

Article

Frequency-Aware and Interactive Spatial-Temporal Graph Convolutional Network for Traffic Flow Prediction

Guoqing Teng ¹, Han Wu ¹, Hao Wu ², Jiahao Cao ¹ and Meng Zhao ^{1,*}¹ School of Computer Science and Engineering, Chongqing University of Science and Technology, Chongqing 400044, China; jahaur2001@outlook.com (J.C.)² School of Electrical and Electronic Engineering, North China Electric Power University, Beijing 102206, China

* Correspondence: zhaomeng@cqust.edu.cn

Abstract

Accurate traffic flow prediction is pivotal for intelligent transportation systems; yet, existing spatial-temporal graph neural networks (STGNNs) struggle to jointly capture the long-term structural stability, short-term dynamics, and multi-scale temporal patterns of road networks. To address these shortcomings, we propose FISTGCN, a Frequency-Aware Interactive Spatial-Temporal Graph Convolutional Network. FISTGCN enriches raw traffic flow features with learnable spatial and temporal embeddings, thereby providing comprehensive spatial-temporal representations for subsequent modeling. Specifically, it utilizes an interactive dynamic graph convolutional block that generates a time-evolving fused adjacency matrix by combining adaptive and dynamic adjacency matrices. It then applies dual sparse graph convolutions with cross-scale interactions to capture multi-scale spatial dependencies. The gated spectral block projects the input features into the frequency domain and adaptively separates low- and high-frequency components using a learnable threshold. It then employs learnable filters to extract features from different frequency bands and adopts a gating mechanism to adaptively fuse low- and high-frequency information, thereby dynamically highlighting short-term fluctuations or long-term trends. Extensive experiments on four benchmark datasets demonstrate that FISTGCN delivers state-of-the-art predictive accuracy while maintaining competitive computational efficiency.

Keywords: traffic prediction; spatial-temporal fusion; frequency domain; multi-scale interaction

Academic Editor: George Drosatos

Received: 7 September 2025

Revised: 11 October 2025

Accepted: 13 October 2025

Published: 21 October 2025

Citation: Teng, G.; Wu, H.; Wu, H.; Cao, J.; Zhao, M. Frequency-Aware and Interactive Spatial-Temporal Graph Convolutional Network for Traffic Flow Prediction. *Appl. Sci.* **2025**, *15*, 11254. <https://doi.org/10.3390/app152011254>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

With accelerating urbanization and the large-scale deployment of intelligent transportation systems (ITSs), traffic flow prediction has become a critical technological foundation for urban traffic management and scheduling. By analyzing and modeling historical traffic data, accurately forecasting future traffic conditions can help alleviate congestion, optimize signal timing, improve road efficiency, and provide scientific support for public travel and decision-making in traffic management [1]. In practical applications, traffic flow data are continuously collected by diverse sensors deployed throughout road networks (e.g., cameras, GPS), forming high-dimensional time series with complex spatial-temporal dependencies. Influenced by road topology, participant behavior, weather, and unexpected incidents, traffic data exhibit pronounced nonlinearity, non-stationarity, and dynamic correlations, posing significant challenges to high precision modeling and forecasting [2]. For instance, traffic flow varies significantly at different times of the day, with peak hours causing abrupt high-frequency fluctuations. Furthermore, events such as accidents or road closures lead to sudden disruptions, further intensifying the dynamic nature of traffic data.

