–810.

[2] Y. Zheng, L. Capra, O. Wolfson, and H. Yang, “Urban computing: Concepts, methodologies, and applications,” *ACM Trans. Intell. Syst. Technol.*, vol. 5, no. 3, p. 38, 2014.

[3] J. Zhang, F.-Y. Wang, K. Wang, W.-H. Lin, X. Xu, and C. Chen, “Data-driven intelligent transportation systems: A survey,” *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 4, pp. 1624–1639, Dec. 2011.

[4] W. Huang, G. Song, H. Hong, and K. Xie, “Deep architecture for traffic flow prediction: Deep belief networks with multitask learning,” *IEEE Trans. Intell. Transport. Syst.*, vol. 15, no. 5, pp. 2191–2201, Oct. 2014.

[5] Y. Lv, Y. Duan, W. Kang, Z. Li, and F.-Y. Wang, “Traffic flow prediction with big data: A deep learning approach,” *IEEE Trans. Intell. Transport. Syst.*, vol. 16, no. 2, pp. 865–873, Apr. 2015.

[6] N. G. Polson and V. O. Sokolov, “Deep learning for short-term traffic flow prediction,” *Transp. Res. C, Emerg. Technol.*, vol. 79, pp. 1–17, Jun. 2017.

[7] J. Zhang, Y. Zheng, and D. Qi, “Deep spatio-temporal residual networks for citywide crowd flows prediction,” in *Proc. AAAI*, 2017, pp. 1655–1661.

[8] S. Shekhar and B. M. Williams, “Adaptive seasonal time series models for forecasting short-term traffic flow,” *Transp. Res. Rec., J. Transp. Res. Board*, vol. 2024, no. 1, pp. 116–125, Jan. 2007.

[9] J. Guo, W. Huang, and B. M. Williams, “Adaptive Kalman filter approach for stochastic short-term traffic flow rate prediction and uncertainty quantification,” *Transp. Res. C, Emerg. Technol.*, vol. 43, pp. 50–64, Jun. 2014.

[10] J. Zheng and L. M. Ni, “Time-dependent trajectory regression on road networks via multi-task learning,” in *Proc. AAAI*, 2013, pp. 1048–1055.

[11] D. Deng, C. Shahabi, U. Demiryurek, L. Zhu, R. Yu, and Y. Liu, “Latent space model for road networks to predict time-varying traffic,” in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 1525–1534.

[12] J. Zhang, Y. Zheng, D. Qi, R. Li, and X. Yi, “DNN-based prediction model for spatio-temporal data,” in *Proc. SIGSPATIAL*, 2016, p. 92.

[13] Z. Xu, Y. Wang, M. Long, and J. Wang, “PredCNN: Predictive learning with cascade convolutions,” in *Proc. 27th Int. Joint Conf. Artif. Intell.*, Jul. 2018, pp. 2940–2947.

[14] J. Zhang, Y. Zheng, J. Sun, and D. Qi, “Flow prediction in spatio-temporal networks based on multitask deep learning,” *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 3, pp. 468–478, Mar. 2019.

[15] Z. Zhao, W. Chen, X. Wu, P. C. Y. Chen, and J. Liu, “LSTM network: A deep learning approach for short-term traffic forecast,” *IET Intell. Transport. Syst.*, vol. 11, no. 2, pp. 68–75, Mar. 2017.

[16] J. Zhang, Y. Zheng, D. Qi, R. Li, X. Yi, and T. Li, “Predicting citywide crowd flows using deep spatio-temporal residual networks,” *Artif. Intell.*, vol. 259, pp. 147–166, Jun. 2018.

[17] S. Sharma, R. Kiros, and R. Salakhutdinov, “Action recognition using visual attention,” 2015, *arXiv:1511.04119*. [Online]. Available: http://arxiv.org/abs/1511.04119

[18] J. Lu, C. Xiong, D. Parikh, and R. Socher, “Knowing when to look: Adaptive attention via a visual sentinel for image captioning,” 2016, *arXiv:1612.01887*. [Online]. Available: http://arxiv.org/abs/1612.01887

[19] L. Liu, H. Wang, G. Li, W. Ouyang, and L. Lin, “Crowd counting using deep recurrent spatial-aware network,” in *Proc. Int. Joint Conf. Artif. Intell. Org.*. AAAI Press, 2018, pp. 849–855.

[20] C.-P. Tay, S. Roy, and K.-H. Yap, “AANet: Attribute attention network for person re-identifications,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 7134–7143.

[21] L. Liu, R. Zhang, J. Peng, G. Li, B. Du, and L. Lin, “Attentive crowd flow machines,” in *Proc. ACM Multimedia Conf.*, 2018, pp. 1553–1561.

[22] B. M. Williams, P. K. Durvasula, and D. E. Brown, “Urban freeway traffic flow prediction: Application of seasonal autoregressive integrated moving average and exponential smoothing models,” *Transp. Res. Rec., J. Transp. Res. Board*, vol. 1644, no. 1, pp. 132–141, Jan. 1998.

[23] M. Castro-Neto, Y.-S. Jeong, M.-K. Jeong, and L. D. Han, “Online-SVR for short-term traffic flow prediction under typical and atypical traffic conditions,” *Expert Syst. Appl.*, vol. 36, no. 3, pp. 6164–6173, Apr. 2009.

[24] X. Li *et al.*, “Prediction of urban human mobility using large-scale taxi traces and its applications,” *Frontiers Comput. Sci.*, vol. 6, no. 1, pp. 111–121, 2012.

[25] M. Lippi, M. Bertini, and P. Frasconi, “Short-term traffic flow forecasting: An experimental comparison of time-series analysis and supervised learning,” *IEEE Trans. Intell. Transport. Syst.*, vol. 14, no. 2, pp. 871–882, Jun. 2013.

[26] S. Johansen, “Estimation and hypothesis testing of cointegration vectors in Gaussian vector autoregressive models,” *Econometrica, J. Econ. Soc.*, vol. 59, no. 6, pp. 1551–1580, Nov. 1991.

[27] G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time Series Analysis: Forecasting and Control*. Hoboken, NJ, USA: Wiley, 2015.

[28] Y. Duan, Y. Lv, Y.-L. Liu, and F.-Y. Wang, “An efficient realization of deep learning for traffic data imputation,” *Transp. Res. C, Emerg. Technol.*, vol. 72, pp. 168–181, Nov. 2016.

[29] Z. Chen, J. Zhou, and X. Wang, “Visual analytics of movement pattern based on time-spatial data: A neural net approach,” 2017, *arXiv:1707.02554*. [Online]. Available: http://arxiv.org/abs/1707.02554

[30] M. Fouladgar, M. Parchami, R. Elmasri, and A. Ghaderi, “Scalable deep traffic flow neural networks for urban traffic congestion prediction,” 2017, *arXiv:1703.01006*. [Online]. Available: http://arxiv.org/abs/1703.01006

[31] J. Ke, H. Zheng, H. Yang, and X. M. Chen, “Short-term forecasting of passenger demand under on-demand ride services: A spatio-temporal deep learning approach,” *Transp. Res. C, Emerg. Technol.*, vol. 85, pp. 591–608, Dec. 2017.

[32] H. Wei, G. Zheng, H. Yao, and Z. Li, “IntelliLight: A reinforcement learning approach for intelligent traffic light control,” in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2018, pp. 2496–2505.

[33] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[34] X. Geng *et al.*, “Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting,” in *Proc. AAAI*, 2019, pp. 3656–3663.

[35] L. Wang, X. Geng, X. Ma, F. Liu, and Q. Yang, “Cross-city transfer learning for deep spatio-temporal prediction,” 2018, *arXiv:1802.00386*. [Online]. Available: http://arxiv.org/abs/1802.00386

[36] H. Yao, Y. Liu, Y. Wei, X. Tang, and Z. Li, “Learning from multiple cities: A meta-learning approach for spatial-temporal prediction,” 2019, *arXiv:1901.08518*. [Online]. Available: http://arxiv.org/abs/1901.08518

[37] M.-T. Luong, H. Pham, and C. D. Manning, “Effective approaches to attention-based neural machine translation,” 2015, *arXiv:1508.04025*. [Online]. Available: http://arxiv.org/abs/1508.04025

[38] A. Graves, A.-R. Mohamed, and G. Hinton, “Speech recognition with deep recurrent neural networks,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, May 2013, pp. 6645–6649.

[39] N. Kalchbrenner *et al.*, “Video pixel networks,” in *Proc. ICML*, 2017, pp. 1771–1779.

[40] W. Lotter, G. Kreiman, and D. Cox, “Deep predictive coding networks for video prediction and unsupervised learning,” in *Proc. ICLR*, 2017, pp. 1–18.

[41] Y. Wang, M. Long, J. Wang, Z. Gao, and S. Y. Philip, “PredRNN: Recurrent neural networks for predictive learning using spatiotemporal LSTMs,” in *Proc. NIPS*, 2017, pp. 879–888.

[42] Y. Tian and L. Pan, “Predicting short-term traffic flow by long short-term memory recurrent neural network,” in *Proc. IEEE Int. Conf. Smart City/SocialCom/SustainCom (SmartCity)*, Dec. 2015, pp. 153–158.

[43] R. Fu, Z. Zhang, and L. Li, “Using LSTM and GRU neural network methods for traffic flow prediction,” in *Proc. 31st Youth Acad. Annu. Conf. Chin. Assoc. Autom. (YAC)*, Nov. 2016, pp. 324–328.

[44] J. Mackenzie, J. F. Roddick, and R. Zito, “An evaluation of HTM and LSTM for short-term arterial traffic flow prediction,” *IEEE Trans. Intell. Transport. Syst.*, vol. 20, no. 5, pp. 1847–1857, May 2019.

[45] G. Li, Y. Gan, H. Wu, N. Xiao, and L. Lin, “Cross-modal attentional context learning for RGB-D object detection,” *IEEE Trans. Image Process.*, vol. 28, no. 4, pp. 1591–1601, Apr. 2019.

[46] Y. Shi, G. Li, Q. Cao, K. Wang, and L. Lin, “Face hallucination by attentive sequence optimization with reinforcement learning,” *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, May 7, 2019, doi: 10.1109/TPAMI.2019.2915301.

[47] H. Xu and K. Saenko, “Ask, attend and answer: Exploring question-guided spatial attention for visual question answering,” in *Proc. Eur. Conf. Comput. Vis.* Springer, 2016, pp. 451–466.

[48] L. Liu, Z. Qiu, G. Li, Q. Wang, W. Ouyang, and L. Lin, “Contextualized spatial–temporal network for taxi origin-destination demand prediction,” *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 10, pp. 3875–3887, Oct. 2019.

[49] D. Harris and S. Harris, *Digital Design and Computer Architecture*. San Mateo, CA, USA: Morgan Kaufmann, 2010.

[50] A. Paszke *et al.*, “Automatic differentiation in PyTorch,” in *Proc. NIPS Workshop*, 2017.

[51] X. Glorot and Y. Bengio, “Understanding the difficulty of training deep feedforward neural networks,” in *Proc. AISTATS*, 2010, pp. 249–256.

[52] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980

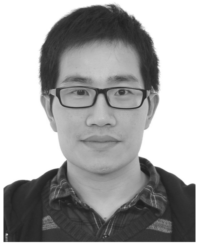
[53] H. Yao *et al.*, “Deep multi-view spatial-temporal network for taxi demand prediction,” 2018, *arXiv:1802.08714*. [Online]. Available: http://arxiv.org/abs/1802.08714

[54] D. K. Duvenaud *et al.*, “Convolutional networks on graphs for learning molecular fingerprints,” in *Proc. NIPS*, 2015, pp. 2224–2232.

[55] J. Chen, L. Liu, H. Wu, J. Zhen, G. Li, and L. Lin, “Physical-virtual collaboration graph network for station-level metro ridership prediction,” 2020, *arXiv:2001.04889*. [Online]. Available: http://arxiv.org/abs/2001.04889

Lingbo Liu (Graduate Student Member, IEEE) received the B.E. degree from the School of Software, Sun Yat-sen University, Guangzhou, China, in 2015, where he is currently pursuing the Ph.D. degree in computer science with the School of Data and Computer Science. From March 2018 to May 2019, he was a Research Assistant with The University of Sydney, Australia. He has authorized or coauthorized more than ten papers in top-tier academic journals and conferences. His current research interests include machine learning and intelligent transportation systems.

Jiajie Zhen received the B.E. degree from the School of Mathematics, Sun Yat-sen University, Guangzhou, China, in 2018, where he is currently pursuing the master’s degree in computer science with the School of Data and Computer Science. His current research interests include computer vision and intelligent transportation systems.



Guanbin Li (Member, IEEE) received the Ph.D. degree from the University of Hong Kong in 2016. He is currently an Associate Professor with the School of Data and Computer Science, Sun Yat-sen University. He has authorized or coauthorized more than 40 papers in top-tier academic journals and conferences. His current research interests include computer vision, image processing, and deep learning. He was a recipient of the ICCV 2019 Best Paper Nomination Award. He serves as the Area Chair of the Conference of VISAPP. He has been serving as a reviewer for numerous academic journals and conferences, such as the IEEE Transactions on Pattern Analysis and Machine Intelligence, the IEEE Transactions on Image Processing, the IEEE Transactions on Multimedia, TC, CVPR, AAAI, and IJCAI.

Geng Zhan received the B.E. degree from the Faculty of Electronic Information and Electrical Engineering, Dalian University of Technology, in 2017. He is currently pursuing the Master of Philosophy degree with the School of Electrical and Information Engineering, The University of Sydney. His current research interests include computer vision and machine learning.

Zhaocheng He (Member, IEEE) received the B.S. and Ph.D. degrees from Sun Yat-sen University, Guangzhou, China, in 2000 and 2005, respectively. He is currently a Professor with the Guangdong Provincial Key Laboratory of Intelligent Transportation Systems (ITS), and the Research Center of ITS, Sun Yat-sen University. His research interests include traffic flow dynamics and simulation, traffic data mining, and intelligent transportation systems.

Bowen Du (Member, IEEE) received the Ph.D. degree in computer science and engineering from Beihang University, Beijing, China, in 2013. He is currently a Professor with the State Key Laboratory of Software Development Environment, Beihang University. His research interests include smart city technology, multisource data fusion, and traffic data mining.

Liang Lin (Senior Member, IEEE) was a Post-Doctoral Fellow with the University of California, Los Angeles, from 2008 to 2010. From 2014 to 2015, he was with The Hong Kong Polytechnic University and The Chinese University of Hong Kong, as a Senior Visiting Scholar. He is currently a Full Professor with Sun Yat-sen University. He is also the Excellent Young Scientist of the National Natural Science Foundation of China. He also leads the SenseTime research and development teams to develop cutting-edge and deliverable solutions on computer vision, data analysis and mining, and intelligent robotic systems. He has authored or coauthored more than 100 papers in top-tier academic journals and conferences. He is a fellow of IET. He was a recipient of the Best Paper Runners-Up Award at the ACM NPAR 2010, the Google Faculty Award in 2012, the Best Paper Diamond Award at the IEEE ICME 2017, and the Hong Kong Scholars Award in 2014. He served as the Area/Session Chair for numerous conferences, including ICME, ACCV, and ICMR. He has been serving as an Associate Editor for the IEEE Transactions on Human–Machine Systems, *Visual Computer*, and *Neurocomputing*.