ST-Mamba: Spatial-Temporal Mamba for Traffic Flow Estimation Recovery using Limited Data

Doncheng Yuan*, Jianzhe Xue*, Jinshan Su†, Wenchao Xu‡, and Haibo Zhou*
*School of Electronic Science and Engineering, Nanjing University, Nanjing, China, 210023.
†Key Laboratory of Vibration Signal Capture and Intelligent Processing, Yili Normal University, Yining, China, 835000.
‡Department of Computing, the Hong Kong Polytechnic University.
Email: {dongchengyuan,jianzhexue}@smail.nju.edu.cn, sqsjs1968@aliyun.com, w74xu@uwaterloo.ca,
and haibozhou@nju.edu.cn.

2024 IEEE/CIC International Conference on Communications in China (ICCC) | 979-8-3503-7841-2/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICCC62479.2024.10681692

***Abstract*—Traffic flow estimation (TFE) is crucial for urban intelligent traffic systems. While traditional on-road detectors are hindered by limited coverage and high costs, cloud computing and data mining of vehicular network data, such as driving speeds and GPS coordinates, present a promising and cost-effective alternative. Furthermore, minimizing data collection can significantly reduce overhead. However, limited data can lead to inaccuracies and instability in TFE. To address this, we introduce the spatial-temporal Mamba (ST-Mamba), a deep learning model combining a convolutional neural network (CNN) with a Mamba framework. ST-Mamba is designed to enhance TFE accuracy and stability by effectively capturing the spatial-temporal patterns within traffic flow. Our model aims to achieve results comparable to those from extensive data sets while only utilizing minimal data. Simulations using real-world datasets have validated our model's ability to deliver precise and stable TFE across an urban landscape based on limited data, establishing a cost-efficient solution for TFE.**

***Index Terms*—ST-Mamba, traffic flow estimation, recovery method, limited data.**

I. Introduction

Traffic flow estimation (TFE) assumes paramount importance in the operation of traffic systems, particularly in light of the escalating volume of vehicles and the expansion of transportation networks [1] [2]. The conventional methods of TFE, employing on-road detectors like cameras and inductive loops, exhibit limited coverage and substantial expenses associated with infrastructure installation and maintenance [3]. Utilizing data from vehicular network can expand the scope of TFE, while the collection and transmission of massive data also encounter technical and economic challenges. Considering the constraints of existing methods, it is cost-effective to utilize limited data from vehicular network to obtain real-time TFE by cloud computing [4] [5]. However, when the amount of traffic data decreases, the resulting TFE tends to be inaccurate and unstable [6]. With the advancement of artificial intelligence, employing deep learning methods to recover TFE from limited data emerges as a feasible and promising approach.

Lately, deep learning methods have been developed to explore features of traffic flow [7] [8]. The convolutional neural network (CNN) is proficient in extracting spatial features of input through convolutional operations [9]. The recurrent neural network (RNN) and its variants such as long short-term memory (LSTM) are designed to capture sequential or temporal features of input by utilizing recurrent connections [10]. Mamba, an advanced neural network based on state space model (SSM), is effective at processing sequential data and modeling temporal correlation via selection mechanism [11]. Moreover, some methods combine the functions of CNN and RNN. For instance, the Conv-LSTM module is proposed to forecast short-term traffic flow via learning spatial-temporal dependencies [12]. However, existing works primarily focus on traffic flow prediction rather than estimating real-time traffic flow, and are provided with sufficient data instead of limited data. To enhance estimation accuracy with limited data, there is a need to design a specialized spatial-temporal model for traffic flow estimation recovery.

In this paper, we investigate the TFE utilizing limited data and propose a spatial-temporal deep learning method to recover both the accuracy and stability of TFE. The framework of TFE with limited vehicular network data is illustrated in Fig. 1. We divide the city map evenly into grids, each of which represent a specific region in the city. The data limitation in our scheme refers to the practice of randomly sampling a subset of vehicles across the city with equal probability of recruitment at each time interval [6]. Following the collection of limited data, the traffic information comprising vehicle speeds and GPS coordinates are matched to corresponding grids. The condition of traffic flow within each grid is represented by the average speed of vehicles over a defined period. For the recovery of TFE, we propose a novel deep learning model, named as spatial-temporal Mamba (ST-Mamba), to generate stable and accurate TFE via effectively leveraging the spatial-temporal features of traffic flow. Specifically, we employ the CNN to capture the local spatial correlation of traffic flow, and utilize the Mamba to model the temporal correlation of each grid in the map. The main contributions of this paper are listed as the following:

- We present a novel approach for cost-effective TFE, utilizing only a small fraction of vehicular network data to alleviate the burden on network communication and mitigate the expenses of traffic systems.
- We design a spatial-temporal deep learning model named as ST-Mamba, consisting of CNN and Mamba, to recover the TFE with limited vehicular network data.