Traditional traffic flow-prediction methods, such as Auto-Regressive Integrated Moving Average (ARIMA) [3] and Kalman Filtering (KF) [4], are limited in handling heterogeneous temporal patterns and spatial dependencies inherent in traffic flow data. These models are designed for linear, stationary time series, making them unsuitable for modeling the non-linear and non-stationary nature of traffic flow, which includes periodic traffic variations and sudden fluctuations. Moreover, they do not capture spatial dependencies between sensors in the network, which are crucial for accurate prediction [5]. While effective for certain simple temporal relationships, these models are limited in their ability to handle heterogeneous temporal patterns and complex spatial dependencies inherent in traffic flow series. With the rapid advances in deep learning, data-driven approaches—owing to their powerful feature-extraction and modeling capacities—have become mainstream in traffic-prediction research. Temporally, recurrent neural networks (RNNs) [6] and their variants, such as long short-term memory (LSTM) [7] and gated recurrent units (GRU) [8], have demonstrated effectiveness in modeling dynamic temporal dependencies. Spatially, convolutional neural networks (CNNs) are adept at extracting local features on regular grids. However, real-world road networks are typically non-Euclidean, which inherently limits grid-based CNNs [9] in capturing complex spatial dependencies. To overcome this, research has shifted towards spatial-temporal graph neural networks (STGNNs) [10–12], which integrate GNN-based spatial modules with temporal modeling modules such as RNNs or temporal convolutional networks (TCNs), enabling joint modeling of traffic spatial-temporal patterns. Despite their progress in predictive accuracy, several challenges remain.

Spatially, existing Spatial-Temporal Graph Neural Networks (STGNNs) rely on adjacency matrices to express spatial dependencies among nodes in traffic networks. Mainstream construction approaches include predefined graphs, adaptive graphs, and dynamic graphs. Predefined graphs, which are constructed based on geographic distances or node similarity, often introduce structural bias. This bias arises when topological information or domain knowledge is insufficient, leading to inaccurate representations of spatial dependencies in traffic networks. Additionally, predefined graphs fail to adapt to the dynamic nature of traffic conditions [13,14], limiting their ability to capture real-time changes in traffic flows caused by factors such as accidents, construction, or weather events. Adaptive graphs [15,16], on the other hand, learn spatial dependencies directly from data, which allows them to uncover hidden connectivity patterns. However, they still face significant limitations. While they are more flexible, these graphs are typically static during the testing phase, which means they do not adapt to evolving traffic conditions. This restricts their ability to capture sudden shifts in spatial dependencies, ultimately reducing their effectiveness in dynamic environments.

In recent years, dynamic graph methods that generate adjacency matrices from real-time data have emerged [17–19], enabling models to dynamically adapt spatial structures based on real-time traffic conditions and thereby more precisely model time-varying node relationships. Nevertheless, most dynamic graph methods only address single-structure assumptions, lacking sufficient consideration for balancing the long-term structural stability of traffic networks and their short-term dynamic variations. As illustrated in Figure 1, while traffic networks remain relatively stable over the long term, their spatial dependencies can exhibit significant temporal fluctuations due to factors such as traffic volume and weather. Additionally, traffic networks inherently possess multi-scale spatial dependencies, and current GNN-based methods often fall short in effectively integrating interactions across different spatial scales.

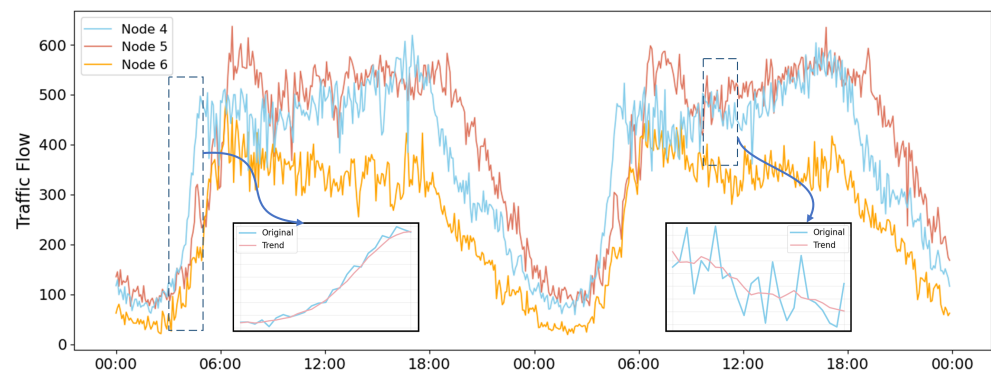


Figure 1. Traffic flow for Nodes 4, 5, and 6 in the PeMS08 dataset over two consecutive days. The main plots show that spatial correlations among nodes remain generally stable over long periods, while both spatial and temporal dimensions exhibit pronounced short-term fluctuations. Off-peak hours are dominated by high-frequency variations, whereas peak and valley periods are characterized by rapid increases and decreases in flow, respectively. Insets display representative segments, comparing the original series with its smoothed trend component to highlight long-term patterns and short-term fluctuations.

