MT-STNet: A Novel Multi-Task Spatiotemporal Network for Highway Traffic Flow Prediction

Guojian Zou, *Member, IEEE*, Ziliang Lai, Ting Wang, Zongshi Liu, and Ye Li

***Abstract*— Multi-step highway traffic flow prediction is crucial for intelligent transportation systems, and existing works have made significant advancements in this field. However, the physical structure, including path, distance, and node degree, is critical information in traffic networks and is often overlooked when encoding spatiotemporal dependencies. Meanwhile, the problem of prediction error propagation in multi-step flow forecasting is challenging to mitigate and can significantly impact overall forecasting performance. Moreover, traffic flows between toll and gantry stations in the highway network exhibit notable differences, leading to heterogeneous flow distributions. To overcome the above issues, a novel multi-task spatiotemporal network for highway traffic flow prediction (MT-STNet) is proposed, consisting of the encoder-decoder structure, a generative inference system, and multi-task learning. The spatiotemporal block with physical transformation is developed to construct both the encoder and decoder, integrating the physical structure information into modeling the highway network's spatiotemporal dependencies. Additionally, the generative inference architecture is designed to extract the correlation between the historical- and target- sequences to generate the target hidden representations rather than a dynamic decoding way, avoiding multi-step prediction error propagation. Furthermore, because of traffic flow heterogeneity in the highway network, multi-task learning divides highway traffic flow prediction into three tasks, sharing the underlying traffic network and knowledge learned, thereby enhancing the prediction performance of each subtask. The evaluation experiments used monitoring data from a highway in Yinchuan City, Ningxia Province, China. The experimental results demonstrate that the performance of our proposed prediction model is better than that of the baseline methods.**

***Index Terms*— Multi-step highway traffic flow prediction, spatiotemporal correlation, physical structure information, multi-task learning, graph neural network, generative inference system.**

I. Introduction

TRAFFIC flow prediction has a pivotal role in intelligent transportation systems (ITS), which is the basis of traffic management, the internet of vehicles, and many other applications. Accurate traffic flow prediction is technically challenging due to the complex spatiotemporal correlations and the topology structure of road networks [1], but it attracts many researchers' interest.

Traffic data is a type of time series structure, and in the early years, statistical methods were adopted to predict traffic flow, such as the seasonal autoregressive integrated moving average (SARIMA) [2]. These methods have been successfully applied in traffic prediction tasks; however, they face the challenge of not being able to extract the traffic data that has inherent limitations, including uncertainty, nonlinearity, and complexity. As researchers experiment with new methods, traditional machine learning algorithms have gradually appeared in traffic prediction tasks, such as k-nearest neighbor (KNN) [3], support vector regression (SVR) [4], and hidden markov models (HMM) [5]. Traditional machine learning methods break through the limitation of statistics-based methods to a certain extent, effectively handling nonlinear traffic data and achieving good prediction results. However, these methods rely on complex mathematical algorithms and prior knowledge to obtain shallow features, and it is impossible to extract deep, complex spatiotemporal correlations.

Deep learning methods, automatically processing high-dimensional data and extracting complex spatiotemporal correlations, offer novel insights to accomplish traffic flow prediction tasks [6]; such methods include LSTM_BILSTM [7], DELA [8], and ConvLSTM [9]. However, traffic network is a topological structure, and traffic data conform to a discrete distribution, that is, they have non-Euclidean structure. The methods that have been developed have difficulty modeling the spatiotemporal correlation of non-Euclidean structural data, and it is not easy to improve the accuracy of multi-step traffic flow prediction. Recently, graph neural networks (GNNs) have been widely used for traffic prediction [10], [11], [12], since they can provide a novel paradigm to modeling non-Euclidean structure data. Despite significant progress in both spatiotemporal correlation extraction and prediction accuracy, the following challenges still exist,

Spatial modeling has been limited by methods that only consider static spatial dependency or dynamic spatial correlation due to a need for physical structure information on the traffic network [13]. In practice, spatial modeling is

AQ:1 Manuscript received 1 May 2022; revised 17 October 2023 and 21 April 2024; accepted 5 June 2024. This work was supported in part by the Scholarship of China Scholarship Council under Grant 202306260111, in part by the Project of the National Key Research and Development Program of China under Grant 2018YFB1601301 and Grant 2023YFC3305802, and in part by the National Natural Science Foundation of China under Grant 71961137006. The Associate Editor for this article was Y. Chen. *(Corresponding authors: Guojian Zou; Ye Li.)*

Guojian Zou is with the Key Laboratory of Road and Traffic Engineering, Ministry of Education, and the College of Transportation Engineering, Tongji University, Shanghai 201804, China, and also with the Department of Geography, University of Zurich, 8057 Zürich, Switzerland (e-mail: 2010768@tongji.edu.cn).

Ziliang Lai, Ting Wang, Zongshi Liu, and Ye Li are with the Key Laboratory of Road and Traffic Engineering, Ministry of Education, and the College of Transportation Engineering, Tongji University, Shanghai 201804, China (e-mail: 2033402@tongji.edu.cn; 2110763@tongji.edu.cn; chuochuoliu@tongji.edu.cn; JamesLI@tongji.edu.cn).

Digital Object Identifier 10.1109/TITS.2024.3411638

usually associated with the traffic network's physical structure, including the in and out-degree of the station, the shortest path between stations, and the shortest path distance between stations. For instance, when several routes are provided from the exact origin station to the destination, drivers prefer to choose the shortest one, increasing the traffic flow passing stations in the selected route. Therefore, an effective way to adaptive fuse the physical structure information with spatial correlation extraction methods (e.g., GNNs [10], [12], [13]) is needed.

Regarding temporal modeling, recurrent neural networks (RNNs) are extensively employed in performing dynamic step-by-step encoding and decoding. However, fixed sequential extraction patterns lead to error propagation and temporal dependency molding insufficient [7]. Subsequently, attention mechanism networks emerge as the times required used to extract temporal correlation [9], especially transformer-based models that have achieved satisfactory predictive performance [11], [12], [14]. Although the transformer architecture effectively captures temporal dependency, like RNNs are used in dynamic decoding and cause error accumulation in inference propagation [15], [16]. Hence, modeling temporal dependence precisely and avoiding error accumulation in multi-step flow prediction is imperative.

For the highway network, there is tremendous traffic flow heterogeneity with three root causes: (1) the difference in speed limitation between ramps and main roads, (2) trip purpose, and (3) the types of roads, especially ramps and main roads. According to these properties, highway traffic flow forecasting is divided into three categories: traffic flow passing the entrance toll, gantry, and exit toll, regarded as multi-task learning [17]. In addition, the highway network is a closed-loop graph, and traffic flow passing the entrance and exit toll stations is directly related to the flow passing the gantries in the main rods. Multi-task learning assignments each subtask an independent nonlinear transformation function, sharing the underlying traffic network and knowledge learned, enhancing the prediction performance of each subtask [17], [18].

Therefore, we propose a novel multi-task spatiotemporal network (MT-STNet) for traffic flow prediction to overcome the challenges mentioned above. The contributions of this paper are summarized as follows:

1) A new spatiotemporal module, ST-Physical Block, is proposed as a primary component of the encoder and decoder. Dynamic traffic and highway network architecture properties are two critical aspects for traffic flow forecasting, which are molded by ST-Physical Block simultaneously and then adaptively fused without additional parameters. Moreover, such as node degrees, to our knowledge, ST-Physical Block is the first to incorporate this physical structure information in modeling spatiotemporal dependencies.
2) To avoid prediction error propagation, a specialized generative-style inference system is introduced, which bridges the historical and future spatiotemporal presentations to generate multi-step traffic flow in a single step rather than through dynamic decoding. This technique thereby prevents error accumulation during inference and significantly enhances inference speed.
3) Multi-task learning addresses the unbalanced traffic flow distribution in the highway network for flow prediction across three categories. Each subtask shares the underlying traffic network and knowledge learned to distinguish the traffic flow heterogeneity via different task layers, improving prediction accuracy.
4) The experimental results demonstrate that the proposed MT-STNet outperforms all baselines on the real-world highway dataset, especially for multi-step and high-traffic flow forecasting. In addition, detailed analyses are conducted to examine the functions and contributions of the model's designed components.

The rest of this paper is organized as follows. In Section II, we summarize the previous related studies. Section III describes the relative definitions. Section IV describes the proposed network for traffic flow in detail. Section V presents the experiments and the results in detail. Finally, we conclude this paper and discuss future work in Section VI.

II. Related Work

The existing traffic flow prediction approaches can be divided into two categories: classical methods and deep learning methods.

Classical methods include data-driven statistical approaches and traditional machine learning algorithms. Statistical approaches, including vector autoregression (VAR) [19], and the Kalman filter [20], ARIMA [21], and its variants are the most representative time series modeling algorithms in traffic flow prediction tasks [2], [22]. For example, Williams et al. [21] deemed traffic forecasting a univariate time series prediction problem and applied SARIMA in the station with seasonal patterns, capturing both seasonal and trend components. Due to time series stability assumptions and data record integrity constraints, and because they ignore the inherent nonlinear characteristics of traffic data, those methods cannot provide satisfactory prediction results [6].

Subsequently, traditional machine learning algorithms have been used in traffic flow prediction [23], including KNNs [3], HMMs [5], gradient boosting decision trees (GBDT) [24], SVR [4], and Bayesian networks [25], due to their ability to process multi-dimensional data and capture complex nonlinear correlations. For instance, Castro-Neto et al. [4] developed an application of a supervised learning model based on support vector regression (SVR) for the prediction of short-term freeway traffic flow under distinct conditions. The existing studies have proved that traffic flow prediction is not only affected by the temporal dimension, but also by the spatial dimension [26]. However, these traditional machine learning methods are limited to extracting shallow features and cannot model the deep, complex spatiotemporal correlations of traffic data.

Recently, the potential of deep learning methods in traffic flow prediction tasks has been developed gradually. The deep learning methods automatically extract input features and output the final prediction values; examples of the feedforward neural network (FNN) [27], deep belief network (DBN) [28],

and efficient hinging hyperplanes neural network (EHHNN) [29]. Since traffic flow prediction is a typical time series prediction problem, it is essential to extract the temporal correlation. Therefore, RNN-based time series prediction models have been widely used in traffic flow prediction to capture temporal dependency [30]. For example, Ma et al. [7] proposed an LSTM_BILSTM model that consists of a long short-term memory network (LSTM) and a bidirectional long short-term memory network (BILSTM) to model the temporal correlation. Moreover, some researchers use the Transformer technique [31] and its variants to model dynamic temporal correlation in traffic prediction [32]. However, if one only needs to process the temporal correlation of traffic data, the impact of the spatial dependency on the prediction may be ignored.

Aiming at this problem, most researchers have applied convolutional neural networks (CNNs) to capture spatial correlations in traffic networks [33], [34]. For example, Cheng et al. [35] and Zheng et al. [8] proposed a CNN-LSTM hybrid neural network model that takes full advantage of the spatiotemporal features of the traffic flow data. Zheng et al. [9] developed an attention-based Conv-LSTM module to extract the spatial and short-term temporal features using the CNN, LSTM, and attention networks. However, traffic data conforms to a discrete distribution, that is, it has non-Euclidean structure. The common problem with these methods is that they extract the spatial features using the CNNs, but traditional CNNs deal with this issue in the context of Euclidean space, and therefore they are not suitable for non-Euclidean structural data.

To address this limitation, graph neural networks (GNNs) have better performance in extracting the spatial features of non-Euclidean structural data than the architecture based on CNNs [36], [37], [39]. Li et al. [40] propose a diffusion convolutional recurrent neural network (DCRNN) to model the traffic diffusion process on the directed graph, consisting of diffusion convolution and the gated recurrent unit (GRU), which incorporates spatiotemporal dependencies in the traffic flow. Lv et al. [41] proposed a temporal multi-graph convolutional network (T-MGCN) that jointly models the spatial, temporal, and semantic correlations with various global features in the traffic network for traffic flow prediction. Zhao et al. [10] proposed a novel traffic flow prediction method, the temporal graph convolutional network (T-GCN) model based on the graph convolutional network (GCN) and GRU, to extract the spatial and temporal correlations. However, the spatial correlation of the traffic network changes dynamically over time [11], [12]. Most of these methods focus solely on local static spatial dependencies, which lack consideration of the dynamic spatial correlation of the traffic network.

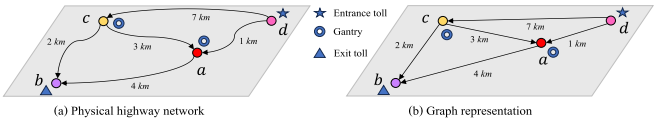
Therefore, adaptive dependency matrices (ADMs) without relying on pre-defined graphs and graph attention networks (GATs) are utilized to model the spatial dependency of traffic networks. For ADMs, Bai et al. [42] proposed an adaptive graph convolutional recurrent network (AGCRN) to effectively forecast traffic via node adaptive parameter learning and data-adaptive graph generation modules without relying on pre-defined graphs. Wu et al. [43] introduced a novel graph neural network architecture called Graph-WaveNet, which develops a novel adaptive dependency matrix learned through node embedding to capture hidden spatial dependencies effectively. Wu et al. [44] introduced a framework, referred to as MTGNN, that consists of three components, the graph learning layer, the graph convolution module, and the temporal convolution module, to address the issue of multivariate time series lacking an explicit graph structure. Yu et al. [45] proposed a regularized graph structure learning (RGSL) model consisting of two innovative modules: one component, named Regularized Graph Generation, is used to learn the sparse graph structure, and the other, called Laplacian Matrix Mixed-up Module, is proposed to fuse the explicit and implicit graphs.

Graph attention networks, which rely on excellent performance in dynamic spatial correlation modeling, are widely used in traffic prediction [46], [47]. Guo et al. [39] proposed an attention-based spatiotemporal graph convolutional network (ASTGCN) model precisely, using spatiotemporal attention to capture the dynamic spatiotemporal dependencies and employing graph convolutions to capture the spatial patterns and standard convolutions to describe the temporal features. Wang et al. [48] proposed a novel spatial-temporal graph neural network (STGNN) for traffic flow prediction, which offers a learnable positional attention mechanism to aggregate information from adjacent roads effectively and provides a sequential component to model the temporal correlation. Zheng et al. [12] proposed a graph multi-attention network (GMAN) to predict multi-step traffic flow at different locations on the traffic network and design transform attention to alleviate the error propagation among prediction time steps. Park et al. [49] proposed a novel spatiotemporal graph attention (ST-GRAT) based on the Transformer [15] technique that captures the spatiotemporal dynamic correlations in the road network.

However, these methods lack consideration of simultaneously static- and dynamic spatial correlations, neglecting complete physical structure information in spatiotemporal dependency modeling. In addition, error accumulation in inference propagation (including spatial and temporal dimensions) is inevitable with such methods. Moreover, flow heterogeneity in the traffic network is ignored, and the positive impact of the related tasks on the traffic flow prediction is not considered. In this paper, inspired by recent studies on GNNs, a novel multi-task spatiotemporal network for highway traffic flow prediction, referred to as MT-STNet, is proposed.

III. Preliminary

The highway is a connected topological network in the physical space, as shown in Fig. 1 (a). We need to map the highway network in the physical space to the logical space that the computer can understand in order to complete the necessary preliminary work of modeling traffic data. The mapping work can be interpreted as abstracting the intelligent sensors of electronic toll collection (ETC) stations and gantries as nodes in the graph, and abstracting road segments as edges connecting nodes, as shown in Fig. 1 (b). We define the input graph as $G = (V, E, A)$, where V represents the set of nodes, E represents the set of directed edges, and $A = A^{-} \cup A^{+}$



(a) Physical highway network (b) Graph representation

Fig. 1. **(a)** For a highway network with four monitoring stations, there are significant differences in the length of the road segments between the stations. **(b)** Each color node represents a monitoring station in a different geographic location, and each directed edge represents a road segment.

represents the adjacency matrix, $\mathrm{A}^{-} \in \mathrm{R}^{\mathrm{N}\times\mathrm{N}}$ represents the in-degree matrix, $\mathrm{A}^{+} \in \mathrm{R}^{\mathrm{N}\times\mathrm{N}}$ represents the out-degree matrix, and N represents the number of nodes. When the matrix element $\mathrm{A}_{v_i,v}$ is 'one', there is a direct connection edge between node v_i and node v. On the contrary, if $\mathrm{A}_{v_i,v}$ is 'zero', there is no direct connection edge between node v_i and node v.

MT-STNet aims to predict multi-step highway traffic flow, including highway segment flow, toll station entrance flow, and toll station exit flow. Therefore, the research goal of this paper can be defined as a multi-task prediction. Assume the input time steps are P and the predicted time steps are Q. Given the historical sequence of observations $\mathrm{X} = \{\mathrm{X}_{t_1}, \cdots, \mathrm{X}_{t_\mathrm{P}}\} \in \mathrm{R}^{\mathrm{P}\times\mathrm{N}\times1}$ of N nodes in P time steps, we aim to predict the target sequence values of Q time steps for N nodes, expressed as $\hat{\mathrm{Y}} = \{\hat{\mathrm{Y}}_{t_{\mathrm{P}+1}}, \cdots, \hat{\mathrm{Y}}_{t_{\mathrm{P}+\mathrm{Q}}}\} \in \mathrm{R}^{\mathrm{Q}\times\mathrm{N}\times1}$, where '1' represents the feature dimension of the observed or predicted values.

IV. Proposed Approach

A. Framework Overview

Figure. 2 represents the framework of our proposed MT-STNet model, consisting of the encoder-decoder structure, a generative inference system, and multi-task learning. First, the encoder and decoder are constructed by multiple layers of spatiotemporal block with physical transformation (ST-Physical Block) consisting of temporal attention, spatial attention, and a multi-head graph convolutional network (Multi-head GCN), which is used to model the highway network's spatiotemporal dependencies. Second, the generative inference system is developed to bridge a connection between the historical and the target sequences to generate the target hidden representations, rather than dynamic decoding, to avoid multi-step prediction error propagation. Finally, multi-task learning is proposed to divide highway traffic flow prediction into three tasks, share the underlying traffic network and knowledge learned, and improve the prediction performance of each subtask. Each part of the proposed method will be detailed later.

B. Embedding Layer

This paper proposes that four types of variables must be prepared for model implementation: initial node representation, node degree, directed edge, and direct distance. As shown in Fig. 3 right, the node representation $\mathrm{X}_t \in \mathrm{R}^{\mathrm{N}\times d}$ at time step t is obtained by adding three primary pieces of information, including traffic flow $\mathrm{XF}_t \in \mathrm{R}^{\mathrm{N}\times d}$, timestamp $\mathrm{XT}_t \in \mathrm{R}^{\mathrm{N}\times2\times d}$, and station $\mathrm{XS}_t \in \mathrm{R}^{\mathrm{N}\times d}$ embeddings; $\mathrm{X}_t = \mathrm{XF}_t + \mathrm{STE}_t$ and $\mathrm{STE}_t = sum\,(\mathrm{XT}_t, \mathrm{XS}_t)$. The traffic flow embedding method

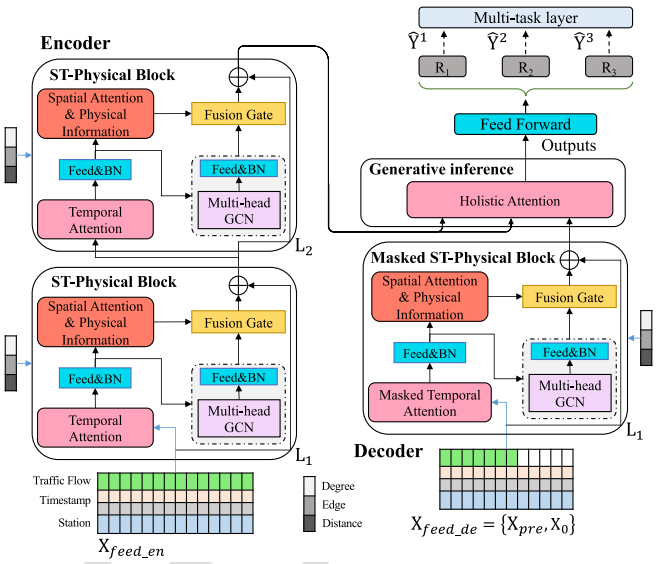


Fig. 2. The framework of the multi-task spatiotemporal network (MT-STNet). (a) L layers of ST-Physical Block can be stacked to extract spatiotemporal dependency, with each layer taking inputs derived from the layer below it. (b) The input historical sequence of observations X contains three types of embeddings: traffic flow embedding, timestamp embedding, and station embedding.

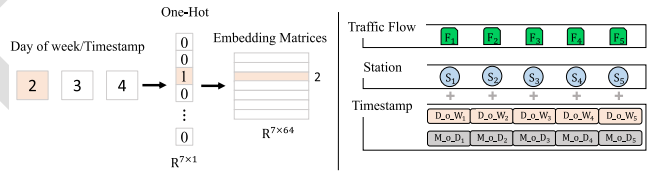


Fig. 3. **Left**: Example of the day of week embedding. **Right**: The node representation consists of three types of primary information: traffic flow embedding, station embedding, and timestamp embedding (including day of week and minute of day, i.e., D_o_W and M_o_D).

realizes the process of mapping low-dimensional traffic values to high-dimensional vectors through two nonlinear transformation layers. Timestamp- and station embedding methods map discrete timestamp and station variables to dense matrices via one-hot vectors [13], [50]. The embedding process can be divided into three steps, for example, day of week: seven days a week, allocating an index from zero to six each day, as shown in Fig. 3 left; second, the index is transferred into a one-hot code. For instance, day two is transferred into a one-hot vector, the value of the corresponding index in the one-hot is set to one, and vice versa; finally, map the one-hot to dense embedding matrices via multiplication, and the matrices are updated adaptively in the training phase [50].

Similar to timestamp embedding, two types of real-valued embeddings (i.e., $\mathcal{Z} \in \mathrm{R}^{\mathrm{N}\times d}$ and $\mathcal{U} \in \mathrm{R}^{|\mathrm{E}|\times d}$) can represent node degrees and directed edges between nodes. For example, the in-degree embedding vector of monitoring station $v \in \mathrm{V}$ is $z^{-}_{deg^{-}(v)} \in \mathrm{R}^{d}$, and the out-degree embedding vector of monitoring station v is $z^{+}_{deg^{+}(v)} \in \mathrm{R}^{d}$. For any directed edge $\vec{e}$, its embedding vector is $u_{\vec{e}} \in \mathrm{R}^{d}$.

The fourth distance variable $\mathcal{S} \in \mathrm{R}^{|\mathrm{E}|\times1}$, the direct distance $s_{v_i,v_j} \in \mathrm{R}$ between monitoring stations $v_i \in \mathrm{V}$ and $v_j \in \mathrm{V}$, is calculated directly from the latitude and longitude of the two stations in the traffic network. If there is no connection

between monitoring stations v_i and v_j, the direct distance s_{v_i,v_j} is infinity $+\infty$.

C. ST-Physical Block

Highway networks have three aspect characteristics in the spatial and temporal dimensions: static- and dynamic spatial dependencies and temporal correlation. As shown in Fig. 2, we utilize the spatial attention and multi-head GCN in the ST-Physical Block to model the spatial dependency, and use temporal attention to extract the temporal correlation. Assume that the output of ST-Physical Block is HST $\in \mathrm{R}^{\mathcal{T}\times \mathrm{N}\times d}$, in which the hidden representation of monitoring station v at time step t ($t = t_1, \ldots, t_{\mathcal{T}}$) is $hst_{v,t} \in \mathrm{R}^d$. The outputs of temporal attention, spatial attention, and multi-head GCN methods in the l^{th} layer are $\mathrm{HDT}^l \in R^{\mathcal{T}\times \mathrm{N}\times d}$, $\mathrm{HDS}^l \in R^{\mathcal{T}\times \mathrm{N}\times d}$ and $\mathrm{HSS}^l \in R^{\mathcal{T}\times \mathrm{N}\times d}$, respectively, while the dynamic temporal correlation, dynamic spatial dependency, and static spatial correlation of monitoring station v at time step t are $hdt^l_{v,t} \in \mathrm{R}^d$, $hds^l_{v,t} \in \mathrm{R}^d$, and $hss^l_{v,t} \in \mathrm{R}^d$, where $\mathcal{T}$ denotes the input series length of ST-Physical Block.

Since this study uses a nonlinear transformation function at high frequencies, it is defined as:

$$f(x) = \mathrm{ReLU}(x\mathrm{W} + b) \tag{1}$$

where x represents the input variables, W and b denote the learnable parameters, and ReLU is the nonlinear activation function.

1) Temporal Attention: The traffic flow at each monitoring station in the traffic network is affected by previous observations and impacts the future; however, relationships between different time steps are subject to change, referred to as dynamic temporal correlation. For example, the traffic flow during the morning peak is affected by the continuous traveling vehicles in the morning, which still negatively impacts the traffic flow after the morning peak. In this study, we design a temporal attention approach to adaptively model the correlation between different time steps and use the multi-head attention mechanism to calculate the correlation coefficient, as shown in Fig. 4.

For monitoring station v, at time step t_i, the correlation coefficient between time steps t_i and t_j in the l^{th} temporal attention layer is,

$$\alpha^{l,m}_{t_i,t_j} = \frac{exp\left(\mu^{l,m}_{t_i,t_j}\right)}{\sum_{r=1}^{\mathcal{T}} exp\left(\mu^{l,m}_{t_i,t_r}\right)} \tag{2}$$

where $\mu^{l,m}_{t_i,t_j}$ denotes the relevance between t_i and t_j, and $\mathcal{T}$ represents the input time series length of ST-Physical Block.

The relevance can be obtained by the inner product of the query vector of monitoring station v at time step t_i and the key vector of monitoring station v at time step t_j,

$$\mu^{l,m}_{t_i,t_j} = \frac{\left\langle f^m_q\left(hdt^{l-1}_{v,t_i} + z_{deg(v)}\right), f^m_k\left(hdt^{l-1}_{v,t_j} + z_{deg(v)}\right)\right\rangle}{\sqrt{d}} \tag{3}$$

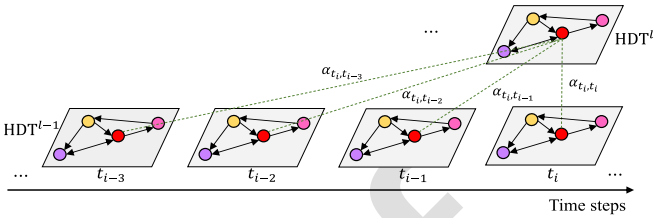


Fig. 4. Temporal attention model of the temporal correlations between different time steps.

where f^m_q and f^m_k represent the nonlinear transformation functions in the m^{th} head attention of the query vector and the key vector, $\langle *, * \rangle$ represents the inner product operator, and monitoring station v degree embedding $z_{deg(v)} = z^-_{deg^-(v)} + z^+_{deg^+(v)}$.

Once the correlation coefficient $\alpha^{l,m}_{t_i,t_j}$ in the m^{th} head attention is obtained, the l^{th} layer temporal correlation $hdt^l_{v,t_i} \in \mathrm{R}^d$ of station v at time step t_i can be formulated as,

$$hdt^{l,m}_{v,t_i} = \sum_{r=1}^{\mathcal{T}} \alpha^{l,m}_{t_i,t_r} \cdot f^m_v\left(hdt^{l-1}_{v,t_r} + z_{deg(v)}\right) \tag{4}$$

$$hdt^l_{v,t_i} = \mathrm{BN}\left(\|^{\mathrm{M}}_{m=1}\ hdt^{l,m}_{v,t_i}\mathrm{W}^{l,m}_{hdt}\right) \tag{5}$$

where f^m_v represents the nonlinear transformation function in the m^{th} head attention of the value vector, and $\|$ represents the concatenation operation; BN represents batch normalization. The final dynamic temporal correlation $hdt_{v,t_i} \in \mathrm{R}^d$ of monitoring station v can be calculated using Equations (1)-(5) at time step t_i, and the output of temporal attention is HDT $\in \mathrm{R}^{\mathcal{T}\times \mathrm{N}\times d}$.

2) Spatial Attention: The traffic flow of a monitoring station in the traffic network is affected by other monitoring stations, and the influence weight changes dynamically with time. For example, a congestion station is influenced by first-order neighbors or second-order and more in the highway network during the congestion period, and the influence weights between target and neighbors change over time. This property is defined as dynamic spatial correlation, and we design a spatial attention approach to adaptively model the correlations between different stations of the traffic network. In addition, we also consider the positive effects of the inherent physical information of the traffic network on the calculation of dynamic spatial correlations, including station in- and out-degree, shortest path, and shortest path distance. For instance, the station in- and out-degree and the shortest path distance between the monitoring stations directly affect the traffic flow of the station on the shortest path [51]. To model these properties, we propose the multi-head attention mechanism with physical information to calculate the correlation coefficient, as shown in Fig. 5.

For monitoring station node v_i, at time step t, the correlation coefficient between nodes v_i and v_j in the l^{th} spatial attention layer is,

$$\beta^{l,m}_{v_i,v_j} = \frac{exp\left(\varphi^{l,m}_{v_i,v_j}\right)}{\sum_{v\in \mathrm{V}} exp\left(\varphi^{l,m}_{v_i,v}\right)} \tag{6}$$

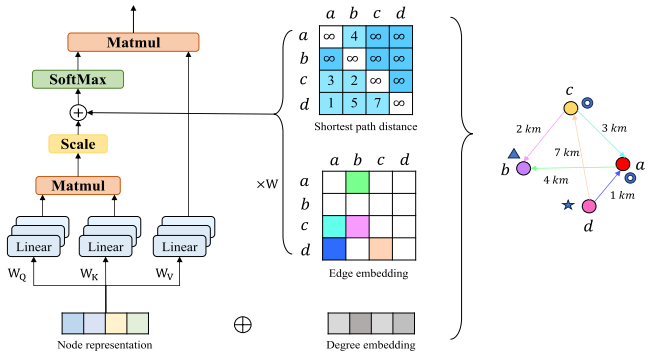


Fig. 5. An illustration of our designed spatial attention mechanism. **Left**: Multi-head attention mechanism. **Right**: Physical transformation.

where $\varphi^{l,m}_{v_i,v_j}$ denotes the relevance between v_i and v_j, V represents the input monitoring stations of ST-Physical Block.

In the traffic network, the traffic flow of the monitoring station is often affected by the in- and out-degree; however, this critical information is often overlooked in the existing research. Similar to temporal attention, we feed the in-degree embedding $z^-_{deg^-(v_r)}$ and out-degree embedding $z^+_{deg^+(v_r)}$ of $v_r \in \mathrm{V}$ into the spatial attention network as additional information of the input. Therefore, the relevance can be obtained by the inner product of the query vector of station v_i and the key vector of station v_j at time step t,

$$\varphi^{l,m}_{v_i,v_j} = \frac{\left\langle f^m_q\left(hds^{l-1}_{v_i,t} + z_{deg(v_i)}\right), f^m_k\left(hds^{l-1}_{v_j,t} + z_{deg(v_j)}\right)\right\rangle}{\sqrt{d}} \tag{7}$$

where f^m_q and f^m_k represent the nonlinear transformation functions in the m^{th} head attention of the query vector and the key vector, and $z_{deg(v_r)} = z^-_{deg^-(v_r)} + z^+_{deg^+(v_r)}$.

The relevance between monitoring stations is affected by the shortest path distance and the shortest path, defined as d_{v_i,v_j} and $\mathrm{SP}_{v_i,v_j} = (\vec{e}_1, \vec{e}_2, \cdots, \vec{e}_{\mathrm{K}})$, where $\mathrm{K} = |\mathrm{E}|$. To model the shortest path distance property, we assign a pre-computed constant scalar d_{v_i,v_j} to the relevance of stations v_i and v_j, which will be used as the bias term in the multi-head attention mechanism, as shown in Equation (8). Furthermore, for a reachable directed station pair (v_i, v_j), we find the shortest path SP_{v_i,v_j} from station v_i to v_j and compute the mean of the dot products of edge features and learnable embeddings along the path. We incorporate the encoded shortest path feature as a bias term into the attention module, as shown in Equations (8) and (9).

$$\varphi^{l,m}_{v_i,v_j} = \varphi^{l,m}_{v_i,v_j} + \frac{1}{d_{v_i,v_j}} + c^m_{v_i,v_j} \tag{8}$$

with

$$c^m_{v_i,v_j} = \frac{1}{\mathrm{K}}\sum_{r=1}^{\mathrm{K}} u^m_{\vec{e}_r}\left(w^m_r\right)^{\mathsf{T}} \tag{9}$$

where d_{v_i,v_j} and SP_{v_i,v_j} can be calculated by the Dijkstra algorithm based on direct distances and directed edges [52], and $w^m_r \in \mathrm{R}^{1\times d/\mathrm{K}}$ denotes a learnable embedding in the m^{th} head attention. Assuming that the shortest path distance between stations v_i and v_j is by adding the valid direct distances contained in the shortest path, SP_{v_i,v_j} is completed with the $\vec{e}_0$; if there is no path between stations v_i and v_j, we set $d_{v_i,v_j} = +\infty$, and SP_{v_i,v_j} is completed with the $\vec{e}_0$. The embedding of edge $\vec{e}_0$ can be represented by a $\vec{0}$ vector.

Once obtaining the correlation coefficient $\beta^{l,m}_{v_i,v_j}$ in the m^{th} head attention, the l^{th} layer dynamic spatial correlation $hds^l_{v_i,t}$ of monitoring station v_i at time step t can be formulated as,

$$hds^{l,m}_{v_i,t} = \sum_{v\in V}\beta^{l,m}_{v_i,v}\cdot f^m_v\left(hds^{l-1}_{v,t} + z_{deg(v)}\right) \tag{10}$$

$$hds^l_{v_i,t} = \mathrm{BN}\left(\|^{\mathrm{M}}_{m=1} hds^{l,m}_{v_i,t}\mathrm{W}^{l,m}_{hds}\right) \tag{11}$$

where f^m_v represents the nonlinear transformation function in the m^{th} head attention of the value vector. The final dynamic spatial correlation $hds_{v_i,t} \in \mathrm{R}^d$ of station v_i can be calculated using Equations (6)-(11) at time step t. The initial inputs of spatial attention are $\mathrm{HDT} \in \mathrm{R}^{\mathcal{T}\times\mathrm{N}\times d}$, $\mathcal{Z} \in \mathrm{R}^{\mathrm{N}\times d}$, $\mathcal{U} \in \mathrm{R}^{|\mathrm{E}|\times d}$, and $\mathcal{S} \in \mathrm{R}^{|\mathrm{E}|\times 1}$, and the output is $\mathrm{HDS} \in \mathrm{R}^{\mathcal{T}\times\mathrm{N}\times d}$.

3) Multi-Head GCN: The traffic network is a directed graph, so the study of traffic flow should consider the connection relationship between graph nodes, that is, the source and destination of the traffic flow at the monitoring station. For example, at a monitoring station (e.g., red station in Fig. 1) in the traffic network, the upstream traffic flow comes from two different monitoring stations directly connected to it, and the downstream traffic flow goes out to one monitoring station that is directly connected to it. This property is defined as a static spatial dependency, and we design a multi-head GCN to focus on the traffic inflow and outflow from different subspaces of different highway stations. The static spatial correlation aggregation is used as the input to a standard nonlinear transformation layer, in order to generate the l^{th} layer embedding of the graph nodes, as shown in Equation (12). For monitoring station v_i, at time step t, the correlation between stations v_i and v_i's first-order neighbors V_{v_i} in the m^{th} head GCN is,

$$hss^{l,m}_{v_i,t} = f\left(\widetilde{\mathrm{D}}^{-0.5}\widetilde{\mathrm{A}}_{v_i}\widetilde{\mathrm{D}}^{-0.5}\mathrm{HSS}^{l-1,m}_t\mathrm{W}^{l-1,m}\right) \tag{12}$$

$$hss^l_{v_i,t} = \mathrm{BN}\left(\|^{\mathrm{M}}_{m=1} hss^{l,m}_{v_i,t}\mathrm{W}^{l,m}_{hss}\right) + hss^{l-1,m}_{v_i,t} \tag{13}$$

where $\widetilde{\mathrm{D}}^{-0.5}\widetilde{\mathrm{A}}_{v_i}\widetilde{\mathrm{D}}^{-0.5}$ denotes the normalized adjacency matrix of the graph with added self-connections, $\widetilde{\mathrm{A}}_{v_i} = \mathrm{A}_{v_i} + \mathrm{I}_{v_i}$ is the adjacency matrix of the node v_i with added self-connection, I represents the identity matrix, and $\widetilde{\mathrm{D}}$ is the degree matrix of $\widetilde{\mathrm{A}}$. Multi-head GCN can be built by stacking multi-convolutional layers in parallel; as Equations (12) and (13), the topological relationship between nodes v_i and v_i's first-order neighbors V_{v_i} can be obtained, and the topological structure of the traffic network and the attributes of the monitoring station are encoded in order to obtain the static spatial dependency. The final static spatial dependency $hss_{v_i,t}$ of monitoring station v_i can be calculated using Equations (12)-(13) at time step t. The initial input of the multi-head GCN is $\mathrm{HDT} \in \mathrm{R}^{\mathcal{T}\times\mathrm{N}\times d}$, and the output is $\mathrm{HSS} \in \mathrm{R}^{\mathcal{T}\times\mathrm{N}\times d}$.

4) Feature Fusion Gate: To obtain the final spatiotemporal correlation $\mathrm{HST} \in \mathrm{R}^{\mathcal{T}\times\mathrm{N}\times d}$, we use the fusion method to adaptively fuse the output $\mathrm{HDS} \in \mathrm{R}^{\mathcal{T}\times\mathrm{N}\times d}$ of spatial attention with the output $\mathrm{HSS} \in \mathrm{R}^{\mathcal{T}\times\mathrm{N}\times d}$ of the multi-head GCN. HDS

and HSS are fused as,

$$\mathrm{HST}^{l} = \mathcal{W} \odot \mathrm{HDS}^{l} + (1 - \mathcal{W}) \odot \mathrm{HSS}^{l} + \mathrm{HST}^{l-1} \tag{14}$$

with

$$\mathcal{W} = \sigma\left(\mathrm{HDS}^{l} \odot \mathrm{HSS}^{l}\right) \tag{15}$$

where σ represents *sigmoid* activation, $\odot$ denotes Hadamard product, and $\mathcal{W} \in \mathrm{R}^{\mathrm{T}\times\mathrm{N}\times d}$ is weight vector that controls the flow of dynamic and static spatial representations at each time step.

D. Encoder-Decoder

1) Encoder: Give a historical input sequence $\mathrm{HDT}^{0} = \mathrm{X}_{feed\_en} \in \mathrm{R}^{\mathrm{P}\times\mathrm{N}\times d}$, $\mathcal{Z} \in \mathrm{R}^{\mathrm{N}\times d}$, $\mathcal{U} \in \mathrm{R}^{|\mathrm{E}|\times d}$, and $\mathcal{S} \in \mathrm{R}^{|\mathrm{E}|\times 1}$, and the encoder based on multi-layer ST-Physical Block to extract the spatiotemporal correlation $\mathrm{HST}' \in \mathrm{R}^{\mathrm{P}\times\mathrm{N}\times d}$ we have introduced the working principle of the ST-Physical Block, and HST' will be used in the decoder to predict target traffic flow.

2) Decoder: We designed a particular decoder structure in Fig. 2, composed of the multi-layer masked ST-Physical Block. In the case where the target traffic flow is not known, we need to model the spatiotemporal dependencies of target input sequence. Therefore, the masked ST-Physical Block is applied in the decoder to model spatiotemporal correlations in order to initialize the target input sequence X_0. This masking method prevents future target sequences from attending previous time steps. We feed the $\mathcal{Z} \in \mathrm{R}^{\mathrm{N}\times d}$, $\mathcal{U} \in \mathrm{R}^{|\mathrm{E}|\times d}$, $\mathcal{S} \in \mathrm{R}^{|\mathrm{E}|\times 1}$, and following embeddings to the decoder,

$$\mathrm{X}_{feed\_de} = Concate\left(\mathrm{X}_{pre}, \mathrm{X}_0\right) \in \mathrm{R}^{(\mathrm{T}+\mathrm{Q})\times\mathrm{N}\times d} \tag{16}$$

where $\mathrm{X}_{pre} \in \mathrm{R}^{\mathrm{T}\times\mathrm{N}\times d}$ is the part of the encoder historical input sequence $\mathrm{X}_{feed\_en} \in \mathrm{R}^{\mathrm{P}\times\mathrm{N}\times d}$ from time step $t_{\mathrm{P}-\mathrm{T}}$ to t_{P}. As Fig. 2 shows, we take the last 7 known time steps before the target sequence, and feed the decoder with $\mathrm{X}_{feed\_de} = Concat\left(\mathrm{X}_{feed\_en}[-7:], \mathrm{X}_0\right)$; $\mathrm{X}_0 \in \mathrm{R}^{\mathrm{Q}\times\mathrm{N}\times d}$ is the target input sequence (setting the traffic flow embedding to 0), and X_0 contains the target input sequence's timestamp embedding and station embedding. After the masked ST-Physical Block of the decoder, the initialized $\mathrm{HST}'' \in \mathrm{R}^{\mathrm{Q}\times\mathrm{N}\times d}$ is obtained from feed inputs X_{feed_de}, $\mathcal{Z}$, $\mathcal{U}$, and $\mathcal{S}$.

E. Generative Inference

For the predictive inference process, we need to consider the propagation of errors and the inference speed. In this paper, we design a particular generative inference system based on the named holistic attention method that directly combines the correlations between historical- and target time steps to generate the final target hidden representations $\mathrm{HGI} \in \mathrm{R}^{\mathrm{Q}\times\mathrm{N}\times d}$. For example, to infer the output $\mathrm{HGI}_{t_{\mathrm{P}+i}} \in \mathrm{R}^{\mathrm{N}\times d}$ at time step $t_{\mathrm{P}+i}$, we calculate it through a holistic attention layer based on the historical sequence $\mathrm{HST}' \in \mathrm{R}^{\mathrm{P}\times\mathrm{N}\times d}$ and the target initialized sequence $\mathrm{HST}'' \in \mathrm{R}^{\mathrm{Q}\times\mathrm{N}\times d}$, and $\mathrm{HST}''' = \mathrm{HST}' \cup \mathrm{HST}''$; this does away with the time-consuming "dynamic decoding" operation, as shown in Fig. 6.

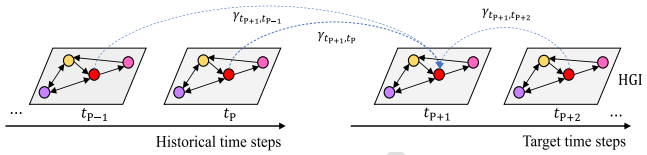


Fig. 6. Holistic attention models the correlations between historical time steps and target time steps directly.

For monitoring station v, the intercorrelation coefficient between the target time step $t_{\mathrm{P}+i}$ $(t_{\mathrm{P}+i} = t_{\mathrm{P}+1}, \cdots, t_{\mathrm{P}+\mathrm{Q}})$ and the input time step t_j $(t_j = t_1, \cdots, t_{\mathrm{P}}, \cdots, t_{\mathrm{P}+\mathrm{Q}})$ is measured,

$$\gamma^{m}_{t_{\mathrm{P}+i},t_j} = \frac{exp\left(\lambda^{m}_{t_{\mathrm{P}+i},t_j}\right)}{\sum_{r=1}^{\mathrm{P}+\mathrm{Q}} exp\left(\lambda^{m}_{t_{\mathrm{P}+i},t_r}\right)} \tag{17}$$

$$\lambda^{m}_{t_{\mathrm{P}+i},t_j} = \frac{\left\langle f^{m}_{q}\left(hst''_{v,t_{\mathrm{P}+i}} + ste_{v,t_{\mathrm{P}+i}}\right), f^{m}_{k}\left(hst'''_{v,t_j} + ste_{v,t_j}\right)\right\rangle}{\sqrt{d}} \tag{18}$$

where $\lambda^{m}_{t_{P+i},t_j}$ denotes the relevance between $t_{\mathrm{P}+i}$ and t_j, and $\mathrm{P}+\mathrm{Q}$ represents the historical and target sequence length.

Once the correlation coefficient $\gamma^{m}_{t_{\mathrm{P}+i},t_j}$ in the m^{th} head attention is obtained, the hidden representation $hgi^{l}_{v,t_{\mathrm{P}+i}}$ of station v at time step $t_{\mathrm{P}+i}$ can be formulated as,

$$hgi^{m}_{v,t_{\mathrm{P}+j}} = \sum_{r=1}^{\mathrm{P}+\mathrm{Q}} \gamma^{m}_{t_{\mathrm{P}+i},t_r} \cdot f^{m}_{v}\left(hgi_{v,t_r} + ste_{v,t_r}\right) \tag{19}$$

$$hgi_{v,t_{\mathrm{P}+i}} = \mathrm{BN}\left(\|^{\mathrm{M}}_{m=1}\, hgi^{m}_{v,t_{\mathrm{P}+i}} \mathrm{W}^{m}_{hgi}\right) \tag{20}$$

The final hidden representation $hgi_{v,t_{\mathrm{P}+i}}$ of monitoring station v can be calculated using Equations (17)-(20) at time step $t_{\mathrm{P}+i}$. The initial input of holistic attention is $\mathrm{HST}' \in \mathrm{R}^{\mathrm{P}\times\mathrm{N}\times d}$, $\mathrm{HST}'' \in \mathrm{R}^{\mathrm{Q}\times\mathrm{N}\times d}$, and $\mathrm{STE} \in \mathrm{R}^{(\mathrm{P}+\mathrm{Q})\times\mathrm{N}\times d}$, and the output of holistic attention is $\mathrm{HGI} \in \mathrm{R}^{\mathrm{Q}\times\mathrm{N}\times d}$.

F. Multi-Task Layer

We use generative-style inference to obtain the final target hidden representations $\mathrm{HGI} \in \mathrm{R}^{\mathrm{Q}\times\mathrm{N}\times d}$, and feed the hidden outputs to different subtask layers to generate target prediction values,

$$\hat{\mathrm{Y}} = \begin{cases} \hat{\mathrm{Y}}^{1} = \mathrm{HGI} \bullet \mathrm{W}_1 \\ \hat{\mathrm{Y}}^{2} = \mathrm{HGI} \bullet \mathrm{W}_2, \hat{\mathrm{Y}} \in \mathrm{R}^{\mathrm{Q}\times\mathrm{N}\times 1} \\ \hat{\mathrm{Y}}^{3} = \mathrm{HGI} \bullet \mathrm{W}_3 \end{cases} \tag{21}$$

where $\mathrm{W}_1 \in \mathrm{R}^{d\times 1}$, $\mathrm{W}_2 \in \mathrm{R}^{d\times 1}$, and $\mathrm{W}_3 \in \mathrm{R}^{d\times 1}$ represent the weight matrices of the three different fully connected layers, and $\bullet$ is a matrix multiplication operation.

The loss function of MT-STNet corresponding to the multi-task layer is defined as the mean absolute error (MAE) between observed values Y and predicted values $\hat{\mathrm{Y}}$,

$$\mathrm{L}(\theta) = \frac{1}{\mathrm{Q}\times\mathrm{N}} \sum_{j=1}^{\mathrm{Q}} \sum_{i=1}^{3} \left|\mathrm{Y}^{i}_{j} - \hat{\mathrm{Y}}^{i}_{j}\right| + \frac{\lambda}{2}\|\theta\|^2 \tag{22}$$

where θ denotes all the learnable parameters in MT-STNet, and λ is the regularization parameter.

V. Experiments

A. Data Description

This study uses real-world monitoring datasets from the highway network, including the gantry, entrance toll, and exit toll sources from the ETC intelligent monitoring sensors at the gantries and the toll stations of the highway in Yinchuan, Ningxia Province, China, as shown in Fig. 7. The 66 ETC intelligent monitoring sensors record the traffic flow in real-time, including 13 highway toll stations (each toll station contains an entrance and exit) and 20 highway gantries (each gantry has two directions, upstream and downstream). In particular, directed connectivity generally exists between the adjacent monitoring sensors, such as flow passing the entrance toll to the gantry. In addition, the degree is then defined according to the directed connectivity: the in-degree is the sum of the first-order connected upstream, and the out-degree is the sum of the first-order connected downstream. Moreover, the distance between two adjacent sensors is computed via the sensors' longitude and latitude values.

Furthermore, the highway traffic flow data includes six factors: traffic flow, timestamp, monitoring sensor position (i.e., longitude and latitude), connectivity matrices, degree of the sensor, and distance between monitoring sensors, and the time range is from June 1, 2021, to August 31, 2021. For this paper, the data from each monitoring sensor were recorded every 5 minutes to obtain the time series form of traffic flow. All these essential elements are added to our GitHub[1] page, such as sensor position, connectivity matrices, source samples, etc. In the experiment, we took 70% of the data as the training set, 10% as the validation set, and 20% as the test set.

B. Baseline Methods

The following algorithms were used as a performance baseline for comparison.

SARIMA [2], is a time series forecasting model that incorporates both non-seasonal and seasonal components to capture patterns and trends in data over time.

SVR [4], is a classic nonlinear machine learning approach used to forecast short-term traffic flow.

LSTM_BILSTM [7], which consists of long short-term memory (LSTM) and bidirectional LSTM (Bi-LSTM) networks. The LSTM network models the input time series and further learns and trains through the Bi-LSTM network to alleviate the prediction errors.

DELA [8], which consists of the embedding, convolutional neural network (CNN), and LSTM components; the embedding can capture the categorical features, the CNN is used to extract spatiotemporal correlations, and the LSTM is employed to model the temporal dependencies.

T-GCN [10], which combines the graph convolutional network (GCN) and the gated recurrent unit (GRU) to model the spatiotemporal dependencies simultaneously.

STGNN [48], which uses a learnable positional attention mechanism to aggregate neighbors' information. In addition, it provides a sequential component to model dynamic temporal dependencies.

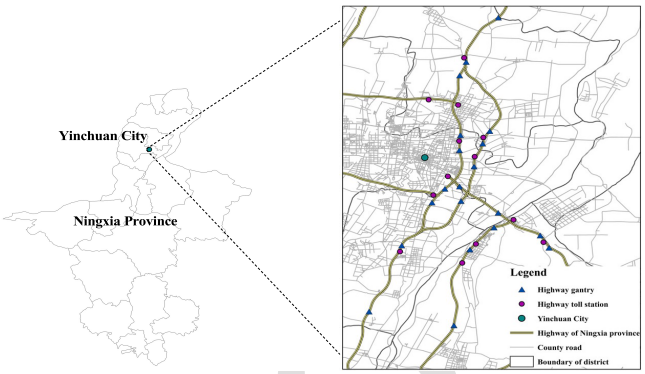


Fig. 7. Study area.

DCRNN [40], an encoder-decoder framework consisting of diffusion convolution and GRU, incorporates both spatial and temporal dependencies in the traffic flow.

AGCRN [42], a node adaptive parameter learning method, maintains a unique parameter space for each node to learn node-specific patterns. Moreover, a data-adaptive graph generation module is designed to infer the hidden inter-dependencies between variables.

ASTGCN [39], which contains the spatiotemporal attention mechanism and -convolution components. In particular, spatiotemporal attention captures the dynamic spatiotemporal dependencies, and the spatiotemporal convolution networks model spatial patterns and describe the temporal correlations. The version released here only consists of the recent component.[2]

MSTGCN [39], a variant of ASTGCN, gets rid of the spatiotemporal attention.

Graph-WaveNet [43], which proposes an adaptive dependency matrix to capture the hidden spatial dependency precisely, and several layers of dilated 1D convolution network are stacked to capture long sequence dependency.

GMAN [12], an encoder-decoder architecture, both encoder and decoder contain spatiotemporal attention to model spatiotemporal dependencies, and a transformer layer is used to connect the encoder and decoder to generate target prediction values.

ST-GRAT [49], a fusion model, especially spatial attention, is used dynamically to adjust spatial correlation between nodes, temporal attention captures long sequence dependency, and the sentinel vectors are proposed to determine whether to absorb new information from spatially correlated nodes.

MTGNN [44], a general graph neural network framework for multivariate time series data, employs a graph learning module to extract the uni-directed relations among variables, and uses a mix-hop propagation and dilated inception layers to capture the spatial and temporal dependencies.

RGSL [45], which consists of two innovative modules: an implicit dense similarity matrix derived through node embedding and the Regularized Graph Generation based on the Gumbel Softmax, is used to learn the sparse graph structure;

[1] https://github.com/zouguojian/Traffic-flow-prediction/tree/main/MT-STNet/data

[2] https://github.com/guoshnBJTU/ASTGCN-r-pytorch

a Laplacian Matrix Mixed-up Module is proposed to fuse the explicit and implicit graphs.

C. Evaluation Metrics

To evaluate the performance of our model and baselines, three metrics are used to determine the difference between the observed values Y and the predicted values $\hat{Y}$: the mean absolute error (MAE), root mean square error (RMSE), and mean absolute percentage error (MAPE).

$$\mathrm{MAE} = \frac{1}{\mathrm{D}\times\mathrm{Q}}\sum_{j=1}^{\mathrm{Q}}\sum_{i=1}^{\mathrm{D}}\left|\mathrm{Y}_{i,j}-\hat{\mathrm{Y}}_{i,j}\right| \tag{23}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{\mathrm{D}\times\mathrm{Q}}\sum_{j=1}^{\mathrm{Q}}\sum_{i=1}^{\mathrm{D}}\left(\mathrm{Y}_{i,j}-\hat{\mathrm{Y}}_{i,j}\right)^2} \tag{24}$$

$$\mathrm{MAPE} = \frac{100\%}{\mathrm{D}\times\mathrm{Q}}\sum_{j=1}^{\mathrm{Q}}\sum_{i=1}^{\mathrm{D}}\frac{\left|\mathrm{Y}_{i,j}-\hat{\mathrm{Y}}_{i,j}\right|}{\mathrm{Y}_{i,j}} \tag{25}$$

where D is the number of samples in test set. Note that low MAE, RMSE, and MAPE values indicate a more accurate prediction performance.

D. Experimental Settings

We trained our MT-STNet model and baselines on the training data set, validated on the validation set to determine the optimal hyperparameters [13], and tested on the test set. The learning process can be described as follows. On the training set, after every epoch, we calculated the value of MAE on the validation set; if the validation error decreased, the model parameters were updated in time; otherwise, the training continued. In all experiments, we used an early-stop mechanism; the number of early-stop epochs and the maximum number of epochs were set to 5 and 100, respectively. After multiple training and evaluation steps, we determined the final MT-STNet model parameters. We set the target time steps to Q = 12, representing the time span is 60 minutes, and the historical time steps P = 12. The MT-STNet model hyperparameters are shown in Table I.

Moreover, this paper provides details on the configuration of SARIMA [2]. The SARIMA framework comprises seven parameters: the order of the integrated autoregressive (AR) and moving average (MA) components, normal and seasonal differencing, seasonal AR and MA parts, and the period. The SARIMA model is constructed using Python's statsmodels package, which is commonly employed for fitting time series data. Maximum likelihood estimation (MLE) serves as the optimization algorithm to estimate the model parameters, which are then utilized to make predictions for future time points based on historical observations. In the experiments, the parameters are configured as follows: AR (p) = 4, MA (q) = 3; seasonal AR (P) = 0, seasonal MA (Q) = 1; normal differencing (d) = 0, seasonal differencing (D) = 1; and period (S) = 2016 (one week).

We implemented the MT-STNet and baselines in TensorFlow-GPU and PyTorch-GPU. The implementation **codes and hyperparameters** of our proposed MT-STNet and baselines are open-source; please refer to our personal **GitHub homepage**.[3] These models can be loaded with parameters and deployed to actual application scenarios.

TABLE I
MODEL HYPERPARAMETER

Module	Layer name	Hyperparameters	Values
Embedding Layer	Nonlinear layer	Hidden nodes Layers Activation function	64 2 ReLU
ST-Physical Block	Multi-head GCN	Hidden nodes M	64 1
	Spatial attention	Hidden nodes M	64 8
	Temporal attention	Hidden nodes M	64 8
	Residual connection	Hidden nodes Layers	64 1
Generative Inference	Holistic attention	Hidden nodes Blocks M	64 1 8
Multi-task Layer	Entrance layer	Hidden nodes Layers	[64, 1] 2
	Exit layer	Hidden nodes Layers	[64, 1] 2
	Gantry layer	Hidden nodes Layers	[64, 1] 2
-	-	Batch size	128
-	-	Dropout	0.3
-	-	λ	0.001
-	-	Decay rate	0.9
-	-	Learning rate	0.001
-	-	Epochs	100
-	-	Training method	Adam
-	-	Blocks	1

E. Experimental Results

1) Comparison With Baselines: Tables II, III, IV, and V respectively show the performance of the MT-STNet and baselines in three different tasks and the whole dataset, including entrance toll, exit toll, and gantry traffic flow prediction of the highway network for the next twelve steps. For example, 8:00-9:00 am is used for the historical time steps, and 9:00-10:00 am is considered as the target time steps.

We can see that the prediction precision of the statistical method SARIMA is less than that of machine learning methods in Tables III, IV, and V. The low prediction accuracy of the statistical approach is due to the fact that it is limited in the capability of modeling nonlinear correlations of traffic variables. However, in Table II, SARIMA outperforms SVR and LSTM_BILSTM for horizons 6 and 12 prediction. This is because some data exhibits obvious seasonal characteristics, and SARIMA may provide more accurate forecasts. In addition, temporal dependency is a critical property for traffic flow prediction, and LSTM has proved an effective technique for time series forecasting in previous studies. Nevertheless, the performance of LSTM_BILSTM is lower than that of SVR because SVR is trained separately for each monitoring station and does not have error propagation in multi-step prediction.

Spatial correlation is another factor and plays an essential role in traffic flow forecasting. The prediction precision of LSTM_BILSTM is lower than DELA, such as on the gantry

[3]https://github.com/zouguojian/Traffic-flow-prediction/tree/main/MT-STNet

TABLE II
THE PREDICTION RESULTS OF THE MT-STNET AND BASELINES ON THE ENTRANCE TOLL TASK

Model	Horizon 3			Horizon 6			Horizon 12			Average		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
SARIMA	4.402	6.334	55.376%	4.470	6.520	56.605%	4.545	6.720	57.632%	4.515	6.619	56.500%
SVR	4.214	6.522	39.194%	4.609	7.113	42.947%	5.412	8.393	51.237%	4.667	7.211	43.649%
LSTM_BILSTM	4.351	6.763	44.938%	4.899	7.724	49.670%	6.142	10.046	58.980%	5.022	8.061	50.149%
DELA	4.101	6.253	41.926%	4.449	6.827	46.651%	5.229	8.184	59.904%	4.546	7.013	49.019%
T-GCN	4.208	6.461	43.834%	4.293	6.647	42.554%	4.675	7.404	45.111%	4.356	6.784	43.165%
STGNN	4.083	6.334	37.417%	4.435	6.937	40.515%	5.157	8.271	45.855%	4.494	7.094	40.821%
DCRNN	3.815	**5.867**	**33.469%**	3.914	6.074	**34.234%**	4.179	6.650	**35.667%**	3.938	6.137	**34.411%**
AGCRN	3.804	5.883	34.451%	3.910	6.084	35.492%	4.114	6.489	37.241%	3.920	6.112	36.113%
ASTGCN	4.010	6.180	35.861%	4.275	6.578	43.248%	4.939	7.770	49.410%	4.342	6.759	41.548%
MSTGCN	4.059	6.217	40.705%	4.415	6.799	44.060%	5.188	8.199	50.652%	4.485	6.982	44.111%
Graph-WaveNet	3.836	5.893	35.720%	4.009	6.221	37.979%	4.160	6.515	38.841%	3.979	6.175	37.521%
GMAN	3.883	5.914	41.163%	3.941	**6.014**	41.401%	**4.060**	**6.236**	42.677%	3.951	**6.036**	41.670%
ST-GRAT	3.980	6.141	33.529%	4.342	7.017	35.170%	5.478	9.390	47.838%	4.484	7.351	38.992%
MTGNN	3.827	5.924	34.029%	3.977	6.193	34.865%	4.321	6.832	38.513%	4.012	6.268	35.420%
RGSL	**3.740**	5.938	38.627%	**3.852**	6.129	39.687%	4.125	6.699	40.560%	**3.882**	6.214	39.412%
MT-STNet (ours)	**3.756**	**5.801**	**35.436%**	**3.807**	**5.885**	**35.520%**	**3.951**	**6.114**	**36.341%**	**3.827**	**5.918**	**35.691%**
Gains	-0.428%	1.125%	-5.877%	1.168%	2.145%	-3.756%	2.685%	1.956%	-1.890%	1.417%	1.955%	-3.720%

TABLE III
THE PREDICTION RESULTS OF THE MT-STNET AND BASELINES ON THE EXIT TOLL TASK

Model	Horizon 3			Horizon 6			Horizon 12			Average		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
SARIMA	4.891	7.375	50.088%	5.004	7.959	49.370%	5.218	8.561	52.795%	5.050	7.982	51.127%
SVR	4.118	6.496	36.771%	4.558	7.246	40.155%	5.473	8.803	48.131%	4.640	7.389	41.008%
LSTM_BILSTM	4.225	6.649	42.442%	4.838	7.783	47.105%	6.218	10.127	57.198%	4.974	8.108	47.912%
DELA	4.007	6.160	40.304%	4.395	6.841	45.175%	5.335	8.287	59.809%	4.509	7.018	47.853%
T-GCN	3.948	6.196	38.519%	4.076	6.442	37.411%	4.490	7.178	39.779%	4.126	6.539	37.928%
STGNN	3.911	6.159	35.291%	4.293	6.912	38.470%	5.138	8.527	44.911%	4.374	7.122	39.060%
DCRNN	**3.548**	**5.493**	**30.741%**	3.728	**5.852**	**31.172%**	3.996	6.387	**32.157%**	3.729	**5.862**	**31.297%**
AGCRN	3.599	5.611	32.873%	3.755	5.963	33.525%	3.945	6.305	34.825%	3.745	5.924	34.289%
ASTGCN	3.790	5.936	33.040%	4.049	6.367	38.639%	4.696	7.528	42.901%	4.108	6.529	37.201%
MSTGCN	3.854	5.945	38.565%	4.239	6.622	41.942%	5.091	8.181	49.853%	4.304	6.817	42.249%
Graph-WaveNet	3.588	5.586	32.219%	**3.714**	5.861	33.498%	**3.920**	6.303	35.042%	**3.714**	5.867	33.568%
GMAN	3.730	5.768	37.797%	3.812	5.949	38.262%	3.974	**6.259**	39.942%	3.824	5.968	38.555%
ST-GRAT	3.709	5.686	31.505%	4.104	6.583	32.018%	5.092	8.325	44.342%	4.181	6.703	35.887%
MTGNN	3.597	5.641	32.244%	3.799	6.043	32.822%	4.130	6.670	35.738%	3.810	6.069	33.256%
RGSL	3.631	5.856	34.811%	3.739	6.087	35.504%	4.070	6.852	36.477%	3.795	6.223	35.612%
MT-STNet (ours)	**3.685**	**5.853**	**32.075%**	**3.729**	**5.948**	**32.249%**	**3.859**	**6.155**	**33.340%**	**3.747**	**5.970**	**32.436%**
Gains	-3.861%	-6.554%	-4.339%	-0.404%	-1.640%	-3.455%	1.556%	1.662%	-3.679%	-0.889%	-1.842%	-3.639%

TABLE IV
THE PREDICTION RESULTS OF THE MT-STNET AND BASELINES ON THE GANTRY TASK

Model	Horizon 3			Horizon 6			Horizon 12			Average		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
SARIMA	7.143	14.607	43.648%	7.343	16.850	44.000%	7.398	17.047	45.397%	7.243	15.835	44.203%
SVR	5.774	8.480	30.719%	6.445	9.602	34.574%	8.008	12.153	44.858%	6.615	9.861	35.890%
LSTM_BILSTM	5.918	8.837	32.796%	6.728	10.297	36.322%	8.822	14.017	44.705%	6.988	10.932	37.180%
DELA	5.460	8.107	29.135%	5.853	8.903	30.878%	6.806	10.861	36.005%	5.959	9.159	31.549%
T-GCN	5.651	8.341	30.439%	5.870	8.782	30.214%	6.571	10.155	32.445%	5.963	8.980	30.663%
STGNN	5.600	8.471	28.324%	6.175	9.589	30.462%	7.486	12.026	34.977%	6.317	9.918	30.874%
DCRNN	**5.050**	**7.423**	27.087%	5.372	7.997	28.037%	5.817	8.933	**28.117%**	5.334	8.004	**27.551%**
AGCRN	5.218	7.712	27.075%	5.397	8.049	28.194%	5.730	8.730	29.373%	5.414	8.097	28.529%
ASTGCN	5.386	8.017	27.724%	5.779	8.733	31.181%	6.775	10.587	35.537%	5.881	8.990	30.620%
MSTGCN	5.419	8.007	29.794%	5.969	8.969	32.631%	7.257	11.199	40.257%	6.086	9.240	33.511%
Graph-WaveNet	5.161	7.655	26.544%	5.367	8.071	**27.731%**	5.733	8.813	28.859%	5.369	8.101	27.585%
GMAN	5.178	7.554	29.078%	**5.281**	**7.746**	29.511%	**5.512**	**8.175**	30.763%	**5.305**	**7.794**	29.684%
ST-GRAT	5.378	7.931	**25.471%**	6.111	9.457	28.087%	7.843	13.029	38.446%	6.213	9.872	30.450%
MTGNN	5.206	7.698	26.929%	5.502	8.226	27.810%	6.063	9.370	30.527%	5.519	8.322	28.096%
RGSL	5.329	8.007	28.138%	5.536	8.490	28.805%	6.081	9.764	29.911%	5.604	8.683	28.700%
MT-STNet (ours)	**5.128**	**7.531**	**26.698%**	**5.206**	**7.651**	**26.941%**	**5.406**	**7.962**	**27.701%**	**5.230**	**7.688**	**27.056%**
Gains	-1.545%	-1.455%	-4.817%	1.420%	1.226%	2.849%	1.924%	2.606%	1.480%	1.414%	1.360%	1.797%

dataset, because this model merely considers temporal correlations and ignores the impact of spatial correlation of traffic data. As shown in Table V, for the next twelve-time steps prediction in the whole dataset, DELA improved MAE, RMSE, and MAPE by 13.117%, 15.454%, and 8.947%, respectively, compared with LSTM_BILSTM. In addition, because highway traffic data is non-Euclidean in structure, DELA using CNN to model spatial correlations faces a fatal drawback. Compared with the classic graph neural network model T-GCN in Tables II, III, IV, and V, the lower prediction precision of DELA is evident. Moreover, an attractive finding can be noticed that the performance of DELA is lower than STGNN,

TABLE V
THE PREDICTION RESULTS OF THE MT-STNET AND BASELINES ON THE WHOLE DATASET

Model	Horizon 3			Horizon 6			Horizon 12			Average		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
SARIMA	6.195	12.262	47.116%	6.349	13.999	47.456%	6.440	14.237	49.150%	6.307	13.264	47.878%
SVR	5.166	7.794	33.463%	5.741	8.765	37.210%	7.039	10.947	46.680%	5.873	8.981	38.322%
LSTM_BILSTM	5.302	8.098	36.911%	6.025	9.417	40.880%	7.824	12.676	49.761%	6.236	9.952	41.657%
DELA	4.929	7.446	33.661%	5.312	8.183	36.559%	6.230	9.951	45.018%	5.418	8.414	37.930%
T-GCN	5.056	7.643	34.494%	5.233	8.010	33.902%	5.820	9.179	36.221%	5.312	8.182	34.395%
STGNN	4.994	7.706	31.358%	5.491	8.678	33.874%	6.602	10.800	38.909%	5.605	8.960	34.300%
DCRNN	**4.533**	**6.818**	28.982%	4.786	7.295	29.799%	5.163	8.106	**30.305%**	**4.767**	7.311	**29.554%**
AGCRN	4.644	7.034	29.564%	4.805	7.349	30.579%	5.087	7.929	31.889%	4.816	7.380	31.050%
ASTGCN	4.824	7.339	30.265%	5.168	7.955	34.868%	6.035	9.583	39.546%	5.255	8.183	33.926%
MSTGCN	4.866	7.339	33.512%	5.348	8.189	36.549%	6.457	10.167	44.034%	5.447	8.432	37.164%
Graph-WaveNet	4.613	6.994	29.349%	4.798	7.370	30.755%	5.093	7.990	31.911%	4.793	7.384	30.591%
GMAN	4.660	6.956	33.007%	**4.750**	**7.131**	33.409%	**4.947**	**7.505**	34.746%	4.769	**7.169**	33.623%
ST-GRAT	4.798	7.236	**28.133%**	5.398	8.551	30.166%	6.876	11.634	41.333%	5.502	8.906	33.090%
MTGNN	4.641	7.036	29.274%	4.892	7.499	**30.089%**	5.369	8.476	33.019%	4.911	7.559	30.454%
RGSL	4.681	7.250	31.379%	4.850	7.641	32.125%	5.299	8.710	33.161%	4.908	7.805	32.028%
MT-STNet (ours)	**4.596**	**6.936**	**29.363%**	**4.663**	**7.046**	**29.563%**	**4.839**	**7.325**	**30.397%**	**4.685**	**7.080**	**29.702%**
Gains	-1.390%	-1.731%	-4.372%	1.832%	1.192%	1.748%	2.183%	2.398%	-0.304%	1.720%	1.241%	-0.501%

MSTGCN, and ST-GRAT in tasks of entrance toll and exit toll flow prediction but outperforms in the job of gantry flow forecasting. For the next twelve-time steps forecasting, STGNN, MSTGCN, and ST-GRAT improved by 1.144%, 1.342%, and 1.364% in terms of MAE compared with DELA in Table II; 2.994%, 4.5465%, 7.274% in Table III; increased by 6.008%, 2.131%, and 4.262% in Table IV. The comparison results verify that connectivity between monitoring stations is significant for traffic flow prediction, and flow heterogeneity in the highway network causes the accuracy difference in main roads and ramps for STGNN, MSTGCN, and ST-GRAT.

Moreover, the traffic network is a dynamic graph, and the correlation between nodes changes over time. However, T-GCN uses the static connectivity matrix to describe the relationship between monitoring stations. Recently, graph attention networks (GATs) and adaptive dependency matrices (ADMs) have been widely utilized to extract the hidden spatial dependency adaptively in traffic prediction, successful cases such as AGCRN, Graph-WaveNet, GMAN, MTGNN, and RGSL. For instance, compared with T-GCN for the average scores of the next twelve horizons in the whole dataset, AGCRN, Graph-WaveNet, GMAN, MTGNN, and RGSL decreased MAE by 9.337%, 9.770%, 10.222%, 7.549%, and 7.605%, respectively; lowered RMSE by 9.802%, 9.753%, 12.381%, 7.614%, and 4.608%; improved MAPE by 9.725%, 11.060%, 2.245%, 11.458%, and 6.882%. The experimental results demonstrate that effective methods to extract hidden spatial correlation can enhance the performance of highway traffic flow prediction, and there is no doubt that GATs and ADMs are reasonable and inspire us. Furthermore, without GATs and ADMs, the performance of DCRNN is also superior to T-GCN because it considers the traffic diffusion and temporal dependency simultaneously in the directed network. For the average scores of the next twelve horizons in the whole dataset, DCRNN decreased MAE, RMSE, and MAPE by 10.260%, 10.645%, and 14.075%, respectively, compared with T-GCN.

In this paper, we absorb the advantages of graph attention networks in automatically modeling dynamic spatial dependency. Tables II, IV, and V show that the proposed

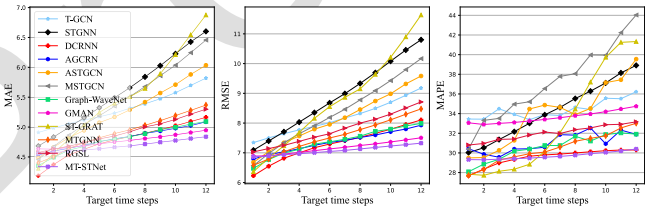


Fig. 8. Multi-step highway traffic flow forecasting ability of prediction models with GNN module. (a) MAE changes of different models in different target time steps; (b) RMSE changes; (c) MAPE changes.

MT-STNet outperforms all baselines for flow prediction regarding all metrics. For average scores of the next twelve horizons, compared with T-GCN, STGNN, DCRNN, AGCRN, ASTGCN, MSTGCN, Graph-WaveNet, GMAN, ST-GRAT, MTGNN, and RGSL, MT-STNet lowered MAE by 12.144%, 14.842%, 2.819%, 2.372%, 11.861%, 14.671%, 3.820%, 3.138%, 14.652%, 4.611%, and 1.417% in entrance toll task (as shown in Table II); 12.292%, 17.208%, 1.950%, 3.399%, 11.070%, 14.065%, 2.589%, 1.414%, 15.822%, 5.236%, and 6.674% in gantry task (as shown in Table IV); 11.803%, 16.414%, 1.720%, 2.720%, 10.847%, 13.989%, 2.253%, 1.761%, 14.849%, 4.602%, and 4.544% in the total dataset (as shown in Table V). In addition, Fig. 8 shows the traffic flow forecasting ability of prediction models with the GNN module in each time step, and the proposed model presents a respectable superiority. The comparison results demonstrate that MT-STNet is more evident in the multi-step traffic flow prediction (e.g., 60 minutes ahead). We argue that multi-step highway traffic flow forecasting is more valuable to deploy in the real world, e.g., it allows traffic managers to have more future traffic information and time to product traffic control strategies in highway networks.

Furthermore, while it may appear that the performance of MTSTNet is inferior to that of DCRNN, Graph-WaveNet, and GMAN in Table III, this is not the case. Firstly, the advantage of MTSTNet lies in its multi-step flow prediction capability, maintaining stable MAE, RMSE, and MAPE values and gradually outperforming all baselines from horizon 3 to horizon 12, as shown in Fig. 8. Secondly, MTSTNet accurately predicts traffic flow in the gantry task, especially with high

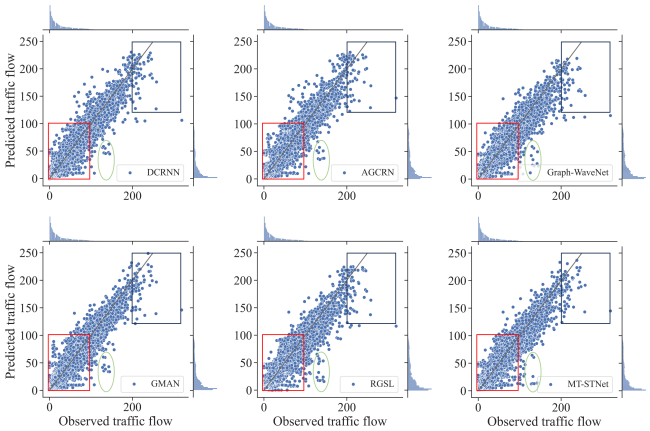


Fig. 9. Degree of fit between the observed and predicted traffic flow values on the whole dataset. The gray line indicates $y = x$, the blue dots indicate the degree of deviation between the observed and predicted values, and the blue histograms represent the distribution of observed and predicted.

traffic flow, which may cause a prediction shift in the exit toll task to ensure the total performance in the whole dataset, as depicted in the following Fig. 11. Thirdly, MTSTNet may exhibit high predicted values in a few steps with low ground truth values, resulting in a large MAPE. Most importantly, forecasting multi-step traffic flow in highway networks, especially high flow, is advantageous for the traffic management department to develop traffic control strategies, thereby preventing traffic networks from experiencing congestion. Note that the denominator is minor when the ground truth values are close to zero, resulting in a large MAPE if the predicted values differ from the ground truths.

2) Fitting Performance: To better demonstrate the performance of MT-STNet, we compare the fitting performance on the whole dataset and visualize the fitting results. Fig. 9 shows the visualization results of the predictive fit ability over the 60-min time span. Compared with the five optimal baseline models, we note the following four findings:

1) Actually, the proposed method presents satisfied performance below traffic flow in 50, and the blue dots in the red box are closer to the diagonal than the other five models. These experiments demonstrate that MT-STNet also maintains accurate forecasting performance for low traffic flow, and its overall performance is more prominent than others across the whole dataset, unlimited by the exit toll task.
2) In addition, compared with DCRNN and GMAN, a few dots in the green oval box are found that the discrete degree is higher for MT-STNet when traffic flows between 100 and 150. The multi-task learning balances the accuracy of each subtask, which may lead to a few prediction shifts appearing.
3) Moreover, when the traffic flow exceeds 150, for baselines, blue dots contained in the black box are gradually far away from the diagonal, but the proposed method is not. In particular, MT-STNet keeps high prediction accuracy without any blue dots out of the black box compared with baselines. The comparison result validates the application value of the proposed method in the heavy traffic flow environment.
4) Furthermore, more critical information deserves attention that both blue histogram arrays depict the observed and predicted distributions, and both arrays in the MT-STNet model are more consistent than others. For the multi-step traffic flow prediction, the practical applicability value of MT-STNet in the highway network is confirmed once more.

3) Influence of Each Component: Seven variants are compared in the ablation experiments to verify the necessity of each component for MT-STNet, and the influence per module is then demonstrated in this part. W/O physical information, neglecting the auxiliary road structure, such as station degree, shortest path, and shortest path distance, in spatiotemporal dependencies modeling. W/O multi-head GCN, ignoring the connectivity between stations, including upstream and downstream. W/O spatial attention, assuming the spatial correlation between stations is static over time. W/O temporal attention, do not consider the dynamic temporal correlation between time steps. W/O fusion gate mechanism, employing the addition operation instead of the automotive fusion method. W/O generative inference, MT-STNet uses dynamic decoding to replace. W/O multi-task learning module, disregarding the traffic flow heterogeneity in the prediction process. Table VI and Fig. 10 show the performance of each variant on the whole dataset, and the contribution is heightened in the following comparisons,

a) W/O physical information: As observed in Table VI and Fig. 10, the variant without physical information performs worse than MTSTNet and the variant without a gate mechanism. Compared with MT-STNet, *w/o physical information* causes the accuracy to decrease. For horizon twelve, the MAE, RMSE, and MAPE increased by 0.434%, 0.396%, and 6.718%, respectively; by 0.192%, -0.254%, and 7.744% for the next twelve horizons forecasting. In addition, there is a significant gap in MAPE between MTSTNet and the variant without physical information. Removing physical information leads to substantial errors in MAPE compared to the baselines. The experimental results reflect the effectiveness of physical information is visible, which indicates the value in traffic prediction, especially highway networks.

b) W/O multi-head GCN: Compared with MT-STNet, the variant without multi-head GCN presents insufficient predictive ability. For horizon six, the MAE, RMSE, and MAPE increased by 1.029%, 1.008%, and 9.045% respectively; by 0.854%, 0.819%, and 8.834% for the next twelve horizons forecasting. In addition, the performance of *w/o multi-head GCN* is also worse than that of variants without physical information, multi-task learning, and fusion gate mechanism, and the inferiority is evident. These comparison results validate the indispensability of static spatial dependency for spatial modeling, highlighting the crucial role of directed connectivity between stations.

c) W/O spatial attention: The performance without the spatial attention module is worse than MT-STNet and the variant without the multi-head GCN. For example, *w/o spatial attention module* increases upon MT-STNet by 2.625%, 3.044%, and 11.791% in terms of MAE, RMSE, and MAPE, respectively, for the horizon twelve; 1.580%, 1.412%, and

TABLE VI
PERFORMANCE OF THE DIFFERENT TIME STEPS PREDICTION FOR DISTINGUISHED VARIANTS

Model	Horizon 3			Horizon 6			Horizon 12			Average		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
w/o Physical Information	**4.592**	**6.881**	31.722%	4.676	7.031	31.998%	4.860	7.354	32.439%	4.694	7.062	32.002%
w/o Multi-head GCN	4.633	6.973	31.933%	4.711	7.117	32.237%	4.873	7.391	33.017%	4.725	7.138	32.326%
w/o Spatial Attention	4.653	6.976	32.715%	4.728	7.132	32.944%	4.966	7.548	33.981%	4.759	7.180	33.110%
w/o Temporal Attention	4.676	7.100	30.487%	4.717	7.154	30.735%	4.915	7.464	31.949%	4.753	7.214	30.939%
w/o Fusion Gate Mechanism	4.600	6.888	32.319%	**4.673**	**7.030**	32.652%	**4.851**	7.334	33.142%	**4.692**	**7.056**	32.672%
w/o Generative Inference	4.650	6.994	31.713%	4.735	7.124	32.512%	4.983	7.538	33.650%	4.766	7.180	32.561%
w/o Multi-task Learning	4.609	6.947	**29.456%**	4.685	7.076	**29.741%**	4.855	**7.332**	**30.715%**	4.700	7.092	**29.863%**
MT-STNet	**4.596**	**6.936**	**29.363%**	**4.663**	**7.046**	**29.563%**	**4.839**	**7.325**	**30.397%**	**4.685**	**7.080**	**29.702%**
Gains	-0.087%	-0.799%	0.316%	0.214%	-0.228%	0.599%	0.247%	0.095%	1.035%	0.149%	-0.340%	0.539%

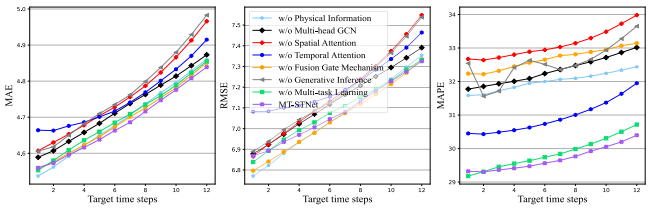


Fig. 10. Multi-step highway traffic flow forecasting ability of all variants. (a) MAE changes of different variants in different target time steps; (b) RMSE changes; (c) MAPE changes.

11.474% for the next twelve horizons. In addition, for the horizon twelve, compared with the variant without the multi-head GCN, the MAE, RMSE, and MAPE increased by 1.908%, 2.124%, and 2.920%; 0.720%, 0.588%, and 2.425% for the next twelve horizons. The experimental results demonstrate that the spatial attention component is critical for dynamic spatial correlation modeling and more significant than the multi-head GCN module in traffic flow prediction.

d) W/O temporal attention: Temporal correlation is another character that directly affects the time series prediction, such as *w/o temporal attention* leads to a lower accuracy, as shown in Table VI. For horizon twelve, *w/o temporal attention component* results in 1.571%, 1.898%, and 5.106% increase in terms of MAE, RMSE, and MAPE compared with MT-STNet; 1.451%, 1.893%, and 4.165% for the next twelve horizons forecasting. Additionally, its prediction performance lags behind most variants in terms of both MAE and MAPE. The experiments demonstrate that the dynamic temporal correlation is an essential element for traffic flow prediction, and the temporal attention method is verified as a practical way to echo.

e) W/O fusion gate mechanism: Compared with MT-STNet, the variant without an adaptive fusion mechanism slightly decreases prediction performance regarding MAE and RMSE. However, removing the fusion gate mechanism causes the MAPE to decrease dramatically, as shown in Fig. 10. For horizon twelve, MAE, RMSE, and MAPE of *w/o fusion gate module* increased by 0.248%, 0.123%, and 9.030%, respectively; 0.149%, -0.339%, and 9.999% for the next twelve horizons. These results demonstrate the positive effect of fusing distinct aspect features in an adaptive manner rather than via simple addition, providing new insights for feature fusion in our future work.

f) W/O generative inference: For the variant without generative inference, time costs on training and inference phases are 0.823 min and 3.114 min, respectively, and the parameters and GPU memory usage remain consistent with MT-STNet. In addition, compared with MT-STNet, MAE, RMSE, and MAPE of *w/o generative inference module* increased by 2.976%, 2.908%, and 10.702% for horizon twelve; 1.729%, 1.412%, and 9.626% for the next twelve time steps. Moreover, the time costs of training and inference stages increased by 82.889% and 29.480%, respectively. Furthermore, its performance is inferior to all of the variants regarding all metrics. The experiments verify the weakness of dynamic decoding, and the proposed generative system is an effective technique to generate multi-step traffic flow without prediction error propagation.

g) W/O multi-task learning: The accuracy is weakened by discarding traffic heterogeneity in the highway system. As shown in Table VI and Fig. 10, the variant without multi-task learning performs worse than MTSTNet and variants lacking physical information and fusion gate mechanism. For instance, compared with MT-STNet, the variant without multi-task learning increased MAE, RMSE, and MAPE by 0.472%, 0.096%, and 1.046%, respectively, for horizon six; 0.320%, 0.169%, and 5.421% for the next twelve time steps. Moreover, the gap in MAPE between MTSTNet and the variant without multi-task learning widens gradually with an increase in time steps. The experiments verify that multi-task learning is an appropriate technique for handling flow heterogeneity in traffic networks, which divides traffic prediction into several relative subtasks, sharing underlying knowledge.

4) Computation Cost: Table V presents the prediction performance comparisons in the whole dataset, and the computation costs of the baselines and the proposed model for the next twelve horizons are shown in Table VII, including total parameters, time cost, and GPU memory usage.

Table VII shows that the SARIMA and SVR cost more time in both training and inference phases but have weak prediction results, as shown in Table V. For the graph neural networks, T-GCN, ASTGCN, and MSTGCN use fewer parameters and occupy less GPU memory, but present satisfactory performance compared with DELA based on CNNs. In addition, six optimal baselines, DCRNN, AGCRN, Graph-WaveNet, GMAN, MTGNN, and RGSL, achieve high prediction performance, but computation costs are various. These six baselines have identical properties that demand high GPU memory usage in training and inference phases, except GMAN, as shown in Table VII. To get a competitive accuracy, AGCRN, GMAN, and RGSL apply more parameters, leading to high time costs in the inference phase. Note that, like ST-GRAT, DCRNN

TABLE VII
COMPUTATION COST DURING THE TRAINING AND INFERENCE PHASES (* MEANS THE MODEL TRAIN ONE TIME ON THE WHOLE TRAINING SET)

Model	Parameters	Training / (100 iterations) (batch size =128)		Inference (batch size =1)	
		Time Cost	GPU Memory Usage	Time Cost	GPU Memory Usage
SARIMA*	-	728.092 (min)	-	226.710 (min)	-
SVR*	-	571.626 (min)	-	108.240 (min)	-
LSTM_BILSTM	1,121,089	0.810 (min)	8691MiB	3.022 (min)	531MiB
DELA	120,423	0.180 (min)	7277MiB	0.601 (min)	1965MiB
T-GCN	37,844	0.064 (min)	1523MiB	0.305 (min)	501MiB
STGNN	617,985	0.165 (min)	2827MiB	1.179 (min)	1619MiB
DCRNN	372,353	2.144 (min)	4153MiB	13.299 (min)	1615MiB
AGCRN	750,240	0.152 (min)	3105MiB	2.108 (min)	2035MiB
ASTGCN	74,312	0.348 (min)	2479MiB	0.701 (min)	1789MiB
MSTGCN	50,956	0.327 (min)	2241MiB	0.596 (min)	1787MiB
Graph-WaveNet	306,580	0.124 (min)	2869MiB	0.647 (min)	1759MiB
GMAN	916,801	1.359 (min)	16899MiB	2.205 (min)	531MiB
ST-GRAT	2,238,849	0.900 (min)	17789MiB	15.033 (min)	1639MiB
MTGNN	204,668	0.121 (min)	2801MiB	0.600 (min)	1727MiB
RGSL	871,312	0.347 (min)	3945MiB	4.407 (min)	1681MiB
MT-STNet (ours)	192,771	0.450 (min)	8707MiB	2.405 (min)	523MiB

employs dynamic decoding to generate targets, which causes high time costs in both stages.

Moreover, we prefer a faster, more efficient, low-complexity model that uses less GPU memory while maintaining accurate prediction. Therefore, MT-STNet is proposed as having superior performance, and its model complexity is less than that of DCRNN, AGCRN, Graph-WaveNet, GMAN, MTGNN, and RGSL. In the training phase, because of the difference in data loading, the GPU memory usage of MT-STNet is higher than that of the optimal baseline, DCRNN. In contrast, in the inference phase, the GPU memory usage is minimal, and the time cost outperforms the top two optimal baselines, DCRNN and GMAN, as shown in Table VII. MT-STNet provides multi-step forecasts in a single pass, reducing the time required for inference compared with DCRNN and GMAN. The computation cost further validates the superiority of MT-STNet in multi-step highway traffic flow prediction.

5) Case Study: We selected three monitoring stations on each task and visualized the prediction results for the twelve horizons on the test set, as shown in Fig. 11 left. In the visualization stage, one hundred samples are randomly sliced from the test dataset, the sliding window is twelve, and the time interval is 2021.8.13 15: 25 - 2021.8.17 19:25. Due to limited space, seven optimal baselines are chosen, including DCRNN, AGCRN, ASTGCN, Graph-WaveNet, GMAN, MTGNN, and RGSL. Fig. 11 left shows that MT-STNet accurately fits the traffic flow compared with baselines, even though the flow meets the inflection point, i.e., zero or maximum value. For example, samples in the red dashed box are selected, and peak values are contained in this short period with tremendous traffic fluctuation, which reflects the difficulty of flow prediction. The MT-STNet is more sensitive to traffic fluctuation and presents an exciting performance in high-traffic flow points, which is closely related to the structure of the model itself.

To further illustrate the forecasting performance of the MT-STNet model and baselines, we visualize the historical, observed, and predicted values (the samples correspond to the red dash box in Fig. 11 left) on these three types of tasks, as shown in Fig. 11 right. The prediction period is 2021.8.16 7: 10 - 2021.8.16 8: 10, in the morning peak.

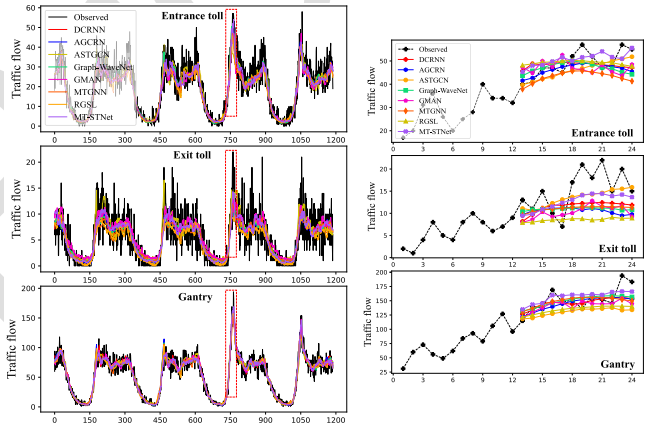


Fig. 11. **Left**: The visualization of fitting results in the gantry, entrance toll, and exit toll. **Right**: example of predicting the next 12 time steps (1-12 is historical time steps, and 13-24 is predicted horizons).

Compared with the optimal seven baselines, MT-STNet adapts to traffic flow situations during peak periods and makes predictions similar to actual observed values. These properties play a vital role in future travel services and traffic control.

Furthermore, based on Fig. 11, capturing the time-varying patterns of ground truths is particularly challenging at a few points, notably in the exit toll task. This challenge arises from two main reasons: (1) the limitation of dataset size (i.e., only three months), preventing models from thoroughly learning traffic patterns and regularities from historical data; (2) the occurrence of traffic incidents, such as adverse weather, accidents, and road maintenance, leading to abnormal fluctuations in traffic flow and making it difficult for models to provide accurate prediction results. These two reasons are common challenges faced by all methods. To address these challenges, two potential directions are envisioned: increasing the number of samples and collecting more relevant source data (e.g., weather and accident data).

VI. CONCLUSION

In this paper, we introduce a novel traffic flow prediction model for highway networks, entitled multi-task

spatiotemporal network (MT-STNet), aimed at alleviating the prediction accuracy pressure in multi-step and high-traffic flow scenarios. To address the problem that existing methods barely consider the insufficient spatiotemporal correlation modeling, prediction error accumulation, and heterogeneous flow distributions, we correspondingly design ST-Physical Block, generative inference system, and multi-task learning modules to handle these perspectives. Specifically, the ST-Physical Block is initially employed to automatically extract dynamic temporal, dynamic spatial, and static spatial dependencies, integrating physical structure information into modeling spatiotemporal dependencies. The issue of error accumulation in multi-step prediction is subsequently addressed by a dedicated generative inference system. Finally, multi-task learning is employed to convert the problem of heterogeneous flow distributions in the highway network into three subtasks.

Experiments on the real-world dataset demonstrate that MT-STNet achieves state-of-the-art results compared to the baselines. Particularly noteworthy is MT-STNet's pronounced advantage in multi-step forecasting, especially when the prediction step exceeds two. Moreover, MT-STNet surpasses all baselines in the high-traffic flow prediction task. Furthermore, through comparative analysis of the prediction performance of all variants with MT-STNet, the function of each component is assessed, and their contributions to traffic flow prediction are determined. In future work, we will explore the influence of external factors, such as traffic accidents, on traffic flow prediction. Additionally, we plan to utilize traffic flow on road segments as a dynamic relationship between monitoring stations, replacing attention and adjacent matrices to simulate traffic diffusion.