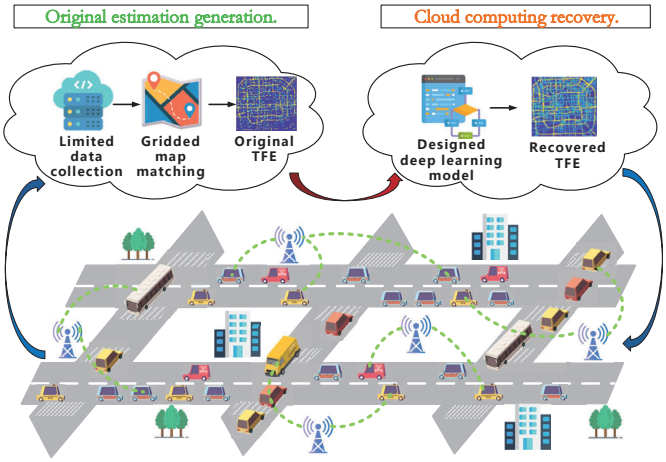

Fig. 1: An illustration of TFE with limited data.

- We validate the effectiveness of the proposed method through comprehensive simulations on the real-world dataset, varying in data limitation from 10% to 50%. The results demonstrate that our method can provide accurate and stable city-wide TFE.

The rest of this paper is organized as follows. Section II demonstrates the framework and problem analysis with limited vehicular network data. The structure of ST-Mamba is detailed in Section III. Then, section IV showcases the simulation performance of our proposed model. At last, we conclude this paper in Section V.

II. Framework and Problem Analysis

This section consists of two parts. First, the framework for TFE with limited network data is delineated. Then, the problem of recovering estimation from limited network data is formulated.

A. Framework for TFE with Limited Data

The process of recovering TFE from limited data is illustrated in Fig. 1. First, we divide the city map evenly into grids. Meanwhile, the limited data of vehicle mobility information is gathered from vehicular network by traffic data collector. Instead of removing data from some specific regions, we acquire the limited data by randomly selecting a small fraction of vehicles as data sources uniformly distributed across the city, with equal probability of recruitment for each vehicle. Following the limited data collection, we design a matching process between the data and the grid-structured city map. Then, the original estimation is acquired via computing the average speeds of vehicles in each grid to construct the traffic flow image. Lastly, the recovery algorithm is applied to generate precise TFE based on the limited data.

As to the specific scheme of our work, we first obtain the city map consisting of grids with equal size and number of $H*W$. Corresponding to reality, each grid in the map represents a particular region of the city. We construct the grid-structured city map as,

$$\mathrm{M} = \begin{bmatrix} \mathrm{m}^{1,1} & \cdots & \mathrm{m}^{1,W} \\ \vdots & \mathrm{m}^{\mathrm{h,w}} & \vdots \\ \mathrm{m}^{H,1} & \cdots & \mathrm{m}^{H,W} \end{bmatrix}, \tag{1}$$

where $\mathrm{m}^{h,w}$ indicates the grid of map which is located at (h, w), H and W can be viewed as the height and width of the map.

With the construction of grid-structured city map, the vehicular network data are matched to respective grids based on the GPS coordinates of vehicles. We first obtain the vehicle mobility dataset V_t with sufficient data, which is collected at time slot t under ideal circumstances. Subsequently, with the grid matching outcomes of V_t and M, the ideal estimation of traffic flow is calculated as,

$$z_t^{h,w} = \mathrm{Mean}(v^{h,w} | v^{h,w} \in V_t), \tag{2}$$

$$\boldsymbol{Z}_t = \begin{bmatrix} z_t^{1,1} & \cdots & z_t^{1,W} \\ \vdots & z_t^{\mathrm{h,w}} & \vdots \\ z_t^{H,1} & \cdots & z_t^{H,W} \end{bmatrix}, \tag{3}$$

where the mobility information of vehicles in the grid located at (h, w) is described as $v^{h,w}$, $z_t^{h,w}$ is the average speed of vehicles obtained from sufficient data in the grid located at (h, w) at time slot t, Z_t represents the ideal estimation of traffic flow at time slot t.

Then, the acquisition of TFE from limited data is conducted as follows. We denote the limited vehicle mobility information gathered at time slot t as $V_t^{'}$, which is obtained through random sampling from the sufficient dataset V_t, i.e., $V_t^{'} \subseteq V_t$. This sampling approach assigns equal selection probability to each piece of vehicle mobility information. With a similar calculating and matching process, the original estimation of traffic flow at time slot t is denoted as,

$$x_t^{h,w} = \mathrm{Mean}(v^{h,w} | v^{h,w} \in V_t^{'}), \tag{4}$$

$$\boldsymbol{X}_t = \begin{bmatrix} x_t^{1,1} & \cdots & x_t^{1,W} \\ \vdots & x_t^{\mathrm{h,w}} & \vdots \\ x_t^{H,1} & \cdots & x_t^{H,W} \end{bmatrix}, \tag{5}$$

where $x_t^{h,w}$ is the average speed of vehicles obtained from limited data in the grid located at (h, w) at time slot t, X_t represents the city-wide original estimation of traffic flow at time slot t. Note that both X_t and Z_t can be perceived as images with a resolution of $H * W$ and four channels, where the amount of channels is decided by the types of driving directions defined in our work, i.e., east, south, west, and north.

Finally, we apply the specifically designed deep learning model to recover the original estimation X_t. The ideal estimation Z_t not only serves as the ground truth label for the learning of the model, but also is the target data that the model expects to output. Following the training and inference of the model, we obtain the recovered TFE from limited data, which is reliable and cost-effective.

B. Problem of Traffic Flow Estimation

Based on the limited data, we can obtain the original estimation of traffic flow approximating the ideal one. However, the original estimation exhibits a considerable degree of inaccuracy and instability regardless of the similarity. In other words, we found that there exits abundant noise in each grid of original TFE $\boldsymbol{X}_t$, which fails to serve as the definitive estimation for the applications of urban transportation systems. Therefore, the task of real-time TFE with limited network data is conceptualized as a grid-structured data recovery problem.

We set the input of the recovery model to be the historical and current estimations of traffic flow derived from limited data, represented as $[\boldsymbol{X}_{t-L+1}, ..., \boldsymbol{X}_{t-1}, \boldsymbol{X}_t]$. The objective is to reform an estimation of current traffic flow with high accuracy and stability, mirroring the ideal estimation $\boldsymbol{Z}_t$ derived from sufficient data. Therefore, the crux of the TFE problem lies in ascertaining the mapping function $\mathcal{F}(\cdot)$,

$$\boldsymbol{Z}_t = \mathcal{F}([\boldsymbol{X}_{t-L+1}, ..., \boldsymbol{X}_{t-1}, \boldsymbol{X}_t]), \tag{6}$$

which aims to fully explore the spatial-temporal dependencies of traffic flow, mitigate the noise resulting from the data limitation, and transform the original TFE derived from limited data into a precise estimation of current traffic flow.

III. ST-Mamba for Traffic Flow Recovery

In this section, we present the ST-Mamba designed to recover accurate and stable TFE from limited network data by exploring and aggregating the features of traffic flow. The network structures employed for capturing the spatial and temporal correlations by the CNN and Mamba are discussed. Subsequently, we provide an overview of the architecture belonging to the ST-Mamba.

A. Spatial Correlation Modeling

In our framework, the estimation of traffic flow can be viewed as an image with a resolution of $H * W$ and four channels. The spatial correlation of traffic flow is inherent in the grid structure. Given that the condition of traffic flow in one region is impacted by those in adjacent regions while the influence wanes with increasing spatial distance, the aggregation of spatial information from neighboring regions is imperative for accurate TFE. CNN is a classic and concise neural network suitable for processing grid-structured data, the principle of which is convolving input data with learnable filters to detect local patterns [9]. In recognition of the features of traffic flow, we employ CNN for capturing local spatial correlation of traffic flow from the entire traffic flow image.

We first conduct convolutional operations by CNN separately on four channels of the image, capturing spatial correlation of each driving direction. The working process of the CNN in our model can be denoted as,

$$X_n^{'} = \mathrm{CNN_e}(X_n), \tag{7}$$

where subscript n ranges from $t-L+1$ to t, $X_n \in \mathbb{R}^{4\times H\times W}$ is the original TFE derived from limited data at time slot n,

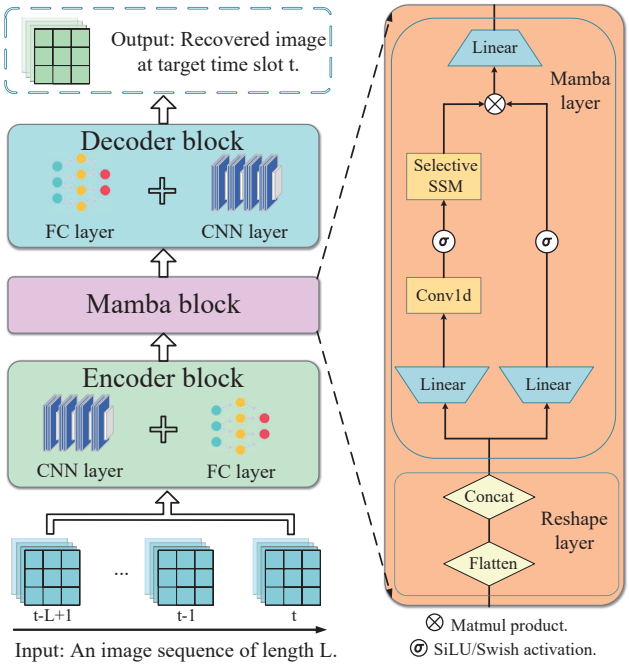

Fig. 2: ST-Mamba framework.

$X_n^{'} \in \mathbb{R}^{4\times H\times W}$ contains the spatial correlation captured by CNN. Then, we utilize a fully connected (FC) layer to lift data dimension and integrate spatial correlations of different driving directions. The FC layer is operated as,

$$X_n^{''} = \mathrm{FC_e}(X_n^{'}), \tag{8}$$

where $X_n^{''} \in \mathbb{R}^{K\times H\times W}$ is the high-dimensional feature representation of $X_n^{'}$ in latent space. Subsequently, one reshape layer is constructed to serialize the format of traffic data. The sequential traffic data of each grid can be obtained as,

$$I^{h,w} = \mathrm{Concat}(x_n^{h,w} | n = t-L+1, ...t-1, t), \tag{9}$$

where $\mathrm{Concat}(\cdot)$ is the concatenating operation, x_n is the spatial feature representation in one grid of $X_n^{''}$, $I \in \mathbb{R}^{K\times L}$ is the sequence comprising spatial correlation of traffic flow in one grid of $X_n^{''}$.

B. Temporal Correlation Modeling

The temporal correlation of traffic flow is hidden in the sequential structure. The original TFE at time slot t, combined with those at preceding $(L-1)$ time slots, are utilized to construct sequential traffic data for recovering TFE at current time slot t. Mamba is an advanced neural network especially effective at processing sequential data and modeling temporal correlation. Integrating the selection mechanism which associates the parameters of SSM with the input, Mamba is able to focus on relevant inputs and ignore irrelevant ones, executing linear-time processing of sequences while maintaining high performance [13]. It is appropriate to employ Mamba for learning the complex temporal correlation of traffic flow.

We aim to explore temporal correlation in each grid of the original TFE, which directs us to set sequence $I \in \mathbb{R}^{K\times L}$ as the input of Mamba. As the pre-foundation of Mamba, we first define the continuous-time SSM to describe the state representation of the sequence at each time step and predict the next state based on the input, with $\mathbf{A}$ as the evolution parameter and $\mathbf{B}$, $\mathbf{C}$ as the projection parameters. The continuous-time SSM is formulated as,

$$H^{'}(t) = \mathbf{A}H(t) + \mathbf{B}I(t), \tag{10}$$

$$O(t) = \mathbf{C}H(t), \tag{11}$$

where $H(t)$ is the implicit hidden state at time slot t, $O(t)$ is the output of the SSM and is obtained via the projection of the latest latent state.

Then, we construct Mamba with four matrix format parameters $\mathbf{\Delta}$, $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, which is the discrete version of the continuous-time SSM. The transformation method for discretization in Mamba is zero-order hold, which can be written as,

$$\overline{\mathbf{A}} = \exp(\mathbf{\Delta A}), \tag{12}$$

$$\overline{\mathbf{B}} = (\mathbf{\Delta A})^{-1}(\exp(\mathbf{\Delta A}) - \mathbf{E}) \cdot \mathbf{\Delta B}, \tag{13}$$

where $(\cdot)^{-1}$ represents the inverse matrix operation, $\mathbf{E}$ stands for the identity matrix, $\mathbf{\Delta}$ is the timescale parameter targeted at transforming the continuous parameters $\mathbf{A}$, $\mathbf{B}$ to discrete ones $\overline{\mathbf{A}}$, $\overline{\mathbf{B}}$.