Temporally, traffic flow data generally exhibit complex and highly heterogeneous temporal characteristics. This heterogeneity is manifested in both long-term periodic trends (e.g., rush hours, weekday-weekend differences) and high-frequency fluctuations caused by incidents or weather. As illustrated in Figure 1, traffic flow varies significantly at different times of the day, with peak hours causing abrupt high-frequency fluctuations. Furthermore, events such as accidents or road closures lead to sudden disruptions, further intensifying the dynamic nature of traffic data. Traffic signals essentially represent a nonlinear superposition of trends and short-term fluctuations, with significant interactions among their components and dynamic shifts in their relative dominance depending on time and conditions. For example, in cases of severe congestion or accidents, sudden events become the dominant factors, whereas under normal conditions, periodic and trend factors play a leading role. However, prevailing approaches mostly use a single unified structure [20–22], treating all temporal information uniformly and neglecting the essential differences among temporal scales and their dynamic evolution. This not only limits the ability to represent multi-scale dependencies, but also obscures the dominant role of short-term disturbances versus long-term trends at different times, thus constraining predictive performance in complex, dynamic scenarios.

The objective of this research is to propose a novel Frequency-Aware and Interactive Spatial-Temporal Graph Convolutional Network (FISTGCN) that addresses the limitations of current STGNNs by simultaneously capturing dynamic spatial dependencies, multi-scale temporal features, and incorporating frequency-domain analysis to separate long-term trends and short-term fluctuations. Specifically, during the embedding stage, a gating mechanism integrates raw traffic signals with temporal and spatial embeddings, thereby providing richer representations for subsequent spatial-temporal modeling tasks. One of the main challenges with conventional graph convolutional models is their inability to capture both long-term stability and short-term dynamics simultaneously, while also handling complex multi-scale spatial interactions. To overcome this, we introduce a dynamic adjacency matrix generation method combined with a spatial interaction mechanism. By incorporating both static structural attributes and time-varying features, we construct a time-evolving fused adjacency matrix that effectively models spatial dependencies that change over time. Our architecture employs a dual-layer graph convolution scheme. In the first layer, dynamic spatial edges are partitioned based on their sparsity levels. This is followed by two separate graph convolutions that capture spatial dependencies at different scales. The cross-scale interaction

mechanism between these layers not only strengthens the capture of deep spatial correlations but also creates a positive feedback loop, which facilitates efficient information exchange and enhances the mutual reinforcement of spatial features.

Furthermore, to tackle the challenge of separating and integrating multi-scale temporal features in traffic data, FISTGCN introduces a frequency-domain modeling strategy. By applying the Fast Fourier Transform (FFT) to project original time series into the frequency domain and employing learnable thresholds for adaptive separation of low- and high-frequency components, we introduce learnable filters to extract features from different frequency bands. This enables layered modeling of short-term fluctuations and long-term trends. Finally, a gated fusion unit dynamically integrates features from different frequencies, adaptively adjusting the weight of trends and fluctuations over time. This design significantly enhances the model's adaptability to dynamic shifts in dominant components and improves its performance in representing and predicting heterogeneous, diverse temporal scenarios. In summary, our main contributions are as follows:

1. In the spatial dimension, we propose a dynamic adjacency matrix generation method that integrates both adaptive and dynamic matrices, together with a spatial interaction mechanism. Leveraging a dual-layer graph convolutional structure with varying sparsity, our approach models complex, multi-scale spatial dependencies and enable positive feedback learning for multi-scale interactions.
2. In the temporal dimension, we design a Gated Spectral Block (GSB) to uncover multi-scale temporal features and dynamically switch between dominant components. Leveraging FFT-based spectral decomposition and learnable filters, our module enables adaptive separation, deep feature extraction, and gated fusion of low- and high-frequency components, capturing the intricate interplay between trends and fluctