Neural Networks 161 (2023) 25–38

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

Interpretable local flow attention for multi-step traffic flow prediction

Xu Huang [a], Bowen Zhang [b], Shanshan Feng [a], Yunming Ye [a,c,*], Xutao Li [a,c]

[a] School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen, China
[b] College of Big Data and Internet, Shenzhen Technology University, Shenzhen, China
[c] Peng Cheng Laboratory, Shenzhen, China

ARTICLE INFO

Article history:
Received 5 September 2022
Received in revised form 9 November 2022
Accepted 19 January 2023
Available online 28 January 2023

Keywords:
Traffic flow prediction
Attention mechanism
Neural networks
Explainable artificial intelligence

ABSTRACT

Traffic flow prediction (TFP) has attracted increasing attention with the development of smart city. In the past few years, neural network-based methods have shown impressive performance for TFP. However, most of previous studies fail to explicitly and effectively model the relationship between inflows and outflows. Consequently, these methods are usually uninterpretable and inaccurate. In this paper, we propose an interpretable local flow attention (LFA) mechanism for TFP, which yields three advantages. (1) LFA is flow-aware. Different from existing works, which blend inflows and outflows in the channel dimension, we explicitly exploit the correlations between flows with a novel attention mechanism. (2) LFA is interpretable. It is formulated by the truisms of traffic flow, and the learned attention weights can well explain the flow correlations. (3) LFA is efficient. Instead of using global spatial attention as in previous studies, LFA leverages the local mode. The attention query is only performed on the local related regions. This not only reduces computational cost but also avoids false attention. Based on LFA, we further develop a novel spatiotemporal cell, named LFA-ConvLSTM (LFA-based convolutional long short-term memory), to capture the complex dynamics in traffic data. Specifically, LFA-ConvLSTM consists of three parts. (1) A ConvLSTM module is utilized to learn flow-specific features. (2) An LFA module accounts for modeling the correlations between flows. (3) A feature aggregation module fuses the above two to obtain a comprehensive feature. Extensive experiments on two real-world datasets show that our method achieves a better prediction performance. We improve the RMSE metric by 3.2%–4.6%, and the MAPE metric by 6.2%–6.7%. Our LFA-ConvLSTM is also almost 32% faster than global self-attention ConvLSTM in terms of prediction time. Furthermore, we also present some visual results to analyze the learned flow correlations.

1. Introduction

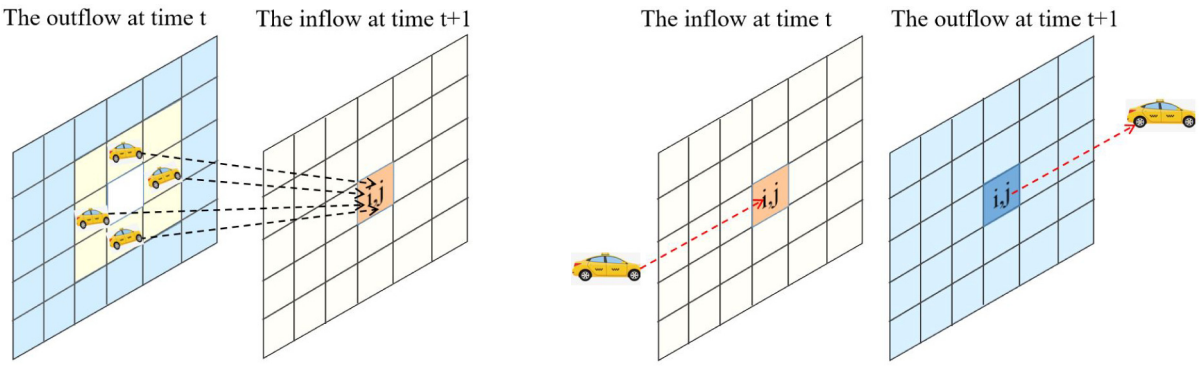
Traffic flow prediction is an important task in intelligent transportation systems, which aims to predict the inflow and outflow of crowds in each part of a city. It can help citizens make a better route plan, and assist government agencies in reducing traffic congestion and accidents. With the popularity of ride-hailing applications, such as Uber and Didi, it has become easier to collect large-scale traffic data. In this context, data-driven traffic flow prediction has drawn considerable attention.

Recently, neural network-based methods have dominated the study of traffic flow prediction (Ali, Zhu, & Zakarya, 2022; An et al., 2021; Huang et al., 2022; Ren, Zhao, Luo, Ma, & Duan, 2020; Tedjopurnomo, Bao, Zheng, Choudhury, & Qin, 2020; Yao, Tang, Wei, Zheng, & Li, 2019; Zhang, Zheng, & Qi, 2017). They usually achieve significant performance improvements over conventional statistical methods, such as autoregressive integrated moving average (ARIMA) (Shekhar & Williams, 2007; Williams & Hoel, 2003), kalman filtering (Ojeda, Kibangou, & De Wit, 2013), etc. In terms of various neural networks, existing methods can be grouped into three types.[1] (1) *Methods based on convolutional neural networks (CNNs)* (Ren et al., 2020; Zhang et al., 2017). These works employ convolution operation to capture spatial information of grid map-based traffic data. However, they cannot effectively model the temporal dynamics, thus showing a weakness in multi-step prediction. (2) *Methods based on recurrent neural networks (RNNs)* (Jiang et al., 2018; Ma, Dai, & Zhou, 2021; Wang, Su, & Ding, 2020). This line takes advantage of RNNs to capture temporal dependencies in historical sequences. Nevertheless, they do not exploit the spatial correlations efficiently. (3) *Methods combining CNNs and RNNs* (Du et al., 2019; Shi et al., 2015; Yao et al., 2019; Zheng, Lin, Feng, & Chen, 2020). Inspired

* Corresponding author.
E-mail addresses: huangxuu@outlook.com (X. Huang), zhang_bo_wen@foxmail.com (B. Zhang), victor_fengss@foxmail.com (S. Feng), yym@hit.edu.cn (Y. Ye), lixutao@hit.edu.cn (X. Li).

[1] In this paper, we focus on grid-based traffic data rather than graph-based one.

https://doi.org/10.1016/j.neunet.2023.01.023
0893-6080/



(a) The inflow of a region (i, j) at time $t + 1$ is affected by the outflows of its neighborhood at time t.

(b) The outflow of a region (i, j) at time $t + 1$ depends on its inflow at time t.

Fig. 1. The relationship between inflows and outflows.

by the previous two methods, some studies naturally combine CNNs and RNNs to capture complex spatiotemporal dynamics in traffic data. Here, a representative benchmark is ConvLSTM (convolutional long short-term memory), which integrates the convolution operation into LSTM (Lin, Li, Zheng, Cheng, & Yuan, 2020; Shi et al., 2015).

Based on the above methods, neural attention mechanism is incorporated to make a more accurate prediction (Lin et al., 2020; Shi, Qi, Shen, Wu, & Yin, 2020; Yao et al., 2019; Zheng et al., 2020; Zhou, Shen, Zhu, & Huang, 2018). This mechanism has achieved great success in natural language processing and computer vision (Devlin, Chang, Lee, & Toutanova, 2018; Dosovitskiy et al., 2020; Guo, Gu, Qiao, & Bi, 2021; Peng, Yang, Liu, & Lü, 2021; Vaswani et al., 2017; Xu et al., 2015; Yang et al., 2016), and also attracted increasing attention in traffic flow prediction. The idea behind it is to calculate the correlations across temporal domains or spatial domains, which can help learn better spatialtemporal dynamics.

Despite the effectiveness of existing studies, there are still some issues for traffic flow prediction. (1) The relationship between inflows and outflows is not explicitly modeled and well interpreted. Specifically, as shown in Fig. 1, the inflow of a region is affected by the outflows of its neighborhood, and the outflow of a region depends on its inflow. Hence, it is important to exploit these correlations. However, most of the existing methods tackle it in a black-box manner. They concatenate the two flows in the channel dimension, and then extract the correlations by convolution operation. This hides the explicit relationship between inflows and outflows, which leads to inadequate utilization of flow correlations and makes the model less interpretable. (2) Existing spatial attention mechanisms for traffic flow prediction are computationally expensive and may introduce long-distance noise. Specifically, spatial dependencies play an important role in prediction. Nevertheless, previous spatial attention mechanisms employ global mode, which calculates the correlations between a region and the whole map. This results in high computational complexity. Moreover, global attention assigns weight to each grid, even if there is no interaction. Consequently, it may introduce false attention and degrade the performance.

To address the above issues, we propose a novel interpretable local flow attention method for multi-step traffic flow prediction. Specifically, we present a two-branch framework to predict inflows and outflows respectively, and further design a local flow attention (LFA) mechanism to explicitly model the flow correlations. First of all, we underline that LFA is formulated by the truisms of traffic flow, i.e., *the inflow of a region is affected by the outflows of its neighborhood, and the outflow of a region depends on its inflow.* Therefore, our method can be understood as self-explanatory, where one can clearly observe the flow correlations. On the implementation side, LFA has two variants, which correspond to inflows and outflows, respectively. Both of them calculate the flow correlations by attention queries, but the details are slightly different according to the truisms. Moreover, the learned attention weights can measure the degree of flow interactions, thus also providing interpretability. Secondly, LFA leverages local spatial attention instead of the global mode. Specifically, motivated by that closer regions are more prone to frequent traffic flows, LFA only considers the flows in adjacent regions. Notice that in the global mode, the attention query is performed on the whole map, even though there is no interaction between distant regions. Hence, the proposed LFA helps reduce the computational cost and focus on relevant regions more effectively.

Based on LFA, we further develop a new prediction cell, named LFA-ConvLSTM, to capture complex spatiotemporal dynamics. It equips the classical ConvLSTM cell with our local flow attention mechanism. Concretely, LFA-ConvLSTM consists of three parts. (1) A ConvLSTM module is utilized to learn flow-specific features. (2) An LFA module accounts for modeling the correlations between flows. (3) A feature aggregation module fuses the above two to obtain a comprehensive feature.

Overall, the main contributions of this paper are summarized as follows:

- We propose a novel framework for multi-step traffic flow prediction, which is formulated by the truisms of traffic flow. In our method, the relationship between inflows and outflows is explicitly modeled and can be well explained.
- We present a novel local flow attention (LFA) mechanism, which is more interpretable and efficient. By local attention querying, LFA can measure the degree of flow correlations. Here, LFA further has two variants complying with the truisms of inflows and outflows, respectively. Moreover, LFA exploits the local attributes of traffic flow to avoid expensive global computations and false attention.
- We develop a new spatiotemporal cell LFA-ConvLSTM, which is based on LFA. It can not only capture the dynamics of a specific flow, but also learn the correlations between flows.

Further, a feature aggregation part in this cell fuses the above two features to obtain a comprehensive one.

- We conduct extensive experiments to evaluate the effectiveness of our method, and the results show that we achieve a better prediction performance. Furthermore, we also present some visual results to analyze the learned flow correlations.

2. Related work

2.1. Traffic flow prediction

Existing studies for traffic flow prediction can be divided into classical statistical methods and neural network-based methods.

Classical statistical methods have been developed for decades, including autoregressive integrated moving average (ARIMA) (Ahmed & Cook, 1979; Van Der Voort, Dougherty, & Watson, 1996; Williams & Hoel, 2003), kalman filtering (Ojeda et al., 2013), support vector regression (SVR) (Castro-Neto, Jeong, Jeong, & Han, 2009), etc. These earlier models focus more on the temporal information in traffic data but ignore the spatial dependencies. Later, some studies began to exploit spatial information to improve prediction (Deng et al., 2016; Tong et al., 2017). However, all of them do not well capture the complex non-linear spatiotemporal dynamics in traffic flow. As a result, they can hardly achieve a good performance in practice.

Recently, a variety of neural network-based methods have shown impressive performance for traffic flow prediction, which can be further grouped into three types. (1) The first line of these studies is based on convolutional neural networks (CNNs). They aim to capture spatial dependencies by treating traffic flow readings as an image (Ren et al., 2020; Zhang et al., 2017). For example, Zhang et al. (2017) employ multi-branch residual convolutional units to model the closeness, period, and trend properties of crowd traffic. Nevertheless, these methods do not effectively capture temporal dynamics, and show a weakness in multi-step prediction. (2) The second line is the recurrent neural networks (RNNs) based methods (Jiang et al., 2018; Ma et al., 2021; Wang, Su, & Ding, 2020). RNNs are born for time series data, thus naturally suitable for traffic prediction. To obtain an effective long-term prediction, Wang, Su, and Ding (2020) leverage a long short-term memory encoder–decoder structure and a calibration layer. However, these methods fail to consider spatial dependencies. Consequently, the performance is less than satisfactory. (3) The third line learns from the previous two by combining CNNs and RNNs (Du et al., 2019; Shi et al., 2015; Yao et al., 2019; Zheng et al., 2020). A classical benchmark is convolutional recurrent neural networks (ConvLSTM) (Shi et al., 2015), which integrates convolution operation into LSTM to capture spatial dependencies and temporal dynamics simultaneously. To learn dynamic similarities over time and space, Yao et al. (2019) propose a novel Spatial–Temporal Dynamic Network (STDN) with a flow gating mechanism and a periodically shifted attention mechanism. Furthermore, Zheng et al. (2020) develop a hybrid and multiple-layer architecture with an attention-based Conv-LSTM module to extract the spatial and short-term temporal features. Recently, methods based on graph neural networks (GNNs) have attracted increasing attention for graph-based traffic data (Jiang & Luo, 2022; Roy, Roy, Ali, Amin, & Rahman, 2021; Zhang et al., 2021; Zhou, Yang, et al., 2020). In this work, we focus on grid-based data, where GNNs are not that suitable.

However, the above methods perform in a black-box manner when they tackle inflows and outflows. This not only makes the models difficult to understand, but also results in inadequate utilization of flow interactions. Therefore, we propose to explicitly and effectively model the correlations between inflows and outflows, which aims at an interpretable and accurate prediction.

2.2. Neural attention mechanism

Neural attention mechanism has achieved great success in the machine learning community, including natural language processing (Bahdanau, Cho, & Bengio, 2014; Liu, Guan, Giunchiglia, Liang, & Feng, 2021; Vaswani et al., 2017; Yang et al., 2016; Zeng, Wu, Yin, Jiang, & Li, 2021; Zhang et al., 2020; Zhou, Pan, Bai, Luo, & Wu, 2021) and computer vision (Hao et al., 2020; Tian et al., 2020; Wang, Jiang, et al., 2017; Woo, Park, Lee, & Kweon, 2018; Xu et al., 2015; Yang, Zhang, Zhou, & Liu, 2021; Yang, Zhou, Chen, & Ngiam, 2021). The basic idea is to guide the model to focus on the important parts through attention weights. In terms of different attention forms for various neural networks, we review three classical works. (1) Temporal attention for recurrent neural networks (RNNs). Bahdanau et al. (2014) are known as the first ones to apply the attention mechanism to machine translation tasks. Later, temporal attention has been widely applied in time series analysis (Fan et al., 2019; Sinha, Dong, Cheung, & Ruths, 2018; Tran, Iosifidis, Kanniainen, & Gabbouj, 2018). The key of temporal attention lies in exploring the correlations between different time steps and their contributions to prediction. (2) Spatial attention for convolutional neural networks (CNNs). Different from temporal attention, spatial attention generates attention mask across spatial domains, which aims to select important spatial regions (Hu, Shen, Albanie, Sun, & Vedaldi, 2018; Ramachandran et al., 2019; Wang, Girshick, Gupta, & He, 2018; Zhao, Jia, & Koltun, 2020). (3) Channel attention for CNNs. Advanced CNNs usually employ numerous channels to extract rich features. Channel attention targets at selecting important channels for prediction, and has become an important component to improve performance (Hu, Shen, & Sun, 2018; Wang, Wu, et al., 2020; Zhang et al., 2018).

Recently, some studies also apply attention mechanism to traffic flow prediction (Fang et al., 2021; Lin et al., 2020; Shi et al., 2020; Yao et al., 2019; Zheng et al., 2020; Zhou, Li, et al., 2020; Zhou et al., 2018). For example, Zhou et al. (2018) propose to incorporate representative citywide demand tensors into prediction by attention. Shi et al. (2020) develop an attention-based periodic temporal neural network, which leverages an encoder attention mechanism to capture both the spatial and periodical dependencies. Inspired by the transformer model in natural language processing (Devlin et al., 2018; Vaswani et al., 2017) and computer vision (Dosovitskiy et al., 2020), Zhou, Li, et al. (2020) adopt a multiple-output strategy without RNN units, i.e., a pure attention model for traffic prediction.

However, these attention mechanisms fail to explicitly and effectively capture the relationship between inflows and outflows, which is exceedingly important for prediction. Furthermore, they usually employ global spatial attention to capture spatial dependencies. This lead to a high computational complexity. Hence, in this paper, we propose a novel attention mechanism, which is flow-aware, interpretable, and efficient.

3. Task definition and framework overview

3.1. Task definition

In this paper, we focus on the multi-step traffic flow prediction of a given region. Formally, we define the task as follows:

Multi-step Traffic Flow Prediction (Ms-TFP): G denotes a target region, which is divided into an $m \times n$ grid map. Given a historical traffic sequence in G with length L: $X_{1:L} = \{X_1, X_2, \ldots, X_L\}$, Ms-TFP aims to predict the most likely length-K sequence in the future $\widehat{X}_{L+1:L+K} = \{\widehat{X}_{L+1}, \widehat{X}_{L+2}, \ldots, \widehat{X}_{L+K}\}$:

$$\widehat{X}_{L+1:L+K} = \arg\max_{X_{L+1:L+K}} p(X_{L+1:L+K} | X_{1:L}). \tag{1}$$

Here, $X_t = [X_t^{in}, X_t^{out}] \in \mathbb{R}^{m \times n \times 2}$ denotes the traffic flow at time t, which stacks the inflow and outflow together.

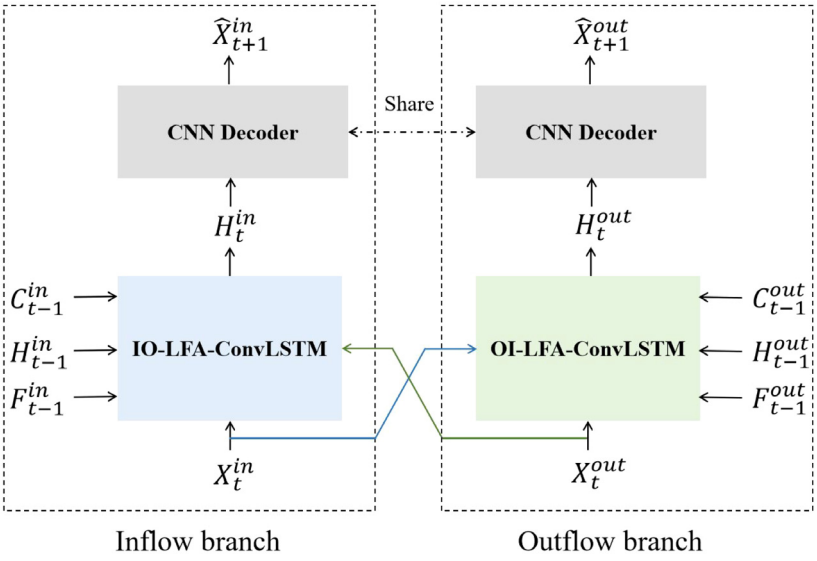


Fig. 2. The overview of our prediction framework. It consists of an inflow branch and an outflow branch, which are utilized to predict inflows and outflows, respectively. LFA-ConvLSTM module is the key for capturing complex spatiotemporal dynamics in traffic data. Three states, i.e., C, H, and F, account for memorizing sequential information.

3.2. Framework overview

As shown in Fig. 1, the correlations between inflows and outflows are crucial for Ms-TFP. Therefore, to explicitly and effectively model them, we present a new framework, which is shown in Fig. 2. The proposed method consists of an inflow branch and an outflow branch, which account for predicting inflows and outflows, respectively. In each branch, a local flow attention-based ConvLSTM (LFA-ConvLSTM) is developed to capture complex dynamics in traffic data. Specifically, at time t, the flows (X^{in}_t and X^{out}_t) are sent to LFA-ConvLSTM to obtain spatiotemporal representations (H^{in}_t and H^{out}_t). Then a shared CNN decoder module is utilized to decode the representations and generate the predictions.

The proposed LFA-ConvLSTM module plays an important role in prediction, where the key lies in its local flow attention (LFA) mechanism. Specifically, LFA yields three advantages. (1) LFA is flow-aware. Different from mixing inflows and outflows in the channel dimension, LFA explicitly exploits the flow correlations with a novel attention mechanism. (2) LFA is interpretable. It is strictly formulated by the truisms of traffic flow, and the concerned correlations can be well explained by attention weights. (3) LFA is efficient. It employs local attention to reduce computational cost and avoid irrelevant distractions.

LFA-ConvLSTM further has two variants for two flows. (1) In the inflow branch, an inflow–outflow LFA-ConvLSTM (IO-LFA-ConvLSTM) is designed to model the influence of outflows on inflows. (2) In the outflow branch, an outflow–inflow LFA-ConvLSTM (OI-LFA-ConvLSTM) can capture the impact of inflows on outflows. The details of these two LFA-ConvLSTM will be introduced in the next section.

Moreover, to make a multi-step prediction, the output at time t will serve as the input at time $t+1$, which is commonly used in previous works (Guen & Thome, 2020; Lin et al., 2020; Yao et al., 2019). Finally, the overall workflow for Ms-TFP is shown in Algorithm 1.

4. Local flow attention-based ConvLSTM

In this section, we depict the proposed local flow attention-based ConvLSTM (LFA-ConvLSTM).

First of all, we need to clarify the truisms of traffic flow, which serve as canons to design LFA-ConvLSTM. Specifically, it has two aspects.

- **Truism of inflows:** The inflow of a region at time $t+1$ is affected by the outflows of its neighborhood at time t.
- **Truism of outflows:** The outflow of a region at time $t+1$ depends on its inflow at time t.

Based on the truisms, we design the LFA-ConvLSTM, which is shown in Fig. 3. Concretely, it consists of three parts. (1) A ConvLSTM part accounts for generating flow-specific representation H'_t. H'_t also provides the future information for truisms. (2) A local flow attention (LFA) part is proposed to explicitly model the correlations between inflows and outflows, which follows the above truisms. LFA further has two variants, namely inflow–outflow LFA and outflow–inflow LFA, corresponding to the two aspects of truisms respectively. The output of this part is denoted as zf. (3) A feature aggregation part is utilized to fuse the flow-specific representation (H'_t) and correlations between flows (zf).

Next, we introduce the three parts of LFA-ConvLSTM in details.

4.1. ConvLSTM

ConvLSTM is a representative benchmark for spatiotemporal prediction. In this paper, we leverage it to generate flow-specific representation H'_t. H'_t plays two important roles.

(1) H'_t can capture the independent spatiotemporal dynamics of a specific traffic flow. For example, as for the inflow, X^1_t in IO-LFA-ConvLSTM means the inflow at time t, i.e., $X^1_t = X^{in}_t$. Then, ConvLSTM part updates the inflow dynamics as follows:

$$\begin{aligned} i_t &= \sigma(W_i * [X^1_t; H_{t-1}] + b_i) \\ f_t &= \sigma(W_f * [X^1_t; H_{t-1}] + b_f) \\ \widetilde{C}_t &= tanh(W_c * [X^1_t; H_{t-1}] + b_c) \\ C_t &= f_t \circ C_{t-1} + i_t \circ \widetilde{C}_t \\ o_t &= \sigma(W_o * [X^1_t; H_{t-1}] + b_o) \\ H'_t &= o_t \circ tanh(C_t) \end{aligned}, \tag{2}$$

where i_t, f_t, and o_t mean the input gate, forget gate, and output gate, respectively. C_t is the cell state and H_t is the hidden state.

Algorithm 1 Multi-step Traffic Flow Prediction

Input: A historical sequence with length L $(X_1, X_2, ..., X_L)$
Output: A prediction future sequence with length K $(\widehat{X}_{L+1}, \widehat{X}_{L+2}, ..., \widehat{X}_{L+K})$

Initialize states: $C_0^{in},\ H_0^{in},\ F_0^{in},\ C_0^{out},\ H_0^{out},\ F_0^{out} = \mathbf{0}$
for $t = 1, 2, ..., L$ **do**
  $C_t^{in}, H_t^{in}, F_t^{in} = \text{IO-LFA-ConvLSTM}(X_t^{in}, X_t^{out}, C_{t-1}^{in}, H_{t-1}^{in}, F_{t-1}^{in})$
  $C_t^{out}, H_t^{out}, F_t^{out} = \text{OI-LFA-ConvLSTM}(X_t^{out}, X_t^{in}, C_{t-1}^{out}, H_{t-1}^{out}, F_{t-1}^{out})$
  $\widehat{X}_{t+1}^{in} = CNN(H_t^{in})$
  $\widehat{X}_{t+1}^{out} = CNN(H_t^{out})$
for $t = L+1, L+2, ..., L+K$ **do**
  $C_t^{in}, H_t^{in}, F_t^{in} = \text{IO-LFA-ConvLSTM}(\widehat{X}_t^{in}, \widehat{X}_t^{out}, C_{t-1}^{in}, H_{t-1}^{in}, F_{t-1}^{in})$
  $C_t^{out}, H_t^{out}, F_t^{out} = \text{OI-LFA-ConvLSTM}(\widehat{X}_t^{out}, \widehat{X}_t^{in}, C_{t-1}^{out}, H_{t-1}^{out}, F_{t-1}^{out})$
  $\widehat{X}_{t+1}^{in} = CNN(H_t^{in})$
  $\widehat{X}_{t+1}^{out} = CNN(H_t^{out})$

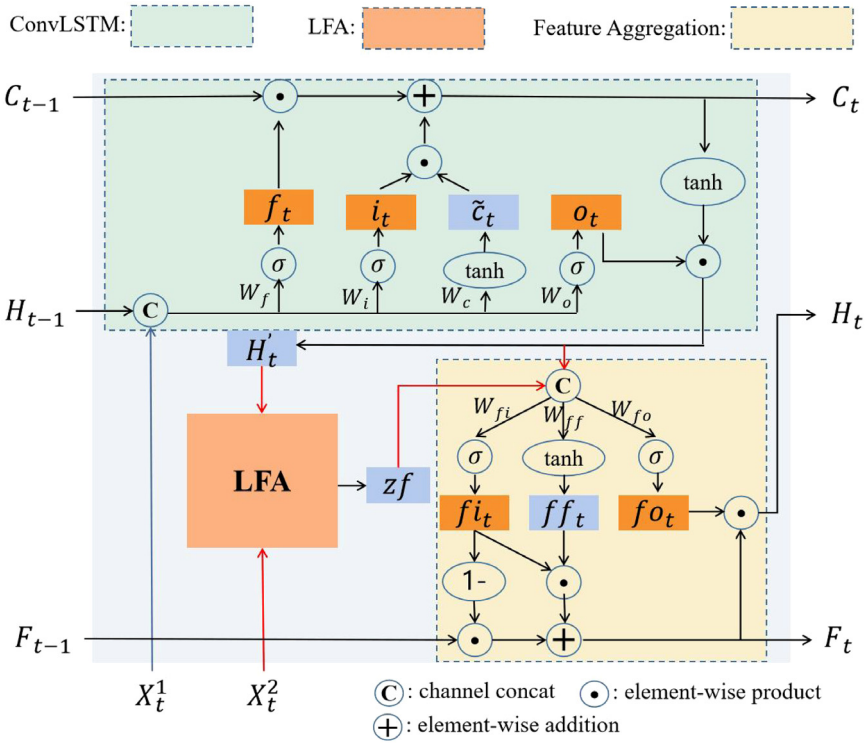


Fig. 3. The proposed LFA-ConvLSTM. It consists of a ConvLSTM part to capture flow-specific dynamics, an LFA part to learn the correlations between flows, and a feature aggregation part to fuse the above two features. Here, C_t and H_t represent the cell state and hidden state, respectively. F_t is a new state to memorize flow-aware information. As for IO-LFA-ConvLSTM, X_t^1 indicates the inflow and X_t^2 is the outflow. As for OI-LFA-ConvLSTM, they mean the opposite.

As for the outflow, $X_t^1 = X_t^{out}$ in OI-LFA-ConvLSTM, and the dynamics of outflow are updated in the same way.

(2) H'_t can provide future information, which will be utilized in LFA. Specifically, according to the truisms of inflow and outflow, one need to acquire the information at time $t+1$ for modeling the flow correlations. However, X_{t+1} has not yet been generated. Fortunately, H'_t can serve as a substitute. This is because H'_t has updated the flow-specific dynamics, and is capable to predict the next flow. Actually, in the vanilla ConvLSTM, H'_t is exactly used to generate the prediction.

4.2. Local flow attention

Local flow attention mechanism is proposed to capture the correlations between inflows and outflows. It explicitly leverages the above two truisms with two variants.

4.2.1. Inflow-outflow LFA

The first form of LFA is the inflow–outflow LFA (IO-LFA), which accounts for modeling the impact of outflows on inflows. **It follows the truism of inflows**. Specifically, the vehicles or crowds flowing into region (i, j) at time $t+1$ depend on those flowing out from its neighborhood at time t. This can be understood that some vehicles or crowds start from (i, j)'s neighborhood at time t and arrive at (i, j) at time $t+1$, which is shown in Fig. 1(a).

The input of IO-LFA consists of two parts. (1) H'_t from ConvLSTM, providing the inflow information at time $t+1$. (2) $X_t^2 = X_t^{out}$, providing the outflow information at time t. Here, we denote the neighborhood of (i, j) as $N_{(i,j)}$, whose size is $k \times k$. Fig. 4 shows the workflow of IO-LFA.

Specifically, as for a particular region (i, j), $H'_t(i, j)$ implies its inflow at time $t+1$, and $X_t^{out}(N_{(i,j)})$ denotes the outflow of its neighborhood at time t. Then, we employ self-attention operations to generate the attention weights for neighborhood $N_{(i,j)}$,

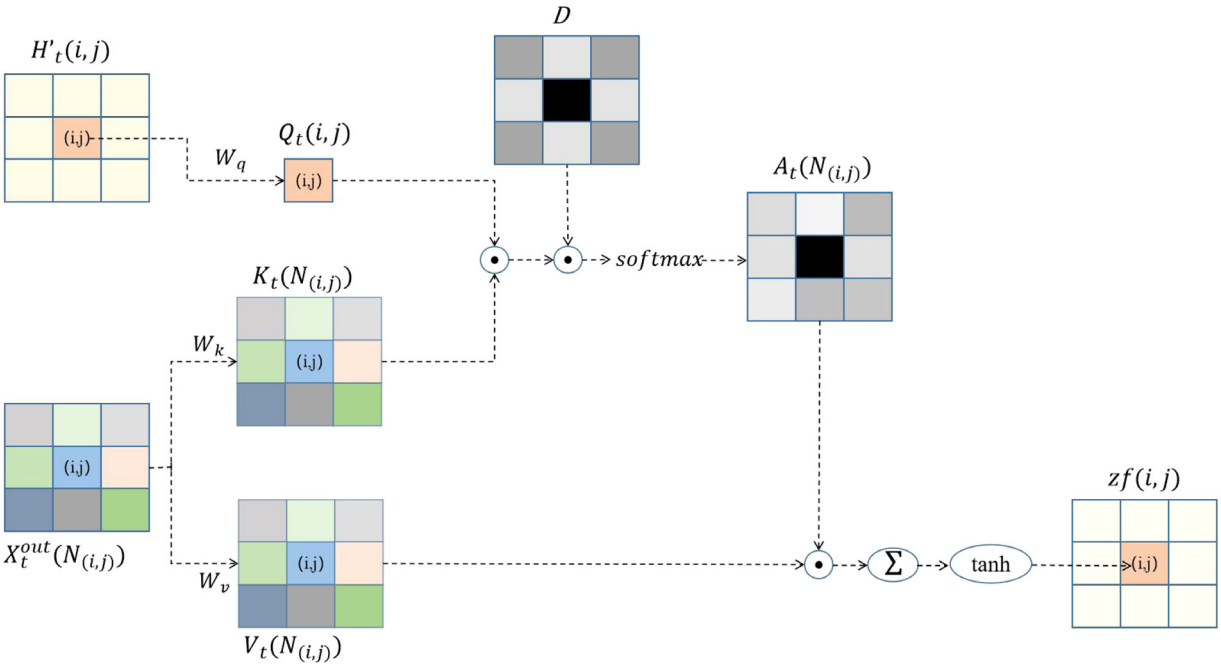


Fig. 4. The workflow of IO-LFA. $Q_t(i,j)$, $K_t(N_{(i,j)})$, and $V_t(N_{(i,j)})$ represent the query, key, and value, respectively. D is a distance-based prior weight matrix. $A_t(N_{(i,j)})$ denotes the attention weights of (i,j)'s neighborhood. $zf(i,j)$ combines the outflows via $A_t(N_{(i,j)})$.

0.60	0.73	0.77	0.73	0.60
0.73	0.88	0.93	0.88	0.73
0.77	0.93	0	0.93	0.77
0.73	0.88	0.93	0.88	0.73
0.60	0.73	0.77	0.73	0.60

Fig. 5. Distance-based prior weights D, where the neighborhood size is 5×5.

which are formulated as follows:

$$\begin{aligned} Q_t(i,j) &= W_q H'_t(i,j) \\ K_t(N_{(i,j)}) &= W_k X_t^{out}(N_{(i,j)}) \\ V_t(N_{(i,j)}) &= W_v X_t^{out}(N_{(i,j)}) \\ A_t(N_{(i,j)}) &= softmax(D \cdot ((Q_t(i,j))^T K_t(N_{(i,j)}))) \end{aligned}, \tag{3}$$

where $\{W_q, W_k, W_v\}$ is a set of mappings, implemented with 1×1 convolutions. Here, we first map $H'_t(i,j)$ to generate the query $Q_t(i,j)$. Also, $X_t^{out}(N_{(i,j)})$ is mapped to provide the key $K_t(N_{(i,j)})$ and value $V_t(N_{(i,j)})$. Afterwards, $Q_t(i,j)^T K_t(N_{(i,j)})$ is utilized to calculate the unnormalized correlations between inflows and outflows.

Before normalized with $softmax(\cdot)$ function, we develop a distance-based prior weights D, which is calculated as follows:

$$D_{x,y} = \begin{cases} exp\left(-\dfrac{(\Delta_x)^2 + (\Delta_y)^2}{(k-1)^2}\right), & (x,y) \neq (i,j) \\ 0, & (x,y) = (i,j) \end{cases}. \tag{4}$$

D shares the same regions with $N_{(i,j)}$. (x,y) means any point of D. Δ_x and Δ_y are the relative distances between (x,y) and (i,j). Fig. 5 shows the prior weights with neighborhood size being 5×5. Here, D has two characteristics. (1) The region closer to (i,j) has a greater weight. This is because vehicles or crowds are more likely to flow in from closer regions. (2) The weight of (i,j) is set to 0. This can be explained that the region (i,j) cannot flow into itself.

After obtaining the attention weights $A_t(N_{(i,j)})$, we aggregate the attention value $V_t(N_{(i,j)})$ as follows:

$$zf(i,j) = tanh\left(\sum A_t(N_{(i,j)}) \cdot V_t(N_{(i,j)})\right). \tag{5}$$

$zf(i,j)$ measures the sum of outflows from (i,j)'s neighborhood.

In the IO-LFA, interpretability is reflected in two aspects. (1) It is designed according to the truism of inflows, where one can plainly observe how the inflows and outflows interact. (2) The learned attention weights $A_t(N_{(i,j)})$ can reflect the degree of flows correlations, i.e., a larger $A_t(N_{(i,j)})$ indicates more interactions.

At the end of this part, we analyze the computational complexity of IO-LFA. Specifically, according to Eq. (3), $Q_t(i,j)^T K_t(N_{(i,j)})$ occupies the majority of computing resources, which is $O(k \times k)$ for a particular region (i,j). Then, for the whole $m \times n$ grid map G, the complexity is $O(mnk^2)$. Here, $k \times k$ denotes the neighborhood size, and it is a constant with small values, such as 3, 5, 7, etc. Notably, the complexity of conventional global spatial attention is $O(m^2n^2)$, i.e., neighborhood size is set to $m \times n$. One can find that our model has a lower computational complexity.

4.2.2. Outflow-inflow LFA

Similar to IO-LFA, an outflow–inflow LFA (OI-LFA) is proposed to model the impact of inflow on outflow. **OI-LFA follows the truism of outflows**. Specifically, the vehicles or crowds flowing out from region (i,j) at time $t+1$ depend on its inflow at time t. It can be understood that a large inflow will lead to a large outflow.

The inputs of OI-LFA also consist of two parts. (1) H'_t from ConvLSTM, providing the outflow information at time $t+1$. (2) $X_t^2 = X_t^{in}$, providing the inflow information at time t. Fig. 6 shows the workflow of OI-LFA.

OI-LFA has a similar calculation with IO-LFA. However, it is performed in a pixel-to-pixel manner, instead of a pixel-to-neighborhood manner in IO-LFA. This is because the outflow of a region depends more on its own previous inflow. Vehicles and

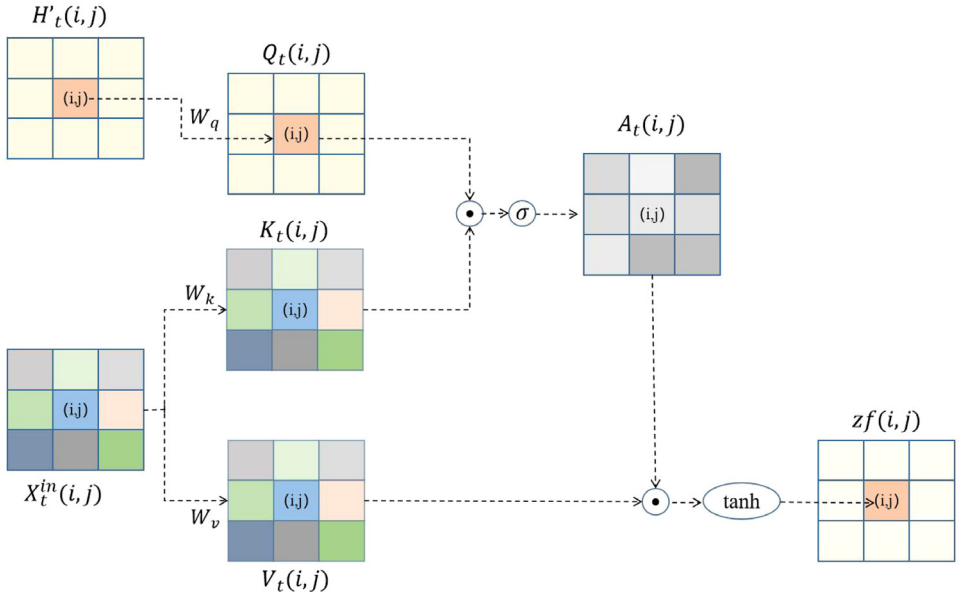


Fig. 6. The workflow of OI-LFA. $Q_t(i,j)$, $K_t(N_{(i,j)})$, and $V_t(N_{(i,j)})$ represent the query, key, and value, respectively. $A_t(N_{(i,j)})$ denotes the attention weights of inflows. $zf(i,j)$ is the flow-aware feature.

crowds cannot flow into (i,j)'s neighborhood and flow out from (i,j) at one time.

Formally, OI-LFA is formulated as follows:

$$\begin{aligned} Q_t(i,j) &= W_q H'_t(i,j) \\ K_t(i,j) &= W_k X_t^{in}(i,j) \\ V_t(i,j) &= W_v X_t^{in}(i,j) \\ A_t(i,j) &= \sigma(Q_t(i,j) \cdot K_t(i,j)) \\ zf(i,j) &= tanh(A_t(i,j) \cdot V_t(i,j)) \end{aligned} . \tag{6}$$

Here, $Q_t(i,j)$, $K_t(i,j)$, and $V_t(i,j)$ are the query, key, and value of self attention. $A_t(i,j)$ measures the correlation between inflows and outflows. $zf(i,j)$ is the flow-aware feature.

In the OI-LFA, its interpretability is also reflected in two aspects. (1) It follows the truism of outflows, where the flows interactions are clearly clarified. (2) The learned attention weights $A_t(N_{(i,j)})$ can also reflect the degree of flow correlations.

At the end of this part, we also show the computational complexity of OI-LFA. According to Eq. (6), one can easily conclude that the complexity is $O(mn)$, which is computationally inexpensive.

4.3. Feature aggregation

After obtaining flow-specific dynamics (H'_t) and correlations between flows (zf), a feature aggregation part is proposed to fuse them.

Specifically, we develop a novel state F_t to memorize the flow-aware information, which is calculated as follows:

$$\begin{aligned} fi_t &= \sigma(W_{fi} * [H'_t; zf] + b_{fi}) \\ ff_t &= tanh(W_{ff} * [H'_t; zf] + b_{ff}). \\ F_t &= fi_t \circ ff_t + (1 - fi_t) \circ F_{t-1} \end{aligned} \tag{7}$$

Here, ff_t is the candidate flow state, which combines H'_t and zf. fi_t is the flow input gate. It is utilized to balance the historical flow information F_{t-1} and candidate flow information ff_t. W and b are convolution parameters and bias, respectively.

Finally, similar to ConvLSTM, the output of feature aggregation is a dot product between an output gate fo_t and the updated flow-aware state F_t:

$$\begin{aligned} fo_t &= \sigma(W_{fo} * [H'_t; zf] + b_{fo}) \\ H_t &= fo_t \circ F_t \end{aligned} . \tag{8}$$

Here, H_t is also the output of LFA-ConvLSTM. It will be sent to the CNN decoder module in Fig. 2 to generate the prediction.

At the end of this part, we summarize all the calculations in LFA-ConvLSTM as follows:

$$\begin{aligned} i_t &= \sigma(W_i * [X_t^1; H_{t-1}] + b_i) \\ f_t &= \sigma(W_f * [X_t^1; H_{t-1}] + b_f) \\ \widetilde{C}_t &= tanh(W_c * [X_t^1; H_{t-1}] + b_c) \\ C_t &= f_t \circ C_{t-1} + i_t \circ \widetilde{C}_t \\ o_t &= \sigma(W_o * [X_t^1; H_{t-1}] + b_o) \\ H'_t &= o_t \circ tanh(C_t) \\ zf &= LFA(H'_t, X_t^2) \\ fi_t &= \sigma(W_{fi} * [H'_t; zf] + b_{fi}) \\ ff_t &= tanh(W_{ff} * [H'_t; zf] + b_{ff}) \\ F_t &= fi_t \circ ff_t + (1 - fi_t) \circ F_{t-1} \\ fo_t &= \sigma(W_{fo} * [H'_t; zf] + b_{fo}) \\ H_t &= fo_t \circ F_t \end{aligned} . \tag{9}$$

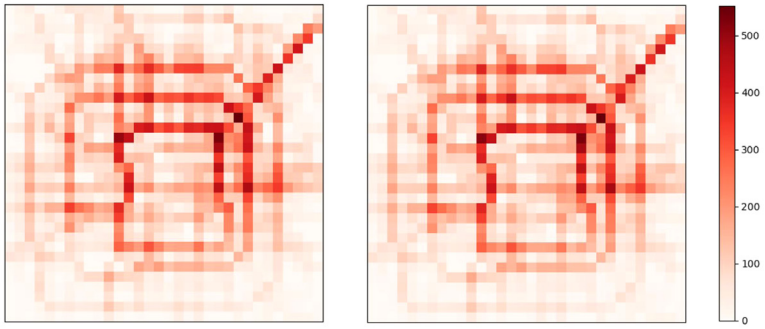
Here, as for the inflow branch, $X_t^1 = X_t^{in}$, $X_t^2 = X_t^{out}$, and $LFA(\cdot)$ means the IO-LFA. As for the outflow branch, $X_t^1 = X_t^{out}$, $X_t^2 = X_t^{in}$, and $LFA(\cdot)$ is the OI-LFA.

5. Experiments

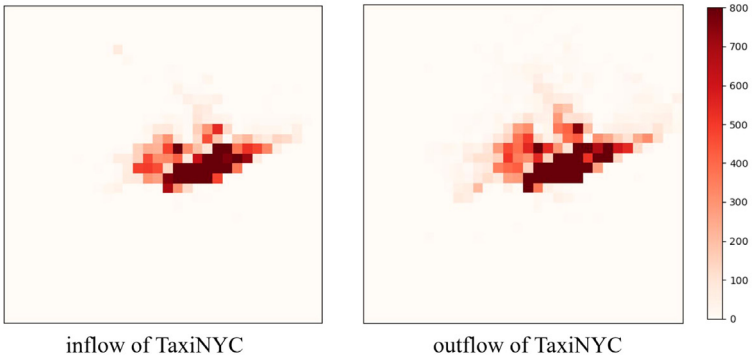
5.1. Datasets

Two popular traffic datasets are utilized in this paper, i.e., TaxiBJ and TaxiNYC. Specifically, (1) TaxiBJ consists of the taxicab GPS trajectory data in Beijing. (2) TaxiNYC records the yellow taxi trip from NYC Taxi and Limousine Commission (TLC). Table 1 lists the details of the two datasets.

Fig. 7 shows the inflow and outflow samples in two datasets. We can observe that the difference between inflow and outflow is small. This is because the time intervals in two datasets are



(a) Inflow and outflow samples in the TaxiBJ dataset.



(b) Inflow and outflow samples in the TaxiNYC dataset.

Fig. 7. Inflow and outflow samples in two datasets.

Table 1
Details of TaxiBJ and TaxiNYC.

Dataset	Time span	Time interval	Grid map size
TaxiBJ	2013.7.1–2013.10.30 2014.3.1–2014.6.30 2015.3.1–2015.6.30 2015.11.1–2016.4.10	30 min	32 × 32
TaxiNYC	2013.7.1–2016.6.30	1 h	32 × 32

relatively short, where the traffic flow can maintain a dynamic balance. Moreover, compared to TaxiBJ, the flows in TaxiNYC is far from uniformly distributed. Most of flows are gathered in small central regions, while few flow in the rest part.

Both datasets are divided into the training set, validation set, and test set. Specifically, (1) as for TaxiBJ, the last three weeks are selected for testing, three weeks before that for validating, and the rest for training. (2) As for TaxiNYC, the last six weeks are selected for testing, six weeks before that for validating, and the rest for training.

Furthermore, the input steps and output steps for two datasets are both 6, i.e., predicting the flows in the next 3 h for TaxiBJ and 6 h for TaxiNYC.

5.2. Parameter settings and evaluation metrics

5.2.1. Parameter settings

In this part, we clarify the parameter settings of proposed model, including those of IO-LFA-ConvLSTM, OI-LFA-ConvLSTM, and CNN decoder.

- Settings of IO-LFA-ConvLSTM: (1) As for ConvLSTM part, the number of convolution channels is 64, and the kernel size is 7 × 7. (2) As for IO-LFA part, the size of neighborhood $N_{(i,j)}$ is 7 × 7. W_q, W_k, and W_v in Eq. (3) for attention query are implemented with 1 × 1 convolutions. (3) As for feature aggregation part, the convolutions in Eqs. (7) and (8) both have 64 channels, with kernel size being 7 × 7.
- Settings of OI-LFA-ConvLSTM: Its ConvLSTM part and feature aggregation part have the same settings with IO-LFA-ConvLSTM, and W_q, W_k, and W_v in Eq. (6) are also implemented with 1 × 1 convolutions.
- Settings of CNN decoder: CNN decoder is a two-layer convolutional network. It reduces the number of channels from 64 to 16, and finally to 1. The kernel sizes in both layers are 3 × 3.

The objective function is the widely used mean square error. The model is trained with Adam optimizer, where the learning rate is 4×10^{-4}. Furthermore, the batch size is set to 32. The source code is available.[2]

The above settings are shared by two datasets in this paper.

5.2.2. Evaluation metrics

We employ two commonly used metrics to evaluate the prediction performance, i.e., root mean square error (RMSE) and mean absolute percentage error (MAPE). Specifically, the ground-truth value is denote as $\mathcal{X}_i$, and the prediction is $\widehat{\mathcal{X}}_i$. Here i indicates the ith sample sequence. Then RMSE and MAPE are

[2] https://github.com/hub5/LFAConvLSTM

calculated as follows:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\mathcal{X}_i - \widehat{\mathcal{X}}_i\right)^2}$$
$$MAPE = \frac{1}{N}\sum_{i=1}^{N}\frac{\left|\mathcal{X}_i - \widehat{\mathcal{X}}_i\right|}{\mathcal{X}_i}, \quad (10)$$

where N is the size of test set.

As for multi-step prediction of inflow and outflow, we will show the step-wise error and step-average error. Furthermore, the flow values less than 10 are not counted for evaluating, which a common practice used in industry and academy (Yao et al., 2019, 2018). This is mainly due to the following two reasons: (1) low-value flow is usually not important in practice. (2) There may be a large number of low-value flows in practical dataset, such as TaxiNYC shown in Fig. 7(b). This will affect the focus on high-value ones.

5.3. Baseline methods

We compare the proposed model with the following baseline methods:

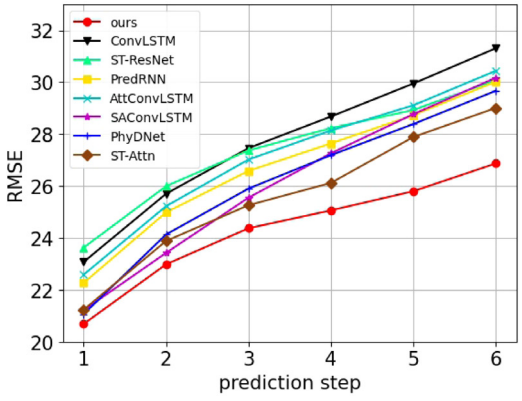
- HA (Historical Average): It averages the values of historical inflows and outflows in the corresponding periods.
- ConvLSTM (Convolutional Long Short-Term Memory) (Shi et al., 2015): It integrates the convolution operation into LSTM to capture spatial dependencies and temporal dynamics simultaneously.
- ST-ResNet (Spatial–Temporal Residual Network) (Zhang et al., 2017): It employs the residual neural network-based framework to model the temporal closeness, period, and trend properties of crowd traffic. ST-ResNet is proposed for next-step prediction, and we use the output at time t as the input at time $t+1$ to make a multi-step prediction.
- PredRNN (Predictive Recurrent Neural Network) (Wang, Long, Wang, Gao, & Yu, 2017): It develops a new spatiotemporal LSTM to deliver memory states both vertically and horizontally in multi-layer architecture.
- AttConvLSTM (Attention-based ConvLSTM) (Zhou et al., 2018): It is based on the ConvLSTM, and proposes an attention mechanism to incorporate historical traffic patterns.
- SAConvLSTM (Self-Attention ConvLSTM) (Lin et al., 2020): It utilizes a self-attention memory (SAM) to memorize features with long-range spatiotemporal dependencies.
- PhyDNet (Physical Dynamic Network) (Guen & Thome, 2020): It proposes a recurrent physical cell to learn physical dynamics in spatiotemporal data.
- ST-Attn (Spatiotemporal Attention Network) (Zhou, Li, et al., 2020): It is a transformer-based model, which employs a pure attention framework without RNNs.

Here, we further clarify the difference in attention mechanism between our model and the above methods. Specifically, AttConvLSTM focuses on incorporating historical traffic patterns, but our model aims to exploit the spatiotemporal correlations between inflows and outflows. As for SAConvLSTM and ST-Attn, two important aspects distinguish ours with them. (1) They both concatenate inflows and outflows in the channel dimension, then the spatiotemporal attention is performed on the mixed flows. Consequently, the flow correlations are not explicitly modeled and well interpreted. (2) They both employ global spatial attention, which is computationally expensive and may introduce false attention.

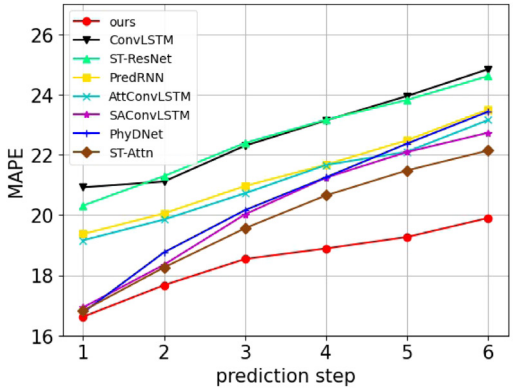
All of the methods run on a computer with an Intel Core i9-10920X CPU at 3.50 GHz, and a GeForce GTX 3090Ti GPU is utilized for acceleration.

Table 2
RMSE and MAPE of all methods.

Dataset	Methods	Inflow		Outflow		Average	
		RMSE	MAPE	RMSE	MAPE	RMSE	MAPE
TaxiBJ	HA	45.94	31.12	46.02	31.23	45.98	31.17
	ConvLSTM	27.65	22.60	27.81	22.76	27.73	22.71
	ST-ResNet	27.32	22.55	27.42	22.65	27.37	22.60
	PredRNN	26.68	21.31	26.77	21.37	26.72	21.34
	AttConvLSTM	27.12	21.08	27.18	20.89	27.15	20.98
	SAConvLSTM	25.75	19.51	26.40	21.02	26.07	20.26
	PhyDNet	25.97	20.45	26.13	20.46	26.05	20.46
	ST-Attn	25.23	19.18	25.89	20.26	25.56	19.72
	Ours	**24.18**	**18.27**	**24.58**	**18.69**	**24.38**	**18.48**
TaxiNYC	HA	80.67	32.67	62.71	29.65	71.69	31.16
	ConvLSTM	59.04	28.26	45.10	28.02	52.07	28.14
	ST-ResNet	57.73	30.24	44.87	29.01	51.30	29.62
	PredRNN	54.98	28.60	42.89	27.27	48.93	27.93
	AttConvLSTM	55.28	28.17	44.16	27.10	49.72	27.63
	SAConvLSTM	60.39	28.95	46.86	28.94	53.62	28.95
	PhyDNet	56.84	25.83	44.05	28.25	50.44	27.04
	ST-Attn	54.87	26.72	41.28	27.23	48.07	26.97
	Ours	**53.92**	**25.10**	**39.11**	**25.20**	**46.51**	**25.15**



(a) Step-wise RMSE of TaxiBJ.



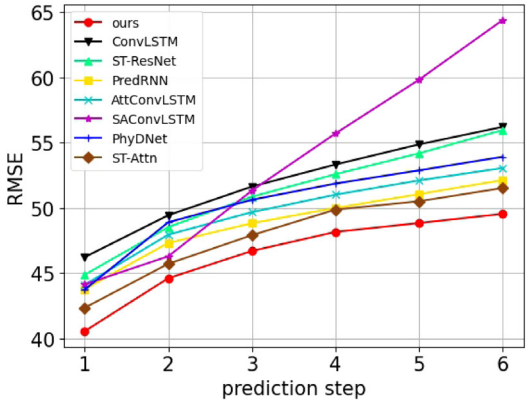
(b) Step-wise MAPE of TaxiBJ.

Fig. 8. Step-wise RMSE and MAPE of TaxiBJ.

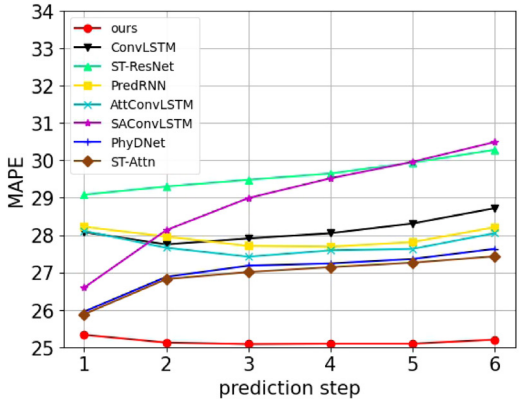
5.4. Results and analysis

5.4.1. Compared with baselines

In this part, we evaluate the performance of all methods. Table 2 reports the step-average RMSE and MAPE. Figs. 8 and 9 show the step-wise RMSE and MAPE.



(a) Step-wise RMSE of TaxiNYC.



(b) Step-wise MAPE of TaxiNYC.

Fig. 9. Step-wise RMSE and MAPE of TaxiNYC.

According to the results, we can find that: (1) our method achieves the best performance against all compared ones. Specifically, as shown in Table 2, we improve the average RMSE by 1.18 and average MAPE by 1.24 for TaxiBJ dataset, where the best existing method is ST-Attn. As for TaxiNYC dataset, we also improve the average RMSE by 1.56 and average MAPE by 1.82. Furthermore, both RMSE and MAPE of our model for each flow are relatively lower too. This indicates that the proposed flow-aware and local attention-based framework is effective and superior. (2) As shown in Figs. 8 and 9, our model consistently outperforms other baselines at all time steps. Although some methods, such as PhyDNet and ST-Attn, have a similar performance at the beginning, we show a significant lead as the prediction step increases. This shows that our model learns better spatiotemporal dynamics, especially for multi-step prediction. The reasons can be explained that we efficiently exploit the correlations between inflows and outflows, which is exceedingly important for traffic prediction. (3) As for SAConvLSTM, which integrates global spatial attention into ConvLSTM, it performs worse than our model. Especially for TaxiNYC dataset, it is even inferior than the vanilla ConvLSTM. According to Fig. 7(b), we can observe that the flows in TaxiNYC is far from uniformly distributed. In this case, more false attention caused by global attention will significantly degrade the prediction performance. Nevertheless, our model can avoid wrong long-distance noise by local attention queries. We only focus on the neighborhood of a given region, which usually show more correlations. As a result, our model predicts more accurately than SAConvLSTM.

Table 3
RMSE and MAPE of LFA-ConvLSTM's variants.

Dataset	Methods	Inflow		Outflow		Average	
		RMSE	MAPE	RMSE	MAPE	RMSE	MAPE
TaxiBJ	w/o LFA	27.37	22.20	27.70	23.03	27.54	22.62
	w/o IO-LFA	26.01	20.27	25.11	18.85	25.56	19.56
	w/o OI-LFA	24.95	19.45	26.19	21.29	25.58	20.37
	Ours	**24.18**	**18.27**	**24.58**	**18.69**	**24.38**	**18.48**
TaxiNYC	w/o LFA	61.31	29.26	42.79	27.37	52.05	28.32
	w/o IO-LFA	57.75	28.66	39.68	25.69	48.71	27.17
	w/o OI-LFA	55.07	26.70	42.69	27.57	48.88	27.14
	Ours	**53.92**	**25.10**	**39.11**	**25.20**	**46.51**	**25.15**

5.4.2. Ablation study

In this part, we investigate the effectiveness of proposed LFA mechanism by removing it. Specifically, we denote the models without IO-LFA, OI-LFA, and both of them as *w/o IO-LFA*, *w/o OI-LFA*, and *w/o LFA*, respectively. Table 3 reports the results of these variants.

We can find that: (1) discarding either IO-LFA or OI-LFA degrades the performance. For example, RMSEs of *w/o IO-LFA* and *w/o OI-LFA* for TaxiNYC dataset are worse by 2.2 and 2.37, respectively. This shows that both of IO-LFA and OI-LFA are important for prediction, and leveraging both of them can achieve the best performance. (2) Compared with the model without both LFA, i.e., *w/o LFA*, *w/o IO-LFA* and *w/o OI-LFA* both obtain an improvement. For instance, RMSEs of *w/o IO-LFA* and *w/o OI-LFA* for TaxiNYC dataset are improved by 3.34 and 3.17, respectively. This indicates that LFA indeed contributes to a better prediction by capturing flow correlations. (3) *w/o IO-LFA* shows a more significant performance degradation to inflows. For example, as for TaxiNYC dataset, the inflow RMSE of *w/o IO-LFA* is worse by 3.83, while the outflow RMSE only worse by 0.57. This suggests that IO-LFA has a greater impact on inflows. Actually, IO-LFA is designed to model the influence of outflows on inflows. Therefore, removing it naturally degrades the prediction of inflows. Furthermore, the performance of outflows also deteriorates, since the inflows and outflows are interactional. Once one of them is predicted inaccurately, the other will also be affected. (4) Similarly, *w/o OI-LFA* shows a more pronounced performance degradation to outflows. For example, the outflow RMSE of *w/o OI-LFA* for TaxiNYC is worse by 3.58, while the inflow RMSE only worse by 1.15. Here, *w/o OI-LFA* is designed to model the impact of inflows on outflows.

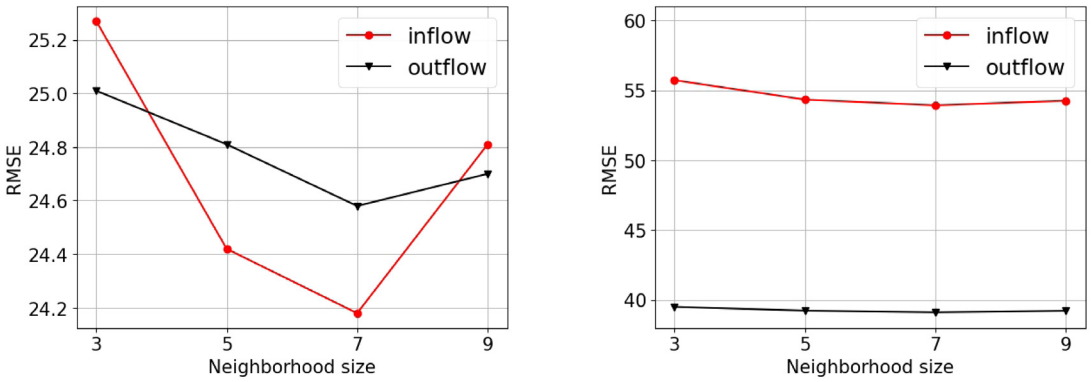
Generally, IO-LFA and OI-LFA both improve the prediction performance, and the best result is achieved by combining them. This demonstrates that the proposed LFA is effective.

Moreover, we also explore the effect of neighborhood size in IO-LFA on prediction. Fig. 10 reports the RMSEs of two datasets, where the neighborhood size is set to 3×3, 5×5, 7×7, and 9×9.

We can find that: (1) the prediction performance is improved gradually as the size changes from 3 to 7. This is because a larger size can provide a wider field, thus capturing more relevant spatial dependencies. (2) Nevertheless, as for 9×9 size, the performance degrades. This can be explained that an oversized neighborhood may introduce false attention.

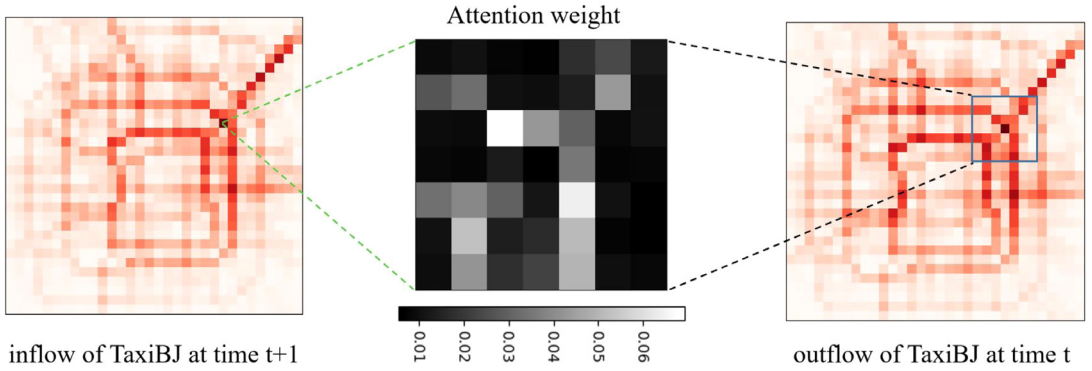
5.4.3. Efficiency analysis

In this paper, a novel local attention mechanism is proposed, and a new ConvLSTM-based spatiotemporal cell is developed. To evaluate the efficiency, we compare our model with the vanilla ConvLSTM and global self-attention ConvLSTM (SAConvLSTM).

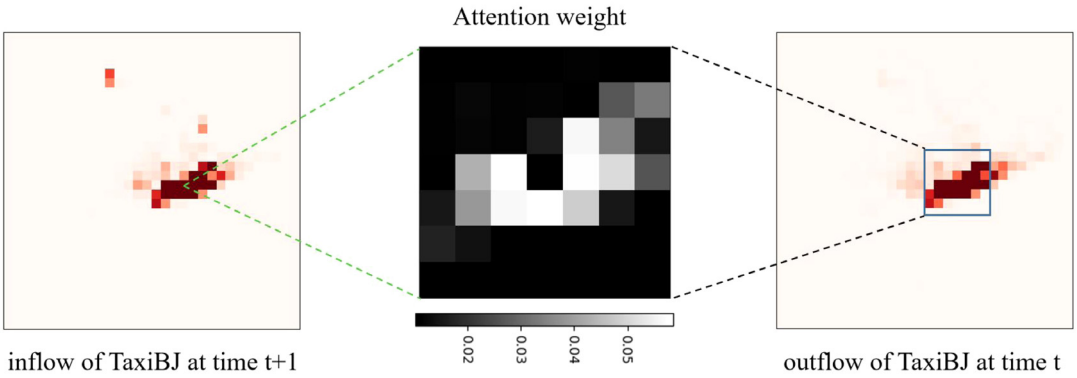


(a) RMSE of TaxiBJ dataset. (b) RMSE of TaxiNYC dataset.

Fig. 10. RMSEs of two datasets with neighborhood size being 3×3, 5×5, 7×7, and 9×9.



(a) IO-LFA attention weights of TaxiBJ sample.



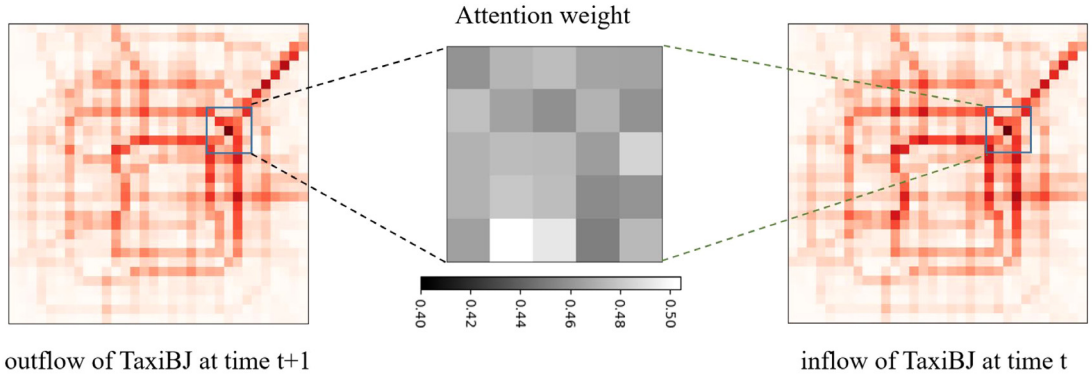
(b) IO-LFA attention weights of TaxiNYC sample.

Fig. 11. IO-LFA attention weights of two datasets. In each subfigure, the left part is the inflow at time $t+1$ and the right part means the outflow at time t. The middle part shows the attention weights, which reflects the correlations of a target region (i, j) and its neighborhood.

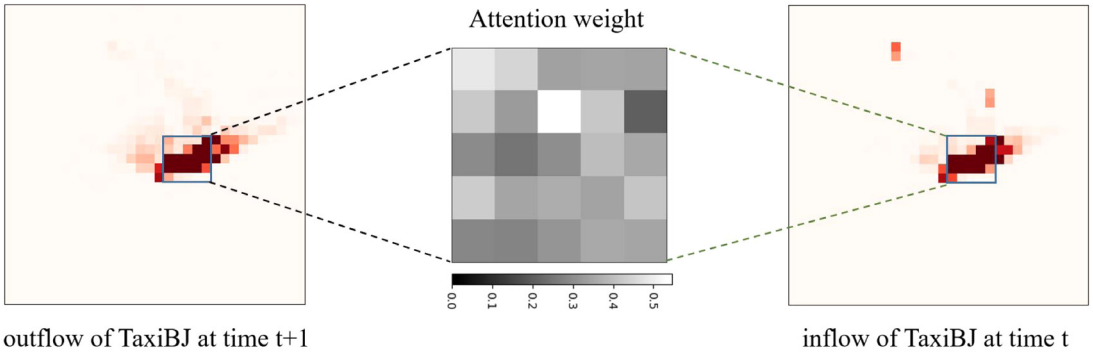
The prediction time is reported in Table 4.[3] Here, we also show the results about various neighborhood sizes in IO-LFA, including 5×5, 7×7, and 9×9.

We can find that: (1) the prediction time of our method is shorter than SAConvLSTM. This means that the proposed local flow attention is more efficient. By effective local queries, we utilize less time to calculate region-wise correlations. Combined with the prediction error reported in Table 2, our model achieves a more accurate prediction with less computation than SAConvLSTM. (2) As the neighborhood size increases, we need more prediction time. In Section 4.2.1, we clarified that the computational complexity of IO-LFA is $o(mnk^2)$. A larger k leads to a longer prediction time because more correlation queries are performed.

[3] Here, the test batch size is set to 24 instead of 32, because their is no enough GPU memory for SAConvLSTM. The total number of test set is 1008.



(a) OI-LFA attention weights of TaxiBJ sample.



(b) OI-LFA attention weights of TaxiNYC sample.

Fig. 12. OI-LFA attention weights of two datasets. In each subfigure, the left part is the outflow at time $t+1$ and the right part means the inflow at time t. The middle part shows the attention weights, which reflects the correlations of a region between its outflow and inflow.

Table 4
Efficiency evaluation in terms of prediction time.

Dataset	Methods	Prediction time (s)
TaxiBJ	ConvLSTM	3.11
	SAConvLSTM	7.37
	Ours (5 × 5)	4.67
	Ours (7 × 7)	5.01
	Ours (9 × 9)	5.97
TaxiNYC	ConvLSTM	3.05
	SAConvLSTM	7.34
	Ours (5 × 5)	4.62
	Ours (7 × 7)	4.98
	Ours (9 × 9)	5.94

5.5. Local flow attention analysis

In this part, we present the visual analysis of local flow attention.

5.5.1. IO-LFA analysis

First, we show the learned flow correlations in IO-LFA, i.e., how the inflow of a given region (i, j) is affected by the outflows of its neighborhood. According to Eq. (3), the attention weights $A_t\left(N_{(i,j)}\right)$ measures the degree of correlations. Fig. 11 reports $A_t\left(N_{(i,j)}\right)$ of two samples.

We can find that the attention weights are not uniformly distributed. Generally, the region with a large weight has two characteristics. (1) It is close to the concerned region (i, j). This can be explained that vehicles or crowds are more likely to flow between adjacent regions. Such a local attribute of traffic flows motivates us to leverage local spatial attention rather than the global mode. (2) It has large outflows, which is an important precondition. Obviously, a region with small outflows cannot contributes a lot to the inflows of other regions, even if they are close. For example, in the TaxiNYC sample, $(i+2, j)$[4] has a quite small weight, even though it is close to (i, j). However, $(i-2, j+3)$ shows more interactions with (i, j), since it has large outflows.

The above two observations well reflect some common sense. Actually, our LFA is formulated by the truisms of traffic flow, thus it can be understood as self-explanatory. In this part, experimental results further demonstrate that the proposed LFA has good interpretability.

5.5.2. OI-LFA analysis

Here, we show the learned flow correlations in OI-LFA, i.e., the relation between the outflow of a region (i, j) and its inflow at the previous time step. According to Eq. (6), the attention weights $A_t(i, j)$ reflects the relation. Fig. 12 shows $A_t(i, j)$ of two samples.

We can observe that the attention weights vary slightly and most of them are close to 0.5. This suggests that there is indeed a significant correlation between inflows and outflows. Specifically, the attention weights in OI-LFA reflect the degree to which the calculation of regional outflow depends on its previous inflow.[5]

[4] $(i+2, j)$ is two pixels below (i, j).

[5] Here, we need to explain that the flow-specific feature H'_t is the other important factor for prediction.

The non-zero weights indicate that the flow-aware features play an important role in prediction. Combining with the ablation studies on the model without OI-LFA, we can conclude that OI-LFA helps learn better outflow dynamics, thus contributing to a more accurate prediction.

6. Conclusion and future work

In this paper, we propose a novel local flow attention (LFA) mechanism for multi-step traffic flow prediction. LFA is formulated by the truisms of traffic flow, where the correlations between inflows and outflows are explicitly modeled. Therefore, our model can be understood as self-explanatory. Furthermore, LFA leverages local attention to learn spatial dependencies, instead of the global mode as in previous works. This not only reduces the computational cost but also avoids long-distance false attention. Based on LFA, we further develop a new spatiotemporal cell LFA-ConvLSTM (LFA-based convolutional long short-term memory) to capture complex dynamics in traffic data. Extensive experiments on two datasets demonstrate that our method can achieve a better prediction performance by explicitly and effectively modeling the flow correlations. Moreover, visual analyses also show that LFA can reflect some common sense about traffic flows. For example, a closer region with large outflows usually contributes more to the inflows of its neighborhood. However, some auxiliary information, such as weather conditions and holiday events, is not taken into account in this work. How to leverage them in an interpretable way leaves a challenge for our study.

In the future, we suggest paying attention to an important and meaningful issue in traffic flow prediction, i.e., joint learning of intra-city and inter-city traffic data. Specifically, in this work, we investigate the traffic flow within a city, such as Beijing and New York. Nevertheless, for a metropolis, there are a large number of vehicles or people flowing in and out of its surrounding cities. Inter-city flows will inevitably affect intra-city flows. Therefore, it is significant to jointly learn the intra-city and inter-city traffic data, especially with the development of urban agglomeration. Such a study may provide new insights into discovering the relationship between cities.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This work was supported in part by National Natural Science Foundation of China under Grant 62272130, and Shenzhen Science and Technology Program, China under Grant KCXFZ20211020163403005, JCYJ20210324120208022, and JCYJ20200109113014456.

References

Ahmed, M. S., & Cook, A. R. (1979). *Analysis of freeway traffic time-series data by using Box–Jenkins techniques, no. 722*.

Ali, A., Zhu, Y., & Zakarya, M. (2022). Exploiting dynamic spatio-temporal graph convolutional neural networks for citywide traffic flows prediction. *Neural Networks*, *145*, 233–247.

An, J., Guo, L., Liu, W., Fu, Z., Ren, P., Liu, X., et al. (2021). IGAGCN: Information geometry and attention-based spatiotemporal graph convolutional networks for traffic flow prediction. *Neural Networks*, *143*, 355–367.

Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473.

Castro-Neto, M., Jeong, Y.-S., Jeong, M.-K., & Han, L. D. (2009). Online-SVR for short-term traffic flow prediction under typical and atypical traffic conditions. *Expert Systems with Applications*, *36*(3), 6164–6173.

Deng, D., Shahabi, C., Demiryurek, U., Zhu, L., Yu, R., & Liu, Y. (2016). Latent space model for road networks to predict time-varying traffic. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 1525–1534).

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., et al. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929.

Du, B., Peng, H., Wang, S., Bhuiyan, M. Z. A., Wang, L., Gong, Q., et al. (2019). Deep irregular convolutional residual LSTM for urban traffic passenger flows prediction. *IEEE Transactions on Intelligent Transportation Systems*, *21*(3), 972–985.

Fan, C., Zhang, Y., Pan, Y., Li, X., Zhang, C., Yuan, R., et al. (2019). Multi-horizon time series forecasting with temporal attention learning. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining* (pp. 2527–2535).

Fang, M., Tang, L., Yang, X., Chen, Y., Li, C., & Li, Q. (2021). FTPG: A fine-grained traffic prediction method with graph attention network using big trace data. *IEEE Transactions on Intelligent Transportation Systems*.

Guen, V. L., & Thome, N. (2020). Disentangling physical dynamics from unknown factors for unsupervised video prediction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 11474–11484).

Guo, N., Gu, K., Qiao, J., & Bi, J. (2021). Improved deep CNNs based on nonlinear hybrid attention module for image classification. *Neural Networks*, *140*, 158–166.

Hao, D., Ding, S., Qiu, L., Lv, Y., Fei, B., Zhu, Y., et al. (2020). Sequential vessel segmentation via deep channel attention network. *Neural Networks*, *128*, 172–187.

Hu, J., Shen, L., Albanie, S., Sun, G., & Vedaldi, A. (2018). Gather-excite: Exploiting feature context in convolutional neural networks. *Advances in Neural Information Processing Systems*, *31*.

Hu, J., Shen, L., & Sun, G. (2018). Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 7132–7141).

Huang, F., Yi, P., Wang, J., Li, M., Peng, J., & Xiong, X. (2022). A dynamical spatial-temporal graph neural network for traffic demand prediction. *Information Sciences*, *594*, 286–304.

Jiang, W., & Luo, J. (2022). Graph neural network for traffic forecasting: A survey. *Expert Systems with Applications*, Article 117921.

Jiang, R., Song, X., Fan, Z., Xia, T., Chen, Q., Miyazawa, S., et al. (2018). Deepurban-momentum: An online deep-learning system for short-term urban mobility prediction. In *Proceedings of the AAAI conference on artificial intelligence, vol. 32, no. 1*.

Lin, Z., Li, M., Zheng, Z., Cheng, Y., & Yuan, C. (2020). Self-attention convlstm for spatiotemporal prediction. In *Proceedings of the AAAI conference on artificial intelligence, vol. 34, no. 07* (pp. 11531–11538).

Liu, Y., Guan, R., Giunchiglia, F., Liang, Y., & Feng, X. (2021). Deep attention diffusion graph neural networks for text classification. In *Proceedings of the 2021 conference on empirical methods in natural language processing* (pp. 8142–8152).

Ma, C., Dai, G., & Zhou, J. (2021). Short-term traffic flow prediction for urban road sections based on time series analysis and LSTM_BILSTM method. *IEEE Transactions on Intelligent Transportation Systems*.

Ojeda, L. L., Kibangou, A. Y., & De Wit, C. C. (2013). Adaptive Kalman filtering for multi-step ahead traffic flow prediction. In *2013 American control conference* (pp. 4724–4729). IEEE.

Peng, D., Yang, W., Liu, C., & Lü, S. (2021). SAM-GAN: Self-attention supporting multi-stage generative adversarial networks for text-to-image synthesis. *Neural Networks*, *138*, 57–67.

Ramachandran, P., Parmar, N., Vaswani, A., Bello, I., Levskaya, A., & Shlens, J. (2019). Stand-alone self-attention in vision models. *Advances in Neural Information Processing Systems*, *32*.

Ren, Y., Zhao, D., Luo, D., Ma, H., & Duan, P. (2020). Global-local temporal convolutional network for traffic flow prediction. *IEEE Transactions on Intelligent Transportation Systems*.

Roy, A., Roy, K. K., Ali, A. A., Amin, M. A., & Rahman, A. M. (2021). Unified spatio-temporal modeling for traffic forecasting using graph neural network. In *2021 international joint conference on neural networks* (pp. 1–8). IEEE.

Shekhar, S., & Williams, B. M. (2007). Adaptive seasonal time series models for forecasting short-term traffic flow. *Transportation Research Record*, *2024*(1), 116–125.

Shi, X., Chen, Z., Wang, H., Yeung, D.-Y., Wong, W.-K., & Woo, W.-c. (2015). Convolutional LSTM network: A machine learning approach for precipitation nowcasting. *Advances in Neural Information Processing Systems*, *28*.

Shi, X., Qi, H., Shen, Y., Wu, G., & Yin, B. (2020). A spatial–temporal attention approach for traffic prediction. *IEEE Transactions on Intelligent Transportation Systems*, *22*(8), 4909–4918.

Sinha, K., Dong, Y., Cheung, J. C. K., & Ruths, D. (2018). A hierarchical neural attention-based text classifier. In *Proceedings of the 2018 conference on empirical methods in natural language processing* (pp. 817–823).

Tedjopurnomo, D. A., Bao, Z., Zheng, B., Choudhury, F., & Qin, A. K. (2020). A survey on modern deep neural network for traffic prediction: Trends, methods and challenges. *IEEE Transactions on Knowledge and Data Engineering*.

Tian, C., Xu, Y., Li, Z., Zuo, W., Fei, L., & Liu, H. (2020). Attention-guided CNN for image denoising. *Neural Networks*, *124*, 117–129.

Tong, Y., Chen, Y., Zhou, Z., Chen, L., Wang, J., Yang, Q., et al. (2017). The simpler the better: a unified approach to predicting original taxi demands based on large-scale online platforms. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 1653–1662).

Tran, D. T., Iosifidis, A., Kanniainen, J., & Gabbouj, M. (2018). Temporal attention-augmented bilinear network for financial time-series data analysis. *IEEE Transactions on Neural Networks and Learning Systems*, *30*(5), 1407–1418.

Van Der Voort, M., Dougherty, M., & Watson, S. (1996). Combining kohonen maps with ARIMA time series models to forecast traffic flow. *Transportation Research Part C (Emerging Technologies)*, *4*(5), 307–318.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., et al. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, *30*.

Wang, X., Girshick, R., Gupta, A., & He, K. (2018). Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 7794–7803).

Wang, F., Jiang, M., Qian, C., Yang, S., Li, C., Zhang, H., et al. (2017). Residual attention network for image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3156–3164).

Wang, Y., Long, M., Wang, J., Gao, Z., & Yu, P. S. (2017). Predrnn: Recurrent neural networks for predictive learning using spatiotemporal lstms. *Advances in Neural Information Processing Systems*, *30*.

Wang, Z., Su, X., & Ding, Z. (2020). Long-term traffic prediction based on lstm encoder-decoder architecture. *IEEE Transactions on Intelligent Transportation Systems*, *22*(10), 6561–6571.

Wang, Q., Wu, B., Zhu, P., Li, P., Zuo, W., & Hu, Q. (2020). ECA-net: Efficient channel attention for deep convolutional neural networks. In *The IEEE conference on computer vision and pattern recognition*.

Williams, B. M., & Hoel, L. A. (2003). Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: Theoretical basis and empirical results. *Journal of Transportation Engineering*, *129*(6), 664–672.

Woo, S., Park, J., Lee, J.-Y., & Kweon, I. S. (2018). Cbam: Convolutional block attention module. In *Proceedings of the European conference on computer vision* (pp. 3–19).

Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., et al. (2015). Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning* (pp. 2048–2057). PMLR.

Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., & Hovy, E. (2016). Hierarchical attention networks for document classification. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: Human language technologies* (pp. 1480–1489).

Yang, K., Zhang, H., Zhou, D., & Liu, L. (2021). TGAN: A simple model update strategy for visual tracking via template-guidance attention network. *Neural Networks*, *144*, 61–74.

Yang, Z., Zhou, Y., Chen, Z., & Ngiam, J. (2021). 3d-man: 3d multi-frame attention network for object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 1863–1872).

Yao, H., Tang, X., Wei, H., Zheng, G., & Li, Z. (2019). Revisiting spatial-temporal similarity: A deep learning framework for traffic prediction. In *Proceedings of the AAAI conference on artificial intelligence, vol. 33, no. 01* (pp. 5668–5675).

Yao, H., Wu, F., Ke, J., Tang, X., Jia, Y., Lu, S., et al. (2018). Deep multi-view spatial-temporal network for taxi demand prediction. In *Proceedings of the AAAI conference on artificial intelligence, vol. 32, no. 1*.

Zeng, J., Wu, S., Yin, Y., Jiang, Y., & Li, M. (2021). Recurrent attention for neural machine translation. In *Proceedings of the 2021 conference on empirical methods in natural language processing* (pp. 3216–3225).

Zhang, X., Huang, C., Xu, Y., Xia, L., Dai, P., Bo, L., et al. (2021). Traffic flow forecasting with spatial-temporal graph diffusion network. In *Proceedings of the AAAI conference on artificial intelligence, vol. 35, no. 17* (pp. 15008–15015).

Zhang, Y., Li, K., Li, K., Wang, L., Zhong, B., & Fu, Y. (2018). Image super-resolution using very deep residual channel attention networks. In *Proceedings of the European conference on computer vision* (pp. 286–301).

Zhang, B., Li, X., Xu, X., Leung, K.-C., Chen, Z., & Ye, Y. (2020). Knowledge guided capsule attention network for aspect-based sentiment analysis. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *28*, 2538–2551.

Zhang, J., Zheng, Y., & Qi, D. (2017). Deep spatio-temporal residual networks for citywide crowd flows prediction. In *Thirty-first AAAI conference on artificial intelligence*.

Zhao, H., Jia, J., & Koltun, V. (2020). Exploring self-attention for image recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 10076–10085).

Zheng, H., Lin, F., Feng, X., & Chen, Y. (2020). A hybrid deep learning model with attention-based conv-LSTM networks for short-term traffic flow prediction. *IEEE Transactions on Intelligent Transportation Systems*, *22*(11), 6910–6920.

Zhou, Y., Li, J., Chen, H., Wu, Y., Wu, J., & Chen, L. (2020). A spatiotemporal attention mechanism-based model for multi-step citywide passenger demand prediction. *Information Sciences*, *513*, 372–385.

Zhou, Y., Pan, L., Bai, C., Luo, S., & Wu, Z. (2021). Self-selective attention using correlation between instances for distant supervision relation extraction. *Neural Networks*, *142*, 213–220.

Zhou, X., Shen, Y., Zhu, Y., & Huang, L. (2018). Predicting multi-step citywide passenger demands using attention-based neural networks. In *Proceedings of the eleventh ACM international conference on web search and data mining* (pp. 736–744).

Zhou, F., Yang, Q., Zhang, K., Trajcevski, G., Zhong, T., & Khokhar, A. (2020). Reinforced spatiotemporal attentive graph neural networks for traffic forecasting. *IEEE Internet of Things Journal*, *7*(7), 6414–6428.