ACKNOWLEDGMENT

The authors would like to express their gratitude to EditSprings (https://www.editsprings.cn) for the expert linguistic services provided.

REFERENCES

[1] S. Guo, Y. Lin, H. Wan, X. Li, and G. Cong, "Learning dynamics and heterogeneity of spatial–temporal graph data for traffic forecasting," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 11, pp. 5415–5428, Nov. 2022.

[2] B. M. Williams and L. A. Hoel, "Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: Theoretical basis and empirical results," *J. Transp. Eng.*, vol. 129, no. 6, pp. 664–672, 2003.

[3] P. Cai, Y. Wang, G. Lu, P. Chen, C. Ding, and J. Sun, "A spatiotemporal correlative k-nearest neighbor model for short-term traffic multistep forecasting," *Transp. Res. C, Emerg. Technol.*, vol. 62, pp. 21–34, Jan. 2016.

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

[5] Y. Qi and S. Ishak, "A hidden Markov model for short term prediction of traffic conditions on freeways," *Transp. Res. C, Emerg. Technol.*, vol. 43, pp. 95–111, Jun. 2014.

[6] X. Yin, G. Wu, J. Wei, Y. Shen, H. Qi, and B. Yin, "Deep learning on traffic prediction: Methods, analysis, and future directions," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 6, pp. 4927–4943, Jun. 2022.

[7] C. Ma, G. Dai, and J. Zhou, "Short-term traffic flow prediction for urban road sections based on time series analysis and LSTM_BILSTM method," *IEEE Trans. Intell. Transp. Syst.*, 2021. AQ:2

[8] Z. Zheng, Y. Yang, J. Liu, H.-N. Dai, and Y. Zhang, "Deep and embedded learning approach for traffic flow prediction in urban informatics," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 10, pp. 3927–3939, Oct. 2019.

[9] H. Zheng, F. Lin, X. Feng, and Y. Chen, "A hybrid deep learning model with attention-based conv-LSTM networks for short-term traffic flow prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 11, pp. 6910–6920, Nov. 2021.

[10] L. Zhao et al., "T-GCN: A temporal graph convolutional network for traffic prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 9, pp. 3848–3858, Aug. 2019.

[11] X. Zhang, "Traffic flow forecasting with spatial–temporal graph diffusion network," in *Proc. AAAI Conf. Artif. Intell.*, 2021, vol. 35, no. 17, pp. 15008–15015.

[12] C. Zheng, X. Fan, C. Wang, and J. Qi, "GMAN: A graph multi-attention network for traffic prediction," in *Proc. AAAI Conf. Artif. Intell.*, Apr. 2020, vol. 34, no. 1, pp. 1234–1241.

[13] G. Zou, Z. Lai, C. Ma, Y. Li, and T. Wang, "A novel spatio-temporal generative inference network for predicting the long-term highway traffic speed," *Transp. Res. C, Emerg. Technol.*, vol. 154, Sep. 2023, Art. no. 104263.

[14] B. Lu, X. Gan, H. Jin, L. Fu, X. Wang, and H. Zhang, "Make more connections: Urban traffic flow forecasting with spatiotemporal adaptive gated graph convolution network," *ACM Trans. Intell. Syst. Technol.*, vol. 13, no. 2, pp. 1–25, Apr. 2022.

[15] A. Vaswani, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017. AQ:3

[16] H. Zhou, "Informer: Beyond efficient transformer for long sequence time-series forecasting," in *Proc. AAAI*, 2021.

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

[18] Y. Zhang and Q. Yang, "A survey on multi-task learning," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 12, pp. 5586–5609, Mar. 2021.

[19] S. R. Chandra and H. Al-Deek, "Predictions of freeway traffic speeds and volumes using vector autoregressive models," *J. Intell. Transp. Syst.*, vol. 13, no. 2, pp. 53–72, 2009.

[20] I. Okutani and Y. J. Stephanedes, "Dynamic prediction of traffic volume through Kalman filtering theory," *Transp. Res. B, Methodol.*, vol. 18, no. 1, pp. 1–11, 1984.

[21] M. Van Der Voort, M. Dougherty, and S. Watson, "Combining Kohonen maps with ARIMA time series models to forecast traffic flow," *Transp. Res. C, Emerg. Technol.*, vol. 4, no. 5, pp. 307–318, 1996.

[22] G. Comert and A. Bezuglov, "An online change-point-based model for traffic parameter prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 3, pp. 1360–1369, Sep. 2013.

[23] P. Sun, N. Aljeri, and A. Boukerche, "Machine learning-based models for real-time traffic flow prediction in vehicular networks," *IEEE Netw.*, vol. 34, no. 3, pp. 178–185, May 2020.

[24] S. Yang, J. Wu, Y. Du, Y. He, and X. Chen, "Ensemble learning for short-term traffic prediction based on gradient boosting machine," *J. Sensors*, vol. 2017, pp. 1–15, Sep. 2017.

[25] S. Sun, C. Zhang, and G. Yu, "A Bayesian network approach to traffic flow forecasting," *IEEE Trans. Intell. Transp. Syst.*, vol. 7, no. 1, pp. 124–132, Mar. 2006.

[26] P. Xie, T. Li, J. Liu, S. Du, X. Yang, and J. Zhang, "Urban flow prediction from spatiotemporal data using machine learning: A survey," *Inf. Fusion*, vol. 59, pp. 1–12, Jul. 2020.

[27] L. Qu, W. Li, W. Li, D. Ma, and Y. Wang, "Daily long-term traffic flow forecasting based on a deep neural network," *Expert Syst. Appl.*, vol. 121, pp. 304–312, May 2019.

[28] P. Wang, W. Hao, and Y. Jin, "Fine-grained traffic flow prediction of various vehicle types via fusion of multisource data and deep learning approaches," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 11, pp. 6921–6930, Nov. 2021.

[29] Q. Tao, Z. Li, J. Xu, S. Lin, B. De Schutter, and J. A. K. Suykens, "Short-term traffic flow prediction based on the efficient hinging hyperplanes neural network," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 9, pp. 15616–15628, Sep. 2022.

[30] S. Hao, D.-H. Lee, and D. Zhao, "Sequence to sequence learning with attention mechanism for short-term passenger flow prediction in large-scale metro system," *Transp. Res. C, Emerg. Technol.*, vol. 107, pp. 287–300, Oct. 2019.

[31] S. Reza, M. C. Ferreira, J. J. M. Machado, and J. M. R. S. Tavares, "A multi-head attention-based transformer model for traffic flow forecasting with a comparative analysis to recurrent neural networks," *Expert Syst. Appl.*, vol. 202, Sep. 2022, Art. no. 117275.

[32] Y. Wen, P. Xu, Z. Li, W. Xu, and X. Wang, "RPConvformer: A novel transformer-based deep neural networks for traffic flow prediction," *Expert Syst. Appl.*, vol. 218, May 2023, Art. no. 119587.

[33] L. Liu et al., "Dynamic spatial–temporal representation learning for traffic flow prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 11, pp. 7169–7183, Nov. 2021.

[34] T. Jia and P. Yan, "Predicting citywide road traffic flow using deep spatiotemporal neural networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 5, pp. 3101–3111, May 2021.

[35] Z. Cheng, J. Lu, H. Zhou, Y. Zhang, and L. Zhang, "Short-term traffic flow prediction: An integrated method of econometrics and hybrid deep learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 6, pp. 5231–5244, Jun. 2022.

[36] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, *arXiv:1609.02907*.

[37] Z. Li et al., "A hybrid deep learning approach with GCN and LSTM for traffic flow prediction," in *Proc. IEEE Intell. Transp. Syst. Conf. (ITSC)*, Oct. 2019, pp. 1929–1933.

[38] S. Wang, M. Zhang, H. Miao, Z. Peng, and P. S. Yu, "Multivariate correlation-aware spatio-temporal graph convolutional networks for multi-scale traffic prediction," *ACM Trans. Intell. Syst. Technol.*, vol. 13, no. 3, pp. 1–22, Jun. 2022.

[39] S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan, "Attention based spatial–temporal graph convolutional networks for traffic flow forecasting," in *Proc. AAAI Conf. Artif. Intell.*, 2019, vol. 33, no. 1, pp. 922–929.

[40] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," in *Proc. Int. Conf. Learn. Represent.*, 2018.

[41] M. Lv, Z. Hong, L. Chen, T. Chen, and S. Ji, "Temporal multi-graph convolutional network for traffic flow prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 6, pp. 3337–3348, May 2020.

[42] L. Bai, L. Yao, C. Li, X. Wang, and C. Wang, "Adaptive graph convolutional recurrent network for traffic forecasting," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 17804–17815.

[43] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, "Graph WaveNet for deep spatial–temporal graph modeling," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, 2019, pp. 1907–1913.

[44] Z. Wu, S. Pan, G. Long, J. Jiang, X. Chang, and C. Zhang, "Connecting the dots: Multivariate time series forecasting with graph neural networks," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2020, pp. 753–763.

[45] H. Yu et al., "Regularized graph structure learning with semantic knowledge for multi-variates time-series forecasting," 2022, *arXiv:2210.06126*.

[46] J. Huang, K. Luo, L. Cao, Y. Wen, and S. Zhong, "Learning multi-aspect traffic couplings by multirelational graph attention networks for traffic prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 11, pp. 20681–20695, Nov. 2022.

[47] Y. Duan, N. Chen, S. Shen, P. Zhang, Y. Qu, and S. Yu, "FDSA-STG: Fully dynamic self-attention spatio-temporal graph networks for intelligent traffic flow prediction," *IEEE Trans. Veh. Technol.*, vol. 71, no. 9, pp. 9250–9260, Sep. 2022.

[48] X. Wang et al., "Traffic flow prediction via spatial temporal graph neural network," in *Proc. Web Conf.*, Apr. 2020, pp. 1082–1092.

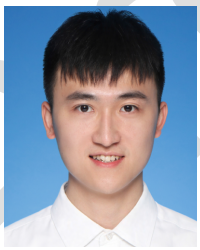
[49] C. Park et al., "ST-GRAT: A novel spatio-temporal graph attention networks for accurately forecasting dynamically changing road speed," in *Proc. 29th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2020, pp. 1215–1224.

[50] G. Zou, Z. Lai, C. Ma, M. Tu, J. Fan, and Y. Li, "When will we arrive? A novel multi-task spatio-temporal attention network based on individual preference for estimating travel time," *IEEE Trans. Intell. Transp. Syst.*, 2023.

[51] C. Ying et al., "Do transformers really perform badly for graph representation?" in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 28877–28888.

[52] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, pp. 287–290, Dec. 2022.

Guojian Zou (Member, IEEE) received the M.S. degree from the College of Information, Mechanical and Electrical Engineering, Shanghai Normal University, Shanghai, China, in 2020. He is currently pursuing the Ph.D. degree with the College of Transportation Engineering, Tongji University, China. From 2023 to 2024, he was a Visiting Student with the Department of Geography, University of Zurich, Switzerland. He is the author of more than 25 articles, including IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, *Transportation Research—Part C: Emerging Technologies*, and *Expert Systems with Applications*. His research interests include intelligent transportation systems, urban computing, deep learning, natural language processing, and computer vision. His awards and honors include the Scholarship of China Scholarship Council and the National Scholarship for Doctoral Students. He serves as a reviewer for IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, IEEE TRANSACTIONS ON INTELLIGENT VEHICLES, IEEE TRANSACTIONS ON ARTIFICIAL INTELLIGENCE, *International Journal of Geographical Information Science*, IEEE INTERNET OF THINGS JOURNAL, and *Expert Systems with Applications*.



Ziliang Lai received the B.S. degree in transportation engineering from Beijing Jiaotong University, Beijing, China, in 2020. He is currently pursuing the M.S. degree with the College of Transportation Engineering, Tongji University, China. He is the author of more than ten articles, including IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS and *Transportation Research—Part C: Emerging Technologies*. His research interests include autonomous vehicle ridesharing, vehicle path optimization, and deep learning.

Ting Wang received the B.S. degree in traffic engineering from Jiangsu University, China, in 2018, and the M.S. degree from Ningbo University, China, in 2021. He is currently pursuing the Ph.D. degree with the College of Transportation Engineering, Tongji University, China. His research interests include the application of artificial intelligence technology in traffic flow modeling and traffic control.

Zongshi Liu received the M.Sc. degree from the Department of Civil, Environmental and Geomatic Engineering, University College London, U.K., in 2020. He is currently pursuing the Ph.D. degree with the College of Transportation Engineering, Tongji University, China. He has published more than five academic articles in international journals. His research interests include traffic safety, intelligent transportation systems, and connected and autonomous vehicles technology.

Ye Li received the B.S., M.S., and Ph.D. degrees in transportation engineering from Tongji University, Shanghai, China, in 1995, 2000, and 2003, respectively. In 2011, he was a Visiting Scholar with the University of California at Berkeley, Berkeley, CA, USA. He is currently the Vice President of Shanghai Normal University, Shanghai, China. He is a Professor with the College of Transportation Engineering, Tongji University. He is the author of one book and more than 100 articles. His research interests include public transportation planning, low-carbon transportation system planning, and transportation service pattern innovation with big data. His awards and honors include the State Science and Technology Prize, China Navigation Technology Award, and the New Century Talents Award.