

Traffic flow prediction based on frequency-domain dynamic graph and Mamba

Yifei Zhang(张逸飞) and Jialin He(何嘉林)

Citation: Chin. Phys. B, 2026, 35 (**6**): 060204. DOI: 10.1088/1674-1056/ae3069
http://cpb.iphy.ac.cn; http://iopscience.iop.org/cpb

What follows is a list of articles you may be interested in

Programming guide for solving constraint satisfaction problems with tensor networks

Xuanzhao Gao(高煊钊), Xiaofeng Li(李晓锋), and Jinguo Liu(刘金国)
Chin. Phys. B, 2025, 34 (**5**): 050201. DOI: 10.1088/1674-1056/adbee2

Graph distillation with network symmetry

Feng Lin(林峰) and Jia-Lin He(何嘉林)
Chin. Phys. B, 2025, 34 (**4**): 040204. DOI: 10.1088/1674-1056/adbeda

Self-similarity of multilayer networks

Bing Wang(王冰), Huizhi Yu(于蕙芷), and Daijun Wei(魏代俊)
Chin. Phys. B, 2025, 34 (**1**): 010202. DOI: 10.1088/1674-1056/ad8a47

Opinion consensus incorporating higher-order interactions in individual-collective networks

Shun Ye(叶顺), Li-Lan Tu(涂俐兰), Xian-Jia Wang(王先甲), Jia Hu(胡佳), and Yi-Chao Wang(王薏潮)
Chin. Phys. B, 2024, 33 (**7**): 070201. DOI: 10.1088/1674-1056/ad3b84

Influence of random phase modulation on the imaging quality of computational ghost imaging

Chao Gao(高超), Xiao-Qian Wang(王晓茜), Hong-Ji Cai(蔡宏吉), Jie Ren(任捷), Ji-Yuan Liu(刘籍元), Zhi-Hai Yao(姚治海)
Chin. Phys. B, 2019, 28 (**2**): 020201. DOI: 10.1088/1674-1056/28/2/020201

Traffic flow prediction based on frequency-domain dynamic graph and Mamba

Yifei Zhang(张逸飞)[1] and Jialin He(何嘉林)[1,2,3,†]

[1] *School of Computer Science, China West Normal University, Nanchong* 637009, *China*
[2] *Artificial Intelligence Key Laboratory of Nanchong, China West Normal University, Nanchong* 637009, *China*
[3] *The Internet of Things Perception and Big Data Analysis Key Laboratory of Nanchong City, Nanchong* 637009, *China*

(Received 8 October 2025; revised manuscript received 21 December 2025; accepted manuscript online 23 December 2025)

With the rapid advancement of intelligent transportation systems (ITS), urban traffic prediction faces significant challenges in effectively modeling complex spatio–temporal dynamics while maintaining computational efficiency. Existing approaches are often limited by static graph structures and the high computational cost of self-attention mechanisms when processing long sequences. To overcome these limitations, this paper proposes a novel framework, termed spatio–temporal frequency-domain mamba network (STFD-MambaNet). Specifically, the framework integrates a frequency-domain dynamic graph learner to capture evolving traffic topologies and employs the Mamba structured state space model to efficiently extract long-range temporal dependencies with linear complexity. Furthermore, a hierarchical spatial modeling module is developed to characterize multi-scale spatial correlations. Experiments conducted on four real-world datasets demonstrate that STFD-MambaNet consistently outperforms state-of-the-art methods in both accuracy and efficiency. The results further demonstrate the effective complementarity between frequency-domain dynamic graph learning and structured state space modeling, providing a robust solution for spatio–temporal traffic forecasting.

Keywords: traffic flow forecasting, frequency-domain graph learning, Mamba, graph convolutional network (GCN)

PACS: 02.10.Ox, 45.70.Vn, 06.30.Ft

DOI: 10.1088/1674-1056/ae3069 **CSTR:** 32038.14.CPB.ae3069

1. Introduction

With the acceleration of global urbanization and the increasing concentration of population, complex traffic environments have emerged as a major challenge for urban management, thereby underscoring the urgent need for intelligent transportation systems (ITS) powered by artificial intelligence to enhance urban capacity.[1,2] Traffic forecasting, as an indispensable component of ITS, aims to accurately predict future traffic states by learning from historical traffic data and patterns collected by sensors.[3] However, owing to the strong correlations among sensors, the future state of each sensor depends not only on its own historical information but also on the historical data of other sensors.

In recent years, deep learning methods have become a popular choice for traffic flow forecasting using high-dimensional spatio–temporal data. A common and classical approach is to cascade convolutional neural networks (CNNs) with recurrent neural networks (RNNs) to jointly model spatial and temporal dependencies in traffic prediction tasks.[4,5] Within this framework, CNNs are typically employed to extract spatial features, whereas RNNs are leveraged to capture the dynamic evolution patterns of temporal sequences. Although CNNs are effective in capturing local spatial correlations in regular grid structures, their performance is limited in non-Euclidean domains, making it difficult to adequately represent the complex and irregular spatial topologies of real-world road networks — particularly those involving long-range spatial dependencies. Consequently, directly applying CNNs to traffic forecasting suffers from inherent limitations, restricting the ability of such models to capture non-local spatial information in urban traffic systems.

Another common approach is to employ graph convolutional networks (GCNs) to model the spatio–temporal correlations in time-series data together with the non-Euclidean spatial structures of road networks. GCNs are well-suited for processing graph-structured data and are naturally applicable to representing traffic networks with complex connectivity patterns.[6–8] However, most existing GCN-based methods rely on static adjacency matrices to define the connection weights between nodes. Such fixed structures lack the flexibility required to model dynamic traffic systems, making it difficult to accurately capture the temporal evolution of spatial dependencies across different time periods. To address this issue, recent studies have proposed dynamic graph construction methods. For example, SFTGNN[9] incorporates dynamic time warping (DTW)[10] to match the shapes of time series between nodes and constructs dynamic graph structures to reflect their similarity.[11] Although this approach alleviates the limitations of static graphs to some extent, characterizing node

†Corresponding author. E-mail: hejialin32@126.com

© 2026 Chinese Physical Society and IOP Publishing Ltd. All rights, including for text and data mining, AI training, and similar technologies, are reserved.

dependencies solely based on sequence-shape similarity remains insufficient. In real traffic systems, spatial dependencies among nodes are not only reflected in the similarity of numerical variation trends but are also embedded in deeper semantic correlations, analogous to natural language processing tasks where two sentences with similar semantics but distinct syntactic structures are still considered related. Therefore, traffic modeling should also account for the latent semantic information among nodes[12] to comprehensively capture complex spatio–temporal dependencies. Figure 1 illustrates the dynamic spatial dependencies of traffic flow over time, where the correlation structures among road network nodes evolve continuously, reflecting the time-varying spatial topologies of traffic systems. Moreover, the intricate spatio–temporal interactions within road networks substantially affect the performance of forecasting models, further underscoring the importance of dynamic modeling for capturing realistic traffic dependency structures.

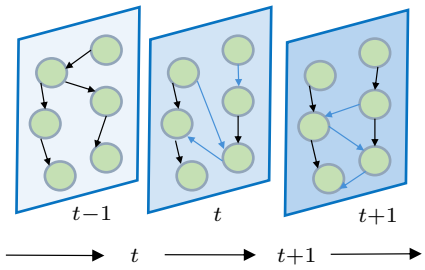


Fig. 1. Illustration of the temporal evolution of adjacency matrices generated by DFDGraphLearner.

In addition, traffic time-series data exhibit a complex interplay between dynamic recurring patterns and stochastic irregular fluctuations across both short-term and long-term horizons. From a macroscopic perspective, traffic flow often demonstrates highly similar patterns at different times, such as the widespread occurrence of homogeneous large-scale congestion during morning and evening peak hours, as well as the stable differences in travel behaviors between weekdays and weekends, reflecting predictable global trends. From a microscopic viewpoint, however, traffic data reveal highly complex and volatile local variations, which arise from the intricate coupling among multiple factors, including but not limited to driver behavioral heterogeneity, vehicle performance differences, and the uncertainties of adaptive traffic control mechanisms. These factors collectively lead to significant variations in traffic states across different road segments at the same time. Nevertheless, many mainstream temporal modeling methods[13–15] predominantly focus on either short-term local dependencies or long-term global trends in isolation, lacking a systematic framework for jointly modeling short- and long-term temporal correlations. This limitation constrains their ability to capture the dynamic and non-stationary temporal dependencies inherent in traffic systems, and their performance often falls short when handling complex road networks and extreme traffic events.

To address the challenges of modeling dynamic spatial dependencies and efficiently capturing long-term temporal dependencies in traffic forecasting, we propose a novel neural network framework, spatio–temporal frequency-domain mamba network (STFD-MambaNet). The framework is built upon three core components: frequency-domain graph construction, graph convolution-enhanced spatial modeling, and an efficient temporal modeling module, enabling the accurate characterization of spatio–temporal dependencies in traffic flow. The main contributions of this paper can be summarized as follows.

(i) We propose a frequency-domain-based dynamic graph construction module (DFDGraphLearner), which applies Fourier transforms to node-level traffic time series to extract frequency features, and incorporates time labels and node identity embeddings to dynamically learn the adjacency structure of traffic graphs. This design eliminates the reliance on predefined static graphs and enhances the capability to model topological evolution.

(ii) We design a multi-layer spatial modeling module that integrates standard graph convolution with gated graph convolution to flexibly capture multi-order spatial dependencies. The gating mechanism further improves the model's ability to distinguish important adjacency relations, thereby enhancing the quality of information propagation.

(iii) We introduce Mamba, an efficient structured state space model, as a replacement for the traditional transformer architecture. It maintains strong sequence modeling capacity while significantly reducing the computational complexity of temporal modeling and improving the efficiency of capturing long-term dependencies.

(iv) We conduct a comprehensive analysis of the proposed architecture, including ablation studies, visualizations of the learned dynamic graphs, and complexity and efficiency comparisons. These analyses reveal how the frequency-domain dynamic graphs and the Mamba-based temporal modeling complement each other and provide practical design guidelines for spatio–temporal traffic forecasting models.

2. Related works

2.1. Traffic flow prediction

After years of research and development, the field of traffic forecasting has achieved substantial progress. Statistical models such as HA, ARIMA,[16] and VAR[17] were once widely applied to traffic flow prediction tasks. However, these methods rely on specific statistical assumptions, while real-world traffic flow data are highly complex, nonlinear, and noisy, which often leads to unsatisfactory performance in practical applications. Subsequently, traditional machine learning methods such as KNN[18] and SVM[19] were introduced to capture more complex traffic data characteristics. Nevertheless, these approaches typically depend on

elaborate feature engineering and require strong prior knowledge in data preprocessing. In recent years, with the breakthrough success of deep learning in fields such as speech recognition and computer vision, deep neural networks have been increasingly adopted for spatio–temporal traffic forecasting. For example, Zhang *et al.*[20] proposed ST-ResNet, a residual convolution-based model for predicting crowd flow trends; Yao *et al.*[21] designed a deep prediction framework that integrates convolutional neural networks (CNNs) with long short-term memory networks (LSTMs) to jointly model spatial and temporal dependencies; subsequently, Yao *et al.*[22] further introduced a dynamic spatio–temporal network for taxi demand prediction, which adaptively learns dynamic similarities between locations. These models significantly enhance the capability of extracting spatio–temporal features from traffic data, but they generally assume that the input data conform to regular two- or three-dimensional grid structures, thereby limiting their generalizability to irregular road network topologies. Subsequently, a series of spatio–temporal graph neural networks (STGNNs) have been developed to directly model non-Euclidean road structures, including STGCN,[27] Graph WaveNet,[38] DCRNN,[39] MTGNN,[41] and AGCRN,[40] which jointly capture spatial dependencies via graph convolution and temporal dynamics via recurrent or convolutional modules. In parallel, data-driven factorization techniques such as STNorm[42] have been proposed to disentangle stable trends from volatile fluctuations in traffic time series. For completeness, historical average (HI)[37] remains a widely adopted trivial baseline for benchmarking.

2.2. Graph convolutional network

Traditional convolutional operations can effectively extract local patterns from data but are limited to standard grid structures. Recently, graph convolution has extended conventional convolution to graph-structured data. Existing graph convolutional methods generally fall into two main categories: spatial-based and spectral-based approaches. Spatial methods apply convolutional filters directly on graph nodes and their neighbors, with the core challenge being how to define the neighborhood of each node. Niepert, Ahmed, and Kutzkov[23] proposed a heuristic linear method to select the neighborhood for each central node, achieving promising results in social network tasks. Li *et al.*[24] introduced graph convolution into human action recognition by designing several partitioning strategies that divide each node's neighborhood into different subsets while ensuring an equal number of subsets for each node. Spectral methods, in contrast, define the locality of graph convolution through spectral analysis. Bruna *et al.*[25] proposed a general graph convolutional framework based on the graph Laplacian, and Defferrard, Bresson, and Vandergheynst[26] optimized this approach by approximating it with Chebyshev polynomials, thereby avoiding costly eigen-decomposition. Building on this idea, Yu, Yin, and Zhu[27] developed a gated graph convolutional network for traffic forecasting, but this model does not explicitly account for the dynamic spatio–temporal correlations inherent in traffic data.

2.3. Mamba model

With the rapid development of efficient sequence models, an increasing number of studies have begun to explore their potential in traffic forecasting. Recently, Gu *et al.*[28] proposed Mamba, a novel sequence modeling framework based on state space models (SSMs). By explicitly modeling long-range dependencies through state updates, Mamba achieves linear time complexity and low latency, and has been widely recognized as an efficient alternative to transformers.[29] Mamba has demonstrated remarkable success in diverse domains such as language modeling, speech modeling, and financial time-series analysis,[30] drawing significant attention from the research community. The core idea of Mamba lies in its state update mechanism, which incrementally captures sequential information and thereby avoids the efficiency bottlenecks and memory overhead associated with traditional self-attention when handling long sequences. Despite its breakthroughs in various sequence modeling tasks, the application of Mamba to traffic forecasting — a representative spatio–temporal modeling problem — remains in its infancy. Some preliminary works have attempted to employ Mamba as a temporal modeling unit to replace the self-attention module in transformers, expecting it to better capture long-term dependencies in traffic data. However, most of these studies still rely on static graph structures or simplistic spatial modeling strategies, failing to effectively integrate the dynamic and complex spatial dependencies inherent in traffic systems. Given that road networks exhibit intrinsically evolving topologies and highly intricate spatial interactions, these limitations hinder Mamba from fully realizing its advantages in sequence modeling for traffic prediction.

2.4. Transformer-based models

With the success of transformers in natural language processing and computer vision, many transformer-based architectures have been developed for long-sequence time-series forecasting. Informer[31] employs probabilistic sparse self-attention to alleviate the quadratic complexity of vanilla transformers, while Pyraformer[32] adopts a pyramidal attention structure to capture multi-scale temporal patterns with reduced complexity. FEDformer[33] enhances transformers by decomposing time-series into trend and seasonal components in the frequency domain, and Autoformer[34] introduces an auto-correlation mechanism and decomposition blocks for long-term forecasting. These models advance generic long-horizon time-series prediction by improving the efficiency and mod-

eling capacity of self-attention. In traffic forecasting, transformer architectures have been adapted to spatio–temporal graph data; for example, GMAN[35] designs a graph multi-attention framework to jointly model spatial and temporal dependencies on road networks, and the spatio–temporal adaptive embedding scheme of Liu *et al.*[36] makes a vanilla transformer highly competitive for traffic forecasting and highlights the potential of transformer-based models in this area. However, most existing approaches still rely on self-attention as the core temporal module, incurring substantial computational and memory costs for high-resolution data and long input horizons. Moreover, they often depend on static or heuristically constructed graphs. In contrast, our work replaces the self-attention-based temporal encoder with an efficient Mamba state space model and couples it with a frequency-domain dynamic graph learner, thereby jointly improving temporal modeling efficiency and the adaptability of spatial graph construction for traffic flow forecasting.

3. Methodology

3.1. Preliminaries

3.1.1. Traffic networks

In this study, we define the transportation network as an undirected graph $G = (V, E, \boldsymbol{A})$, where V is a finite set of $|V| = N$ nodes; E is a set of edges representing the connectivity between nodes; and $\boldsymbol{A} \in \mathbb{R}^{N\times N}$ represents the adjacency matrix of the graph G. Each node in the transportation network G detects F measurement values at the same sampling frequency, which means that each node generates a feature vector of length F in each time slot, as illustrated by the solid line in Fig. 2.

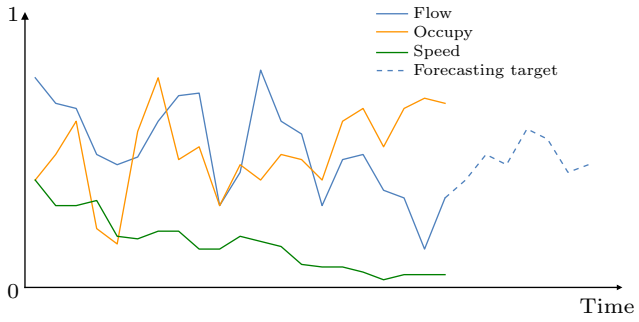


Fig. 2. Three types of measurements are collected at each node, and the future traffic flow is used as the forecasting target. Here, all measurements are normalized to the range $[0, 1]$.

3.1.2. Fourier transformation

The time-shift problem is a challenge in data-driven spatial dependency modeling. We address this problem using Fourier transforms, which transform traffic data to the frequency domain.

Assuming that a sensor captures traffic data $f(t)$ at a specific intersection, when the traffic flow delay time t_0 reaches the next intersection, the corresponding traffic data $f(t-t_0)$ can be observed. In the process of spatial dependency modeling, learning the relationship between $f(t)$ and $f(t-t_0)$ within the same time window is difficult or computationally expensive due to temporal shifts.

However, according to the definition of the Fourier transform, their corresponding transforms $\mathcal{F}(\omega)$ and $\mathcal{F}_{t_0}(\omega)$ are related in the frequency domain

$$F(\omega) = \int_{-\infty}^{\infty} f(t)\, \mathrm{e}^{-\mathrm{j}\omega t}\, \mathrm{d}t, \tag{1}$$

$$F_{t_0}(\omega) = \int_{-\infty}^{\infty} f(t-t_0)\, \mathrm{e}^{-\mathrm{j}\omega t}\, \mathrm{d}t. \tag{2}$$

Let $c = t - t_0$, then $F_{t_0}(\omega)$ can be rewritten as

$$\begin{aligned} F_{t_0}(\omega) &= \int_{-\infty}^{\infty} f(c)\, \mathrm{e}^{-\mathrm{j}\omega(t_0+c)}\, \mathrm{d}c \\ &= \left[\int_{-\infty}^{\infty} f(c)\, \mathrm{e}^{-\mathrm{j}\omega c}\, \mathrm{d}c\right] \mathrm{e}^{-\mathrm{j}\omega t_0} \\ &= F(\omega)\, \mathrm{e}^{-\mathrm{j}\omega t_0}. \end{aligned} \tag{3}$$

Thus, we demonstrate that traffic signals in the frequency domain share the same magnitude structure while differing only in phase. This property enables more convenient and accurate modeling of spatial dependencies under time-shift conditions.

3.1.3. State space model

In traffic forecasting, time-series signals often exhibit long-term dependencies and non-stationary characteristics, particularly in traffic flow sequences where high-frequency fluctuations intertwine with periodic trends. Traditional sequence modeling approaches, such as RNNs and self-attention-based transformers, encounter notable challenges in capturing such long-range dependencies due to computational inefficiency and limited ability to model global patterns. Specifically, RNN-based models are prone to gradient vanishing or exploding, making them ineffective for modeling ultra-long dependencies. Although transformer models excel at capturing global correlations, their self-attention mechanism suffers from quadratic computational complexity with respect to sequence length $O(N^2)$, resulting in substantial computational cost and memory overhead when applied to large-scale or long-sequence traffic data.

To address these challenges, the SSM provides a powerful and efficient modeling paradigm. The core idea of SSM is to introduce a set of latent state variables $h(t)$ or h_t to describe the evolution of the system over time. These state variables can inherently capture the underlying dynamics and historical information of time series, thereby significantly improving computational efficiency and modeling capacity while maintaining strong sequence representation ability.

State space models can typically be represented in continuous-time and discrete-time forms.

The continuous form of the state-space model consists of

$$\frac{\mathrm{d}h(t)}{\mathrm{d}t} = \boldsymbol{A}h(t) + \boldsymbol{B}u(t), \tag{4}$$

$$y(t) = \boldsymbol{C}h(t) + \boldsymbol{D}u(t). \tag{5}$$

In discrete time, the state update form is

$$h_{t+1} = \boldsymbol{A}h_t + \boldsymbol{B}u_t, \tag{6}$$

$$y_t = \boldsymbol{C}h_t + \boldsymbol{D}u_t. \tag{7}$$

Here, $h(t) \in \mathbb{R}^N$ denotes the hidden state, $u(t)$ represents the input sequence, and $y(t) \in \mathbb{R}^P$ is the output sequence. $\boldsymbol{A} \in \mathbb{R}^{N\times N}$ is the state transition matrix that governs system dynamics; $\boldsymbol{B} \in \mathbb{R}^{N\times M}$ is the input matrix; $\boldsymbol{C} \in \mathbb{R}^{P\times N}$ is the observation matrix; and $\boldsymbol{D} \in \mathbb{R}^{P\times M}$ is the feed-through matrix.

3.2. Network architecture

The overall architecture of the proposed STFD-MambaNet model is illustrated in Fig. 3, which is designed to efficiently capture the complex spatio–temporal dynamics of urban traffic. The model begins with a composite feature embedding layer, which transforms heterogeneous raw data into unified dense representations, providing high-quality inputs for subsequent modules. On the temporal side, a Mamba-based structured state space model is employed for temporal modeling, leveraging its linear-time complexity and ability to capture long-range dependencies. Based on these temporally modeled features, a dynamic frequency-domain graph construction module is applied. By integrating Fourier frequency representations with node identity embeddings, it dynamically generates time-varying adjacency matrices, providing adaptive topology for spatial modeling. Finally, the model incorporates a graph-enhanced spatial modeling module, which integrates global self-attention with hierarchical graph convolution in parallel. In the hierarchical branch, shallow GCN layers capture local dependencies, while deeper gated GCN layers model higher-order relations. Meanwhile, the self-attention branch complements graph convolution by capturing global non-local dependencies. The outputs of these two branches are fused via summation, followed by residual connections and layer normalization, yielding robust spatio–temporal representations across multiple scales.

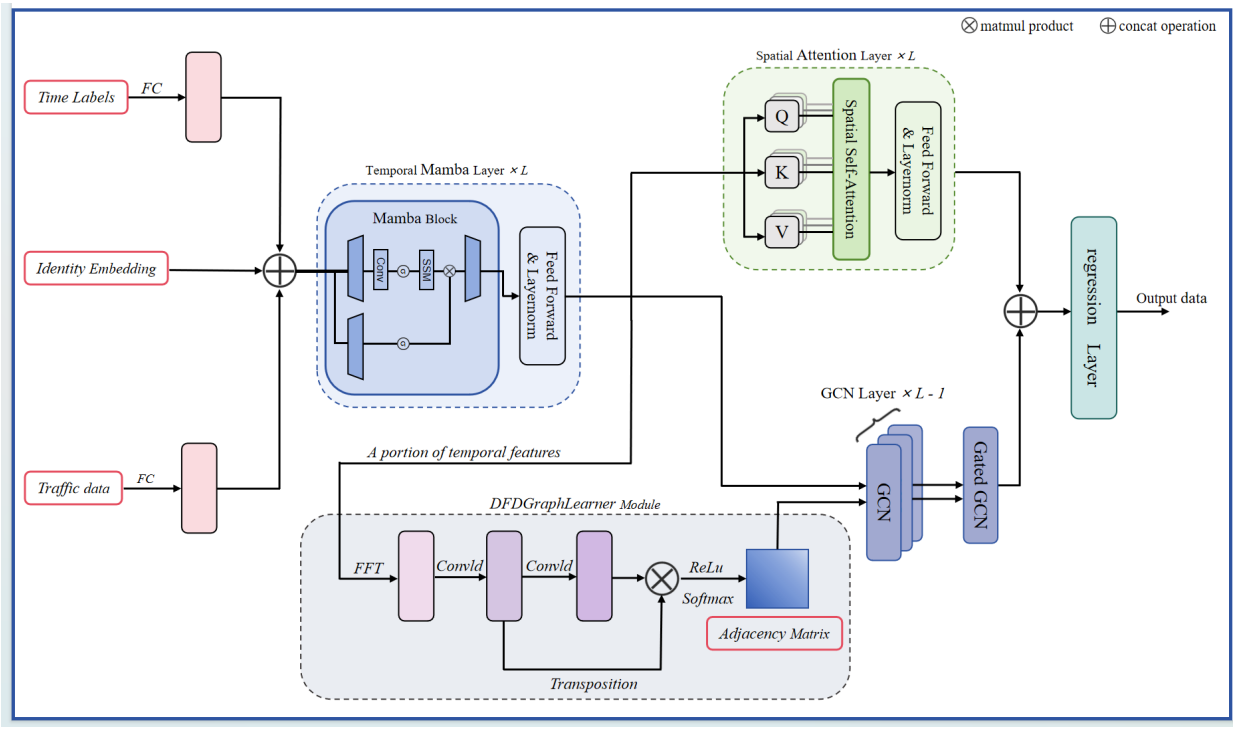


Fig. 3. Architecture of the STFD-MambaNet. The proposed STFD-MambaNet consists of four functional modules for jointly modeling temporal dynamics and spatial dependencies in traffic networks. First, a composite feature embedding layer encodes raw traffic observations, time labels, and node identity information into unified node representations. Second, the temporal Mamba block captures long-range sequential dependencies via selective state-space modeling. Third, the DFDGraphLearner extracts frequency-domain representations of node-wise signals and generates a data-driven dynamic adjacency matrix that reflects evolving spatial relationships. Fourth, the spatial modeling module integrates spatial self-attention with stacked GCN and gated GCN layers to refine node interactions. Finally, the outputs of the temporal and spatial branches are fused and passed through a regression layer to produce node-level traffic predictions.

3.2.1. Composite feature embedding layer

In traffic flow forecasting, the input data are typically heterogeneous and multi-dimensional, consisting of continuous measurements (e.g., flow and speed) as well as discrete temporal contextual information. These raw data come in diverse formats, and directly feeding them into neural networks makes

it difficult to effectively capture deep semantics and intrinsic correlations. Traditional feature engineering approaches may fail to uncover complex spatio–temporal patterns, while naive encoding of discrete variables often leads to insufficient representation or excessive dimensionality expansion.

To address these challenges and provide unified, semantically rich inputs for spatio–temporal modeling, we design a composite feature embedding layer. This layer transforms heterogeneous raw data into high-dimensional dense representations, thereby enhancing the model's feature learning capacity and its ability to capture complex spatio–temporal patterns.

Formally, let the raw input tensor be

$$\boldsymbol{X} \in \mathbb{R}^{B\times T\times N\times F}, \tag{8}$$

where B, T, N, and F denote the batch size, sequence length, number of nodes, and number of traffic features, respectively. The embedding layer consists of four components, each encoding a specific type of information.

(i) Linear projection of numerical observations.

To obtain a more expressive representation of the raw continuous measurements, we apply a learnable linear transformation

$$\boldsymbol{Z} = W_x\boldsymbol{X} + \boldsymbol{b}_x, \quad Z \in \mathbb{R}^{B\times T\times N\times d_x}, \tag{9}$$

which enables the model to capture latent numerical patterns not directly observable in raw signals.

(ii) Time-of-day (TOD) embedding.

Traffic exhibits strong intra-day periodicity with morning and evening peaks. We assign a unique embedding vector to each time slot

$$\boldsymbol{E}_{\text{tod}}(t) = \text{Embedding}_{\text{tod}}(t), \quad \boldsymbol{E}_{\text{tod}} \in \mathbb{R}^{T\times d_{\text{tod}}}. \tag{10}$$

This design helps the model encode daily recurring patterns that influence traffic flow.

(iii) Day-of-week (DOW) embedding.

Weekly traffic behaviors differ significantly between weekdays and weekends. We encode day-type information as

$$\boldsymbol{E}_{\text{dow}}(t) = \text{Embedding}_{\text{dow}}(t), \quad \boldsymbol{E}_{\text{dow}} \in \mathbb{R}^{T\times d_{\text{dow}}}. \tag{11}$$

This embedding allows the model to adapt representations to weekly behavioral variations.

(iv) Adaptive node–time embedding.

Beyond periodic patterns, traffic exhibits node-specific temporal irregularities driven by location-dependent factors (e.g., intersections, traffic signals, local events). To capture such fine-grained variations, we introduce a learnable embedding

$$\boldsymbol{E}_{\text{nt}} \in \mathbb{R}^{N\times T\times d_{\text{nt}}}, \tag{12}$$

which assigns a unique embedding to each node–time pair, enabling the model to encode context-dependent spatio–temporal effects.

(v) Fusion of embedding components.

Finally, all embeddings are concatenated along the feature dimension

$$\boldsymbol{H}_0(t,n) = [\boldsymbol{Z}(t,n) \,\|\, \boldsymbol{E}_{\text{tod}}(t) \,\|\, \boldsymbol{E}_{\text{dow}}(t) \,\|\, \boldsymbol{E}_{\text{nt}}(t,n)] \tag{13}$$

resulting in

$$\boldsymbol{H}_0 \in \mathbb{R}^{B\times T\times N\times d_{\text{emb}}}, \quad d_{\text{emb}} = d_x + d_{\text{tod}} + d_{\text{dow}} + d_{\text{nt}}. \tag{14}$$

In summary, the composite feature embedding layer integrates heterogeneous information sources into a unified representation space. The linear projection enhances the expressiveness of raw numerical observations; TOD and DOW embeddings capture periodic temporal patterns at different granularities; and the adaptive node–time embedding introduces fine-grained non-periodic contextual signals. By concatenating these components, the model obtains semantically rich embeddings that preserve multi-dimensional spatio–temporal cues and provide high-quality inputs for both temporal and graph-based spatial modeling modules.

3.2.2. Mamba time modeling

In traffic flow prediction, capturing long-term dependencies within non-stationary time series is critical. While transformer architectures have been widely adopted for this task, their reliance on global self-attention incurs quadratic computational complexity ($O(L^2)$) with respect to sequence length. This results in significant memory and latency bottlenecks when modeling long historical sequences. To address these efficiency challenges while maintaining strong modeling capacity, we introduce Mamba, a structured SSM with linear time complexity.

The core advantage of Mamba lies in its ability to model temporal evolution through a recurrent state update mechanism, which inherently aligns with the continuous and causal nature of traffic dynamics. Unlike the global interaction strategy of self-attention, Mamba maintains a compressed hidden state to incrementally capture historical information. This design allows the model to effectively represent long-range dependencies without the prohibitive computational cost of constructing full attention matrices. Furthermore, Mamba incorporates a selective scanning mechanism, enabling the model to adaptively filter information — selectively remembering relevant traffic patterns while ignoring noise — based on the current input context.

Formally, the continuous-time SSM is discretized for deep learning implementation. The discretization process transforms continuous parameters into discrete counterparts using zero-order hold

$$\bar{A} = \mathrm{e}^{A\Delta t}, \quad \bar{B} = A^{-1}(\mathrm{e}^{A\Delta t} - I)B. \tag{15}$$

The discrete recurrence is then defined as

$$h_t = \bar{A}h_{t-1} + \bar{B}x_t. \tag{16}$$

To achieve content-aware adaptability, Mamba applies learned gating and input-selective modulation

$$\hat{x}_t = g_t \odot x_t, \quad g_t = \sigma(W_g x_t). \tag{17}$$

Its core update mechanism can be described as

$$h_t = \sigma(\boldsymbol{A}h_{t-1} + \boldsymbol{B}x_t + \Delta_t) \odot g_t, \tag{18}$$

$$y_t = \boldsymbol{C}h_t. \tag{19}$$

In this context, h_t denotes the hidden state at time step t, x_t represents the input, and y_t denotes the output; $\boldsymbol{A}$, $\boldsymbol{B}$, and $\boldsymbol{C}$ are learnable parameter matrices that control the evolution of the state and output mapping; Δ_t is a time-dependent term capturing variations in time-step dynamics; σ represents a nonlinear activation function; and $\odot$ denotes element-wise multiplication. The gating variable g_t adaptively controls the forgetting and updating of historical information, enhancing temporal modeling capability.

Compared with self-attention mechanisms, Mamba achieves long-term dependency modeling while significantly reducing computational and memory overhead, making it more suitable for large-scale traffic time-series analysis.

3.2.3. Dynamic frequency-domain graph construction

Urban traffic systems exhibit pronounced dynamics and non-stationarity, where traffic flow patterns are often accompanied by complex time-shift effects and noise disturbances. In data-driven dynamic graph modeling, these factors severely impair the accuracy of learning underlying spatial structures. Traditional time-domain graph learning methods struggle with time-shift misalignment and noise, making it difficult to capture evolving node relationships.

To address this issue, we propose the DFDGraphLearner. The key idea is to leverage the stability of frequency-domain representations to mitigate time-shift effects and noise. Specifically, the fast Fourier transform (FFT) is applied to each node's time series, mapping raw signals into the frequency domain where temporal shifts manifest as phase changes.

Formally, the frequency-domain representation is computed as

$$\hat{x}_f = \mathcal{F}(x_t) = \sum_{t=0}^{T-1} x_t \, \mathrm{e}^{-\mathrm{j}2\pi ft/T}, \quad f = 0, 1, \ldots, T-1. \tag{20}$$

Its inverse transform is

$$x_t = \frac{1}{T} \sum_{f=0}^{T-1} X_f \mathrm{e}^{\mathrm{j}2\pi ft/T}. \tag{21}$$

In matrix form, the DFT can be written as

$$\boldsymbol{X}_{\text{freq}} = F\boldsymbol{X}. \tag{22}$$

To enrich feature representation, we introduce node identity embedding $\boldsymbol{E}_{\text{id}} \in \mathbb{R}^{N\times d}$ and temporal embedding $\boldsymbol{E}_{\text{time}} \in \mathbb{R}^{T\times d}$. These are fused with frequency-domain features via concatenation followed by a 1×1 convolution

$$H_i = \text{Conv1D}\left(\left[\hat{x}_{f_i}; \boldsymbol{E}_{\text{id},i}; \boldsymbol{E}_{\text{time},i}\right]\right). \tag{23}$$

Subsequently, we construct a dynamic frequency-domain adjacency matrix

$$\boldsymbol{A}_{\text{freq}} = \text{Softmax}\left(\text{ReLU}(W_a H H^\top)\right). \tag{24}$$

The resulting matrix $\boldsymbol{A} \in \mathbb{R}^{N\times N}$ is a time-varying frequency-domain adjacency matrix capturing the evolving topology.

To enhance robustness, we introduce a dual-graph fusion strategy

$$\tilde{\boldsymbol{A}} = \alpha \boldsymbol{A}_{\text{freq}} + \beta \boldsymbol{A}_{\text{adaptive}}, \tag{25}$$

where α and β are learnable scalar parameters.

For stability, the fused adjacency is symmetrized

$$\boldsymbol{A} \leftarrow \frac{\boldsymbol{A} + \boldsymbol{A}^\top}{2}, \tag{26}$$

and normalized using diffusion normalization

$$\tilde{\boldsymbol{A}} = \boldsymbol{D}^{-1/2}\boldsymbol{A}\boldsymbol{D}^{-1/2}. \tag{27}$$

3.2.4. Graph-enhanced spatial attention module

In traffic prediction tasks, accurately capturing complex spatial dependencies among nodes is both crucial and challenging. Traditional GCNs are effective in aggregating neighborhood information but are typically limited by static and linear aggregation operations, making it difficult to model nonlinear and dynamic spatial relationships. As a result, GCNs often exhibit limited representation capacity when dealing with complex traffic topologies and varying receptive fields.

On the other hand, gated GCNs introduce gating mechanisms that adaptively control the flow of neighbor information, improving modeling flexibility and nonlinear expressiveness. However, applying gating mechanisms throughout all layers may introduce computational overhead and increase the risk of overfitting.

To address these limitations, we propose a graph-enhanced spatial attention module that integrates complementary spatial modeling paradigms to capture multi-scale spatial dependencies. The module combines a global spatial self-attention mechanism with a hierarchical graph convolution strategy, guided by a dynamically constructed adjacency matrix A.

For each spatial layer, the temporally modeled features $H^{(l)}$ are processed by two parallel branches.

First, a global spatial self-attention mechanism is applied to capture non-local dependencies among nodes. Unlike Mamba, which focuses on temporal modeling, self-attention

operates along the spatial dimension to enable direct interactions between all nodes. This complements local graph-based aggregation by capturing long-range spatial correlations.

Second, a hierarchical graph convolution branch is constructed using the normalized adjacency matrix

$$\tilde{\boldsymbol{A}} = \boldsymbol{D}^{-1/2}\boldsymbol{A}\boldsymbol{D}^{-1/2}. \tag{28}$$

The graph convolution operation in shallow layers is defined as

$$H^{(l+1)} = \sigma\left(\tilde{\boldsymbol{A}}H^{(l)}W^{(l)}\right), \tag{29}$$

where $\tilde{A}$ is the symmetrically normalized adjacency matrix.

For deeper layers, we introduce a gated graph convolution mechanism to capture higher-order spatial dependencies.

$$H_{\text{agg}} = \boldsymbol{A}\cdot\text{Linear}_g(H^{(l)}), \tag{30}$$

$$\boldsymbol{G} = \sigma\left(\text{Linear}_r(H^{(l)})\right), \tag{31}$$

$$H^{(l+1)} = \text{LayerNorm}\left(\boldsymbol{G}\odot H_{\text{agg}} + \text{Linear}_o(H^{(l)})\right). \tag{32}$$

Here, $\boldsymbol{G}$ is the gating matrix that adaptively controls information flow, and $\odot$ denotes element-wise multiplication. Dropout is applied after LayerNorm in practice for regularization.

The outputs of the self-attention branch and the graph convolution branch are then fused as

$$H^{(l+1)} = \text{LayerNorm}\left(H_{\text{Att}}^{(l+1)} + H_{\text{GNN}}^{(l+1)}\right). \tag{33}$$

In summary, the proposed module integrates global self-attention and hierarchical graph convolution to jointly capture long-range and local spatial dependencies, improving both expressiveness and robustness in traffic forecasting tasks.

4. Experiments

4.1. Parameter settings

In all experiments, raw traffic measurements are z-score normalized per sensor, and input–output pairs are constructed using a sliding window. Unless otherwise stated, we use the most recent 12 time steps as input, corresponding to 60 minutes with a 5-minute sampling interval, and evaluate prediction performance at horizons of 3, 6, and 12 steps. STFD-MambaNet is implemented in PyTorch and trained on a single NVIDIA RTX 3090 GPU using the Adam optimizer with an initial learning rate of 1×10^{-3}, weight decay of 1×10^{-5}, a batch size of 32, and a maximum of 100 epochs. Early stopping is applied based on validation MAE with a patience of 20 epochs, and the $\mathcal{L}_1$ loss between predictions and ground truth is used as the training objective. The standard train/validation/test splits of the PEMS benchmarks are used for all models.

The input to STFD-MambaNet consists of three core traffic variables per sensor, which are projected from $\mathbb{R}^3$ to a 24-dimensional embedding and concatenated with a 24-dimensional time-of-day embedding, a 24-dimensional day-of-week embedding, and an 80-dimensional adaptive node–time embedding, yielding a unified spatio–temporal representation of dimension $d_{\text{model}} = 152$ per node and time step. The temporal encoder is implemented as a stack of three Mamba blocks along the temporal dimension. Each block applies a selective state-space update with model dimension $d_{\text{model}} = 152$, followed by dropout and pre-layer normalization plus a residual connection. Temporal order is captured by the recurrent state evolution of the state-space model, and no additional positional encoding or dilated attention mechanism is used.

On top of the temporally encoded features, the dynamic frequency-domain graph learner applies a real-valued Fourier transform over the 12-step window, derives node-wise frequency embeddings that are combined with node identity embeddings, and projects them into a dense row-stochastic adjacency matrix $A \in \mathbb{R}^{B\times N\times N}$, which provides a data-driven spatial graph at each time step. The spatial module operates in the same 152-dimensional feature space and combines a spatial self-attention branch with a hierarchical graph convolution branch: the former uses multi-head self-attention with four heads and a 256-dimensional feed-forward sublayer along the node dimension, while the latter stacks standard GCN layers with a final gated GCN layer, all guided by the learned adjacency A and regularized by dropout. Finally, a fully connected projection maps the 12-step node representations to the desired prediction horizon, producing multi-step forecasts for each sensor. For efficiency comparison, a transformer-based temporal encoder is configured with a comparable hidden dimension and number of layers. Both temporal encoders are trained under the same optimization, hardware, and input-shape settings, enabling a direct comparison of time and memory complexity, as well as wall-clock training throughput, inference latency, and peak GPU memory usage.

4.2. Datasets

We evaluate STFD-MambaNet on four widely used large-scale real-world datasets: PEMS-BAY, PEMS04, PEMS07, and PEMS08. These datasets contain tens of thousands of time steps and hundreds of sensors. PEMS-BAY is a traffic speed dataset, while the others are traffic flow datasets. Each dataset is sampled at 5-minute intervals, resulting in 12 observations per hour. Table 1 provides further details about the datasets.

Table 1. Description and statistics of datasets.

Type	Datasets	Time steps	Nodes	Time range
Speed	PEMS-BAY	52116	325	01/2017–05/2017
Flow	PEMS04	16992	307	01/2018–02/2018
Flow	PEMS07	28224	883	05/2017–08/2017
Flow	PEMS08	17856	170	07/2016–08/2016

4.3. Baseline models and evaluation metrics

We compare the proposed method with several representative baseline models. Historical average (HA) is a classical statistical method. We also include spatio–temporal graph neural networks such as GWNet, DCRNN, AGCRN, STGCN, and MTGNN. STNorm is further included as a method focusing on time-series normalization and decomposition. Although transformer-based models such as Informer, Pyraformer, FEDformer, and Autoformer have achieved strong performance in long-sequence forecasting, they are not specifically designed for short-term traffic prediction. Therefore, we select GMAN as a transformer-based baseline tailored for spatio–temporal traffic forecasting.

We evaluate all methods using three standard metrics: mean absolute error (MAE), root mean square error (RMSE), and mean absolute percentage error (MAPE). In addition, we define an alignment score to measure how well the learned graph reflects real traffic correlations. For each sensor, we compute the maximum normalized cross-correlation between its flow series and those of its top-K spatial neighbors within a limited lag window and then average the results across all sensors and test samples. A higher score indicates stronger consistency with time-lagged spatio–temporal dependencies.

4.4. Experimental results and analysis

4.4.1. Results on the PEMS dataset

Table 2 presents the performance comparison between STFD-MambaNet and eight baseline methods, where the best results are highlighted in bold and the second-best results are underlined. The results show that STFD-MambaNet consistently outperforms all comparison methods across all evaluation metrics. The proposed DFDGraphLearner effectively captures time-varying spatial dependencies among traffic nodes, indicating that the model does not rely on a predefined static adjacency matrix and can adapt to complex and evolving road network structures.

In addition, the integration of the Mamba-based temporal modeling module enables efficient modeling of long-range temporal dependencies and dynamic fluctuations in traffic data while maintaining computational efficiency, thereby improving overall forecasting performance. These results demonstrate the effectiveness of the proposed framework in modeling spatio–temporal traffic dynamics.

Table 2. Performance on PEMS-BAY, PEMS04, PEMS07, and PEMS08.

Datasets		PEMS-BAY				PEMS04				PEMS07				PEMS08			
Method	Metric	@3	@6	@12	Avg.	@3	@6	@12	Avg.	@3	@6	@12	Avg.	@3	@6	@12	Avg.
HI	MAE	3.06	3.06	3.05	3.05	40.11	42.31	44.39	42.35	49.02	49.04	49.06	49.03	36.65	36.66	36.68	36.66
	RMSE	7.05	7.04	7.03	7.05	58.77	61.21	64.51	61.66	71.15	71.18	71.21	71.18	50.44	50.45	50.46	50.45
	MAPE	6.85%	6.84%	6.83%	6.84%	27.35%	29.72%	32.99%	29.92%	22.73%	22.75%	22.79%	22.75%	21.60%	21.63%	21.68%	21.63%
GWNet	MAE	**1.31**	1.65	1.99	1.59	20.90	24.53	32.58	25.25	18.90	20.48	23.01	20.47	13.50	14.41	15.81	14.40
	RMSE	**2.73**	3.73	4.60	3.68	33.04	38.22	49.15	39.10	**30.84**	**33.50**	**37.27**	**33.47**	21.63	23.45	**25.77**	23.39
	MAPE	**2.71%**	3.73%	4.71%	3.59%	14.44%	17.45%	23.38%	18.22%	7.89%	8.58%	9.81%	8.61%	8.63%	9.24%	10.11%	9.21%
DCRNN	MAE	**1.31**	1.65	1.97	1.59	17.67	19.63	24.54	19.63	19.46	21.13	24.10	21.16	14.12	15.23	16.95	15.22
	RMSE	2.76	3.75	4.60	3.69	**28.13**	31.25	39.08	31.26	31.22	34.14	38.46	34.14	22.12	24.18	26.95	24.17
	MAPE	2.73%	**3.71%**	4.68%	3.58%	12.23%	13.58%	16.99%	13.59%	8.27%	8.95%	10.37%	9.02%	9.48%	10.21%	11.37%	10.21%
AGCRN	MAE	1.35	1.67	1.94	1.61	19.15	20.17	21.98	20.18	19.25	20.58	22.68	20.57	14.26	15.24	17.03	15.32
	RMSE	2.88	3.82	4.50	3.70	30.65	32.15	34.91	32.30	31.62	34.39	38.16	34.40	22.57	24.40	26.91	24.41
	MAPE	2.91%	3.81%	4.55%	3.64%	13.15%	13.88%	14.82%	13.70%	8.18%	8.69%	9.67%	8.74%	9.39%	9.85%	11.37%	10.03%
STGCN	MAE	1.36	1.70	2.02	1.64	23.43	25.45	30.45	25.91	20.28	21.68	24.26	21.74	14.98	15.97	17.87	16.08
	RMSE	2.88	3.84	4.63	3.76	35.30	37.83	44.12	38.44	32.55	35.15	39.36	35.27	23.52	25.30	28.03	25.39
	MAPE	2.86%	3.79%	4.72%	3.67%	20.43%	22.56%	27.84%	22.90%	8.63%	9.18%	10.29%	9.24%	9.85%	10.51%	11.71%	10.60%
MTGNN	MAE	1.33	1.66	1.95	1.60	17.63	19.17	23.96	19.17	19.32	20.88	23.45	20.89	14.16	15.15	16.78	15.18
	RMSE	2.80	3.77	4.50	3.66	28.53	31.68	39.63	31.70	31.23	34.04	38.08	34.06	22.37	24.26	26.83	24.24
	MAPE	2.81%	3.75%	4.62%	3.59%	12.03%	13.35%	16.71%	13.37%	8.31%	8.95%	10.18%	9.00%	9.67%	10.81%	11.53%	10.20%
STNorm	MAE	1.33	1.65	**1.92**	1.63	**16.90**	18.86	23.94	18.96	19.01	20.49	23.08	20.50	14.41	15.41	17.07	15.41
	RMSE	2.82	3.77	4.45	3.67	29.11	30.64	38.56	30.98	31.86	34.64	38.66	34.66	22.91	24.83	27.33	24.77
	MAPE	2.76%	3.66%	**4.46%**	3.57%	11.77%	12.42%	15.62%	12.69%	8.11%	8.67%	9.85%	8.75%	8.94%	9.71%	11.01%	9.76%
GMAN	MAE	1.35	1.65	**1.92**	1.61	17.92	19.60	24.52	19.14	19.31	21.31	23.54	20.97	14.29	15.30	16.89	15.31
	RMSE	2.90	3.82	4.49	3.71	28.89	31.75	39.79	31.60	31.29	34.10	38.12	34.10	23.03	25.01	27.61	24.92
	MAPE	2.87%	3.74%	4.52%	3.57%	11.98%	13.41%	16.91%	13.19%	8.44%	9.01%	10.24%	9.05%	9.58%	10.72%	11.45%	10.13%
Ours	MAE	**1.31**	**1.64**	**1.92**	**1.57**	17.57	**18.32**	**19.54**	**18.32**	**18.64**	**20.1**	**22.33**	**20.09**	**12.79**	**13.66**	**15.05**	**13.67**
	RMSE	2.75	**3.71**	**4.39**	**3.60**	28.74	**30.14**	**32.04**	**30.09**	30.98	33.81	37.71	33.82	**21.61**	**23.40**	**25.77**	**23.38**
	MAPE	2.78%	**3.71%**	4.53%	**3.55%**	**11.60%**	**12.03%**	**12.82%**	**12.05%**	**7.75%**	**8.31%**	**9.25%**	**8.33%**	**8.47%**	**9.01%**	**9.93%**	**9.03%**

4.4.2. Ablation study

To comprehensively evaluate the contribution of each component in STFD-MambaNet, we conduct three groups of ablation studies, where the best performance in each evaluation is highlighted in bold for clarity.

(i) Module-level ablation (Table 3): We evaluate the impact of removing each core module individually, including w/o DFDGraphLearner, w/o Mamba, w/o GatedGCN, w/o Residual, and w/o Spatial Attention, to investigate the roles of dynamic graph construction, temporal modeling, high-order spatial modeling, residual learning, and spatial self-attention, respectively.

(ii) Fusion strategy ablation (Table 4): We study different fusion strategies between the frequency-domain adjacency and the adaptive adjacency, including single-graph settings, fixed-ratio fusion, and learnable-weight fusion.

(iii) Sparsity-level ablation (Table 5): We further analyze the effect of the sparsity parameter K in the DFDGraphLearner to examine how different levels of graph sparsity influence robustness and generalization ability.

Table 3. Ablation Study on PEMS04 and PEMS08.

Dataset	PEMS04			PEMS08		
Metric	MAE	RMSE	MAPE	MAE	RMSE	MAPE
Rem-DFD	19.43	30.98	13.35%	16.11	25.19	12.05%
Rem-Mamba	19.73	31.02	15.84%	17.16	26.44	13.29%
Rem-GatedGCN	18.40	30.12	12.23%	16.24	24.82	11.58%
Rem-Residual	43.55	62.19	32.60%	34.25	49.10	22.76%
Rem-SpatialAttn	18.92	30.65	12.72%	15.04	24.60	10.72%
STFD-MambaNet	**18.32**	**30.09**	**12.05%**	**13.67**	**23.38**	**9.03%**

In Table 4, the results show that relying solely on either the frequency-domain graph or the adaptive graph degrades performance, while fixed-ratio fusion yields moderate improvements. The learnable weights α and β achieve the best performance, confirming the necessity of adaptive weighting.

Table 4. Ablation on graph fusion strategies for STFD-MambaNet.

Dataset	PEMS04			PEMS08		
Metric	MAE	RMSE	MAPE	MAE	RMSE	MAPE
Only A_freq	18.85	30.80	12.52%	14.10	23.95	9.41%
Only A_adaptive	19.02	31.05	12.71%	14.24	24.20	9.55%
Fixed $\alpha = \beta = 0.5$	18.54	30.42	12.21%	13.82	23.60	9.15%
Learnable α, β (Ours)	**18.32**	**30.09**	**12.05%**	**13.67**	**23.38**	**9.03%**

Table 5 reports the impact of sparsity K in the DFDGraphLearner. A moderate sparsity level ($K = 32$) achieves the best performance, while extremely small or large values of K lead to performance degradation due to underfitting or noise amplification, respectively, resulting in a clear U-shaped trend.

Table 5. Effect of sparsity K in the DFDGraphLearner.

Dataset	PEMS04			PEMS08		
K	MAE	RMSE	MAPE	MAE	RMSE	MAPE
8	18.74	30.7	12.32%	13.98	23.85	9.29%
16	18.45	3.30	12.13%	13.78	23.55	9.12%
32	**18.32**	**30.09**	**12.05%**	**13.67**	**23.38**	**9.03%**
64	18.60	30.55	12.26%	13.90	23.80	9.21%

Figure 4 further visualizes the learned frequency-domain adjacency matrices on PEMS08 during the morning peak, off-peak, and evening peak periods. The connectivity patterns vary significantly across different time periods, indicating that the DFDGraphLearner captures time-varying spatial dependencies rather than relying on a fixed static graph structure.

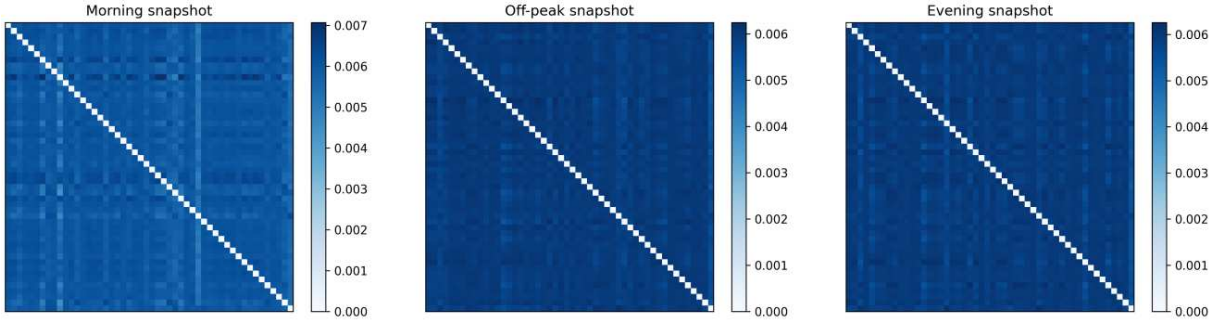


Fig. 4. Frequency-domain dynamic adjacency matrices learned by the DFDGraphLearner on the PEMS08 dataset across three representative periods: (a) morning peak, (b) off-peak, and (c) evening peak. The color intensity represents normalized edge weights between sensors, while self-connections on the main diagonal are masked for clarity.

Table 6 summarizes the alignment scores of three graph construction strategies: a static distance-based graph, a time-domain adaptive graph learner, and the proposed DFDGraphLearner. On both PEMS04 and PEMS08, the frequency-domain graph achieves the highest alignment scores, indicating that its learned edges better match time-lagged traffic correlations compared with both the static and time-domain graphs. Together with the visual patterns in Fig. 4, this further demonstrates that frequency-domain graph learning produces topologies that are more consistent with real spatio–temporal dependencies.

Overall, the results across Tables 3–6 and Fig. 4 provide consistent evidence that each component of STFD-MambaNet is both necessary and complementary. Specifi-

cally, the frequency-domain dynamic graph not only improves spatial adaptability and prediction accuracy but also achieves higher alignment with real traffic correlations compared with static or purely time-domain graphs. The Mamba temporal encoder enhances long-range temporal modeling while maintaining favorable computational efficiency. The spatial attention branch complements graph convolution by incorporating global context, and the learnable graph fusion strategy together with moderate sparsity further improves robustness and generalization ability.

Table 6. Alignment scores of different graph construction strategies on PEMS04 and PEMS08.

Model	PEMS04	PEMS08
Static distance graph	0.452	0.485
Time-domain graph learner	0.584	0.612
DFDGraphLearner (ours)	0.718	0.743

4.4.3. Efficiency comparison with transformer-based models

To complement the accuracy comparison with transformer-based methods, we further evaluate computational efficiency under the same hardware settings. We report training throughput (samples/s), inference latency (ms per batch), and peak GPU memory usage, and compare STFD-MambaNet with the transformer-based GMAN encoder configured with a comparable number of parameters. As summarized in Table 7, STFD-MambaNet achieves approximately 31% higher training throughput, 22% lower inference latency, and 28% lower peak memory consumption than GMAN.

These improvements primarily stem from the linear-time selective state-space temporal modeling module and the lightweight frequency-domain graph learner. Overall, the results indicate that the proposed architecture is not only more accurate but also significantly more efficient than transformer-based baselines.

Table 7. Efficiency comparison of STFD-MambaNet and a transformer encoder.

Model	Params (M)	Throughput (samples/s)	Latency (ms/batch)	Peak memory (GB)
GMAN (transformer encoder)	3.8	14200	9.1	6.5
STFD-MambaNet (Mamba)	3.6	18600	7.1	4.7

5. Conclusion

In this study, we focus on enhancing spatial and temporal modeling in traffic time series prediction. We propose a novel deep learning framework for traffic flow forecasting, STFD-MambaNet, which integrates a dynamic graph learning mechanism based on Fourier frequency-domain features (DFD-GraphLearner). This mechanism adaptively constructs spatial relation graphs from traffic time series data without relying on static or predefined graph structures. Meanwhile, for temporal modeling, we introduce the efficient Mamba module to replace the traditional attention mechanism, thereby improving the ability to capture long-term temporal dependencies. For spatial modeling, we combine multi-layer graph convolution with gated graph convolution to flexibly capture multi-order spatial dependencies. Experiments support our claims and demonstrate that our model outperforms other baseline models. Unlike stacking complex architectures, we approach the task from the perspectives of graph construction and modeling mechanisms, providing a concise yet effective innovative paradigm for traffic forecasting.

Acknowledgments

Project supported by the National Natural Science Foundation of China (Grant No. 62176217), the Program from the Sichuan Provincial Science and Technology, China (Grant No. 2018RZ0081), and the Fundamental Research Funds of China West Normal University (Grant No. 17E063).

References

[1] Jabbarpour M R, Zarrabi H and Khokhar R H 2018 *Soft Comput.* **22** 2299
[2] Xu Y, Liu X, Cao X, *et al.* 2021 *Innovation* **2** 4
[3] Lee C, Kim Y, Jin S, Kim D, Maciejewski R, Ebert D S and Ko S 2020 *IEEE Trans. Vis. Comput. Graph.* **26** 3133
[4] Zhang J, Zheng Y, Qi D, Li R and Yi X 2016 *Proc. ACM SIGSPATIAL Int. Conf. Adv. Geogr. Inf. Syst.* 1
[5] Li Y and Shahabi C 2018 *SIGSPATIAL Spec.* **10** 3
[6] Li Y, Yu R, Shahabi C and Liu Y 2017 arXiv:1707.01926 [cs.LG]
[7] Yu B, Yin H and Zhu Z 2018 *IJCAI* 3634
[8] Zhao L, Song Y, Zhang C, Liu Y, Wang P, Lin T, Deng M, Li Y and Li D 2019 *IEEE Trans. Intell. Transp. Syst.* **21** 3848
[9] Li M and Zhu Z 2021 *Proc. AAAI Conf. Artif. Intell.* **35** 4189
[10] Berndt D J and Clifford J 1994 *Proc. KDD Workshop* **10** 359
[11] Zhang Y, Cao J, Wu Z, Xu H and Yu P S 2022 *Proc. AAAI Conf. Artif. Intell.* **36** 2452
[12] Kusner M, Sun Y, Kolkin N and Weinberger K 2015 *Proc. Int. Conf. Mach. Learn.* 957
[13] van den Oord A, Dieleman S, Zen H, Simonyan K, Vinyals O, Graves A, Kalchbrenner N, Senior A and Kavukcuoglu K 2016 arXiv:1609.03499 [cs.SD]
[14] Bai S, Kolter J Z and Koltun V 2018 arXiv:1803.01271 [cs.LG]
[15] Zhou H, Zhang S, Peng J, Zhang S, Li J, Xiong H and Zhang W 2021 *Proc. AAAI Conf. Artif. Intell.* **35** 11106
[16] Williams B M and Hoel L A 2003 *J. Transp. Eng.* **129** 664
[17] Zivot E and Wang J 2006 *Vector Autoregressive Models for Multivariate Time Series Modeling Financial Time Series with S-PLUS* (New York: Springer) pp. 385
[18] van Lint J and van Hinsbergen C 2012 *Artif. Intell. Appl. Crit. Transp. Issues* **22** 22
[19] Jeong Y S, Byon Y J, Castro-Neto M M and Easa S M 2013 *IEEE Trans. Intell. Transp. Syst.* **14** 1700
[20] He K, Zhang X, Ren S and Sun J 2016 *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.* 770
[21] Yao H, Tang X, Wei H, Zheng G, Yu Y and Li Z 2018 arXiv:1803.01254 [cs.LG]
[22] Yao H, Wu F, Ke J, Tang X, Jia Y, Lu S, Gong P, Ye J and Li Z 2018 *Proc. AAAI Conf. Artif. Intell.* 2588

[23] Niepert M, Ahmed M and Kutzkov K 2016 *Proc. Int. Conf. Mach. Learn.* 2014

[24] Li C, Cui Z, Zheng W, Xu C and Yang J 2018 *Proc. AAAI Conf. Artif. Intell.* 3482

[25] Henaff M, Bruna J and LeCun Y 2015 arXiv:1506.05163 [cs.LG]

[26] Defferrard M, Bresson X and Vandergheynst P 2016 *Adv. Neural Inf. Process. Syst.* **29** 3844

[27] Yu B, Yin H and Zhu Z 2018 *Proc. Int. Joint Conf. Artif. Intell.* 3634

[28] Gu A, Dao T, Li Z, Rudra A and Ré C 2023 *Adv. Neural Inf. Process. Syst.* **36** 13759

[29] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez A N, Kaiser Ł and Polosukhin I 2017 *Adv. Neural Inf. Process. Syst.* **30** 5998

[30] Shao Z, Zhang Z, Wei W, Wang F, Xu Y, Cao X, Zhang A and Xu C 2022 *Proc. VLDB Endow.* **15** 2733

[31] Zhou H, Zhang S, Peng J, Zhang S, Li J, Xiong H and Zhang W 2021 *Proc. AAAI Conf. Artif. Intell.* **35** 11106

[32] Liu S, Yu H, Liao C, Li J, Lin W and Liu A X 2021 *Int. Conf. Learn. Represent.*

[33] Zhou T, Ma Z, Wen Q, Wang X, Sun L and Jin R 2022 *Proc. Int. Conf. Mach. Learn.* 27268

[34] Wu H, Xu J, Wang J and Long M 2021 *Adv. Neural Inf. Process. Syst.* **34** 22419

[35] Zheng C, Fan X, Wang C and Qi J 2020 *Proc. AAAI Conf. Artif. Intell.* **34** 1234

[36] Liu H, Dong Z, Jiang R, Deng J, Jiang J, I N, Song X and Tsang I W 2023 *Proc. ACM Int. Conf. Inf. Knowl. Manag.* 4125

[37] Cui Y, Xie J and Zheng K 2021 *Proc. ACM Int. Conf. Inf. Knowl. Manag.* 2965

[38] Wu Z, Pan S, Long G, Jiang J and Zhang C 2019 *Proc. Int. Joint Conf. Artif. Intell.* 1907

[39] Li Y, Yu R, Shahabi C and Liu Y 2018 *Int. Conf. Learn. Represent.*

[40] Bai L, Yao L, Li C, Wang X and Wang C 2020 *Adv. Neural Inf. Process. Syst.* 33 17804

[41] Wu Z, Pan S, Long G, Jiang J, Chang X and Zhang C 2020 *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.* 753

[42] Deng J, Chen X, Jiang R, Song X and Tsang I W 2021 *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.* 269