Subsequently, we realize the selection mechanism of Mamba by configuring the weight matrix $\mathbf{B}$, $\mathbf{C}$, $\mathbf{\Delta}$ to be input-dependent, which filters out useless information, compresses input sequence selectively into the hidden state, and handles contextual information effectively. Th selection mechanism is formulated as,

$$\mathbf{B} = \mathrm{Linear}_N(I), \tag{14}$$

$$\mathbf{C} = \mathrm{Linear}_N(I), \tag{15}$$

$$\mathbf{\Delta} = \mathrm{softplus}(\mathrm{Broadcast}_K(\mathrm{Linear}_1(I)) + \mathbf{D}), \tag{16}$$

where Linear_d is a parameterized projection to dimension d, N is the dimension of hidden state, $\mathrm{Broadcast}_K$ is targeted at broadcasting one-dimensional data into a K-dimensional space, $\mathbf{D}$ is a constant weight matrix with dimension K, $\mathrm{softplus}$ ensures numerical stability as an activation function. Note that $\mathbf{A}$ comprises constant parameters for simplicity, while $\overline{\mathbf{A}}$ is input-dependent via the process of discretization by $\mathbf{\Delta}$. Meanwhile, although $\mathbf{B}$ and $\mathbf{C}$ are computed in a similar way, they are independent sets of parameters and represent different dynamics in Mamba.

Following the discretization of parameters and the implementation of selection mechanism, we conduct the computation of global convolution to obtain the output of Mamba as,

$$\overline{\mathbf{S}} = (\mathbf{C}\overline{\mathbf{B}}, \mathbf{C}\overline{\mathbf{A}\mathbf{B}}, ..., \mathbf{C}\overline{\mathbf{A}}^{L-1}\overline{\mathbf{B}}), \tag{17}$$

$$O = I * \overline{\mathbf{S}}, \tag{18}$$

where $\overline{\mathbf{S}}$ is a structured convolutional kernel for implementing the selective scan algorithm to speed up the model, $O \in \mathbb{R}^{K\times L}$ is the final output of Mamba which contains spatial and temporal correlations of traffic flow in each grid.

C. Outcome Decoding

The final outcome is decoded from the output of Mamba. The decoder block consists of one FC layer and one CNN layer. In the FC layer, we combine the sequential traffic data O to form an image format with dimension $\mathbb{R}^{(K\times L)\times H\times W}$. Meanwhile, we reduce data dimension to aggregate spatial-temporal correlations previously modeled. The FC layer in decoder block is operated as,

$$G = \mathrm{FC_d}(O), \tag{19}$$

where $G \in \mathbb{R}^{4\times H\times W}$ can be viewed as the image of TFE containing spatial and temporal correlations of traffic flow. Then, we conduct convolutional operations in the CNN layer to decode the outcome as,

$$\boldsymbol{Y} = \mathrm{CNN_d}(G), \tag{20}$$

where $\boldsymbol{Y} \in \mathbb{R}^{4\times H\times W}$ is the final outcome of the model and can be perceived as the recovered TFE approximating the ideal estimation $\boldsymbol{Z}_t$.

D. ST-Mamba Model

The main architecture of ST-Mamba model is illustrated in Fig. 2. We specifically design three parts for ST-Mamba, which are the encoder block, the Mamba block, and the decoder block, respectively.

In specific, the encoder block consists of one CNN layer and one FC layer in turn. Given that vehicles driving in different directions have minimal influence on each other, four CNNs with identical parameters are employed in the CNN layer to independently perform convolutional operations in each channel of input, extracting spatial correlation of traffic flow under four distinct driving directions. Then, we utilize one FC layer to lift the dimension of input data to explore much more detailed information and exchange spatial features of different driving directions. For the Mamba block, we stack one reshape layer before the Mamba layer, serializing the input so that Mamba can process the limited data of traffic flow effectively. Then, the generated sequences are fed into the Mamba layer in parallel to explore the temporal correlation in each gird of the original traffic flow estimations. For the decoder block which is similar to encoder one, the role of the FC layer is to aggregate spatial-temporal correlations captured previously and project the processed data into the image format. Then, convolutional operations are done by the CNN layer to decode the final outcome while form a formal symmetry of the model. Finally, ST-Mamba exports the recovered traffic flow estimation at time slot t from limited data, with high accuracy and stability which can meet the demands of traffic transportation systems.

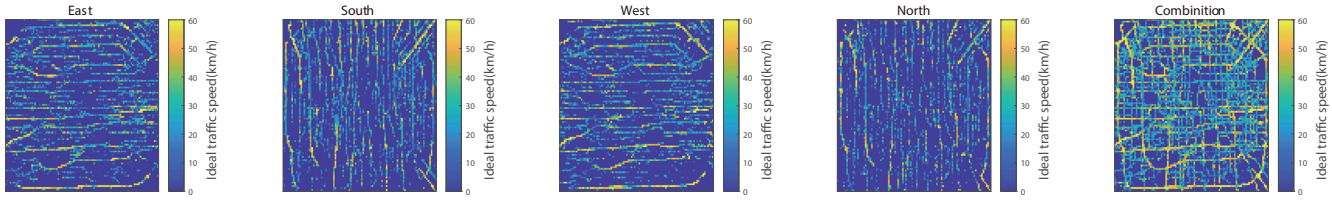

Fig. 3: The ideal TFE on Friday 5:00 PM.

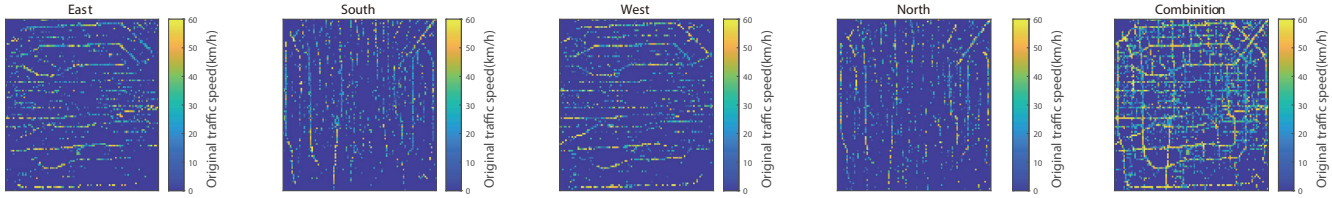

Fig. 4: The original TFE on Friday 5:00 PM at 10% limitation.

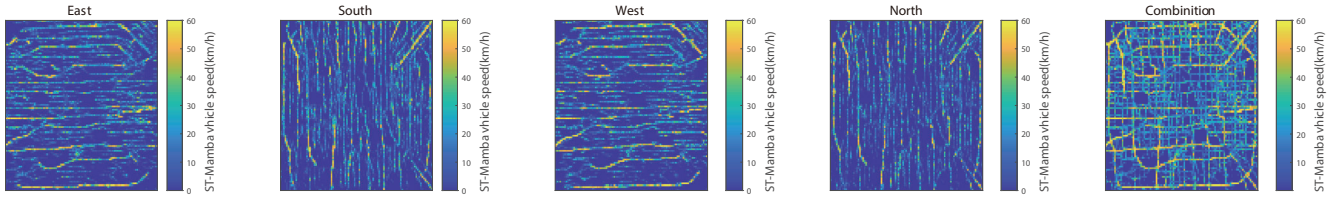

Fig. 5: The recovered TFE on Friday 5:00 PM at 10% limitation.

IV. Simulation and Results

In this section, we assess our method utilizing the real traffic data from Beijing. We begin by the experimental settings comprising the dataset, baselines, and evaluation metrics. Then, we present and analyze the estimation performance.

A. Experimental Setting

The simulation utilizes the real traffic data sourced from the Fourth Ring Road in Beijing. Our dataset includes 6 days in 2012, and the time period for TFE spans from 7:30 AM to 10:30 PM [6]. The resolution of TFE image is 100×100. The limited data is obtained by random sampling from sufficient data, where the limitation degree indicates the proportion of sampled data. Our approach is compared with the following four baselines:

- Original: The original estimation derived from limited data, and the input of deep learning models.
- CNN: A convolutional neural network designed to capture spatial correlation by convolutional operations [9].
- PredRNN: A recurrent neural network proposed to capture spatial-temporal dependencies for coherent future frame forecasting [14].
- SimVP: A simple yet powerful architecture for video prediction, which makes the most of CNN and its variants to explore spatial-temporal correlations [15].

Meanwhile, we select three metrics to evaluate the performance of models. Root mean squared error (RMSE) assesses the recovery accuracy by penalizing larger errors more heavily. Improved percentage (IP) demonstrates the recovery

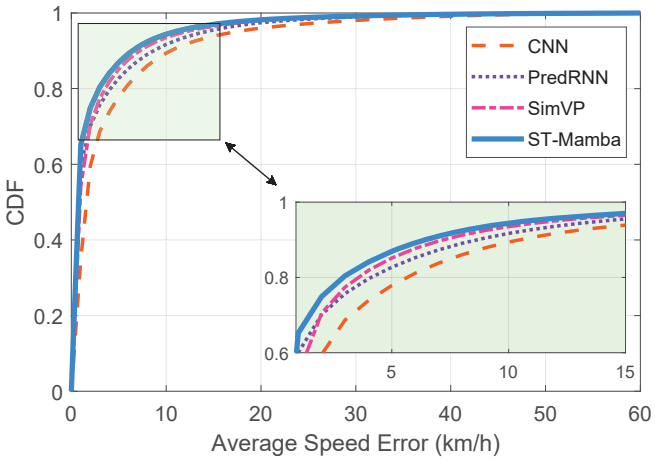

Fig. 6: The CDFs of estimation error on Sunday.

improvement brought by the model. Mean absolute error (MAE) calculates the average absolute difference between the recovered and ideal estimation.

B. Performance Analysis

Table I presents a comparative analysis of recovery performance by the ST-Mamba against other baselines. Models are evaluated across five degrees of data limitation from 10% to 50%. ST-Mamba demonstrates superior recovery capability compared to the other models, consistently attaining the lowest RMSE and MAE, alongside the highest IP. These results indicate that the traffic flow estimations recovered by ST-

TABLE I: The estimation recovery comparison.

Limitation	Model	Metric		
		RMSE	IP	MAE
10%	Original	13.269	*	5.379
	CNN	9.955	24.979%	5.359
	PredRNN	8.067	39.207%	3.965
	SimVP	8.224	40.864%	4.561
	ST-Mamba	**7.504**	**43.449%**	**3.531**
20%	Original	11.072	*	4.073
	CNN	8.786	20.638%	5.124
	PredRNN	6.966	37.082%	3.231
	SimVP	7.049	38.415%	3.680
	ST-Mamba	**6.379**	**42.386%**	**2.851**
30%	Original	9.481	*	3.204
	CNN	7.957	16.065%	4.067
	PredRNN	6.443	32.043%	2.843
	SimVP	6.126	37.042%	3.001
	ST-Mamba	**5.516**	**41.813%**	**2.245**
40%	Original	8.187	*	2.547
	CNN	7.134	12.854%	3.529
	PredRNN	5.967	27.118%	2.615
	SimVP	5.341	36.183%	2.537
	ST-Mamba	**4.857**	**40.670%**	**1.876**
50%	Original	7.056	*	2.015
	CNN	6.631	6.008%	3.077
	PredRNN	5.817	17.549%	2.472
	SimVP	4.703	34.664%	1.989
	ST-Mamba	**4.288**	**39.220%**	**1.540**

Mamba have the least error. In addition, considering the significant errors associated with original TFE, the errors of ST-Mamba are deemed acceptable for practical applications.

As to the recovery performance within space domain, Fig. 3, Fig. 4, and Fig. 5 show the TFE under different conditions. Fig. 3 is the ideal TFE, calculating the average vehicle speeds from 100% data and clearly demonstrating the framework of Fourth Ring Road. Fig. 4 is the original TFE at the data limitation degree of 10%, where considerable number of data missing and errors can be witnessed. Fig. 5 is the recovered TFE obtained by ST-Mamba and is similar to the ideal one in terms of both grid color and image structure, proving excellent recovery performance of ST-Mamba in space domain.

The recovery performance within time domain is illustrated in Fig. 6, where we display the cumulative distribution function (CDF) of the estimation error per minute from 7:30 to 22:30 on Sunday at limitation degree of 30%. It can be observed the CDF of ST-Mamba converges the fastest, which means the errors of ST-Mamba are concentrated on small values and the error range is relatively narrow compared to baselines. In other words, ST-Mamba is able to provide reliable recovery for TFE throughout a day.

In general, ST-Mamba is able to capture spatial-temporal correlations of traffic flow effectively, providing accurate and stable TFE for the entire city on various days.

V. Conclusion

In this paper, we studied the framework for TFE utilizing limited data from the vehicular network. We analyzed the TFE recovery problem caused by the limitation of traffic data availability. To address this issue, we designed a spatial-temporal deep learning model named as ST-Mamba to improve the estimation accuracy and stability. Comprehensive simulation results based on the real-world dataset have validated the effectiveness of ST-Mamba in processing sequential grid-structured data. Our approach provides a cost-effective solution for TFE in scenarios with limited data. For future work, we intend to explore the traffic flow prediction based on limited data.

Acknowledgment

This work is supported in part by the National Natural Science Foundation of China under Grant 623B2052, 62271244, the Natural Science Fund for Distinguished Young Scholars of Jiangsu Province under Grant BK20220067.

References

[1] Y. Zhang, Q. Cheng, Y. Liu, and Z. Liu, "Full-Scale Spatio-Temporal Traffic Flow Estimation for City-Wide Networks: A Transfer Learning Based Approach," *Transportmetrica B: Transport Dynamics*, vol. 11, no. 1, pp. 869–895, 2023.

[2] P. Wang, J. Lai, Z. Huang, Q. Tan, and T. Lin, "Estimating Traffic Flow in Large Road Networks Based on Multi-Source Traffic Data," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 9, pp. 5672–5683, 2020.

[3] N. Caceres, L. M. Romero, F. G. Benitez, and J. M. del Castillo, "Traffic Flow Estimation Models Using Cellular Phone Data," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 3, pp. 1430–1441, 2012.

[4] A. Abadi, T. Rajabioun, and P. A. Ioannou, "Traffic Flow Prediction for Road Transportation Networks With Limited Traffic Data," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 653–662, 2015.

[5] C. Zha, J. Xue, Q. Shen, and H. Zhou, "Low-Cost Traffic Perception for Road Detector Data Estimation: A Deep Learning Approach," in *the Proceedings of International Conference on Wireless Communications and Signal Processing (WCSP)*. IEEE, 2022, pp. 1–5.

[6] J. Xue, Y. Xu, W. Wu, T. Zhang, Q. Shen, H. Zhou, and W. Zhuang, "Sparse Mobile Crowdsensing for Cost-Effective Traffic State Estimation With Spatio-Temporal Transformer Graph Neural Network," *IEEE Internet of Things Journal*, vol. 11, no. 9, pp. 16 227–16 242, 2024.

[7] A. Miglani and N. Kumar, "Deep Learning Models for Traffic Flow Prediction in Autonomous Vehicles: A Review, Solutions, and Challenges," *Vehicular Communications*, vol. 20, p. 100184, 2019.

[8] J. Xue, K. Yu, T. Zhang, H. Zhou, L. Zhao, and X. Shen, "Cooperative Deep Reinforcement Learning Enabled Power Allocation for Packet Duplication URLLC in Multi-Connectivity Vehicular Networks," *IEEE Transactions on Mobile Computing*, vol. 23, no. 8, pp. 8143–8157, 2024.

[9] L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamaría, M. A. Fadhel, M. Al-Amidie, and L. Farhan, "Review of Deep Learning: Concepts, CNN Architectures, Challenges, Applications, Future Directions," *Journal of Big Data*, vol. 8, pp. 1–74, 2021.

[10] A. Sherstinsky, "Fundamentals of Recurrent Neural Network (rnn) and Long Short-Term Memory (LSTM) Network," *Physica D: Nonlinear Phenomena*, vol. 404, p. 132306, 2020.

[11] L. Li, H. Wang, W. Zhang, and A. Coster, "STG-Mamba: Spatial-Temporal Graph Learning via Selective State Space Model," *arXiv preprint arXiv:2403.12418*, 2024.

[12] Y. Liu, H. Zheng, X. Feng, and Z. Chen, "Short-Term Traffic Flow Prediction with Conv-LSTM," in *the Proceedings of International Conference on Wireless Communications and Signal Processing (WCSP)*, 2017, pp. 1–6.

[13] A. Gu and T. Dao, "Mamba: Linear-Time Sequence Modeling with Selective State Spaces," *arXiv preprint arXiv:2312.00752*, 2023.

[14] Y. Wang, H. Wu, J. Zhang, Z. Gao, J. Wang, S. Y. Philip, and M. Long, "PredRNN: A Recurrent Neural Network for Spatiotemporal Predictive Learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 2, pp. 2208–2225, 2022.

[15] Z. Gao, C. Tan, L. Wu, and S. Z. Li, "SimVP: Simpler yet Better Video Prediction," in *the Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, 2022, pp. 3170–3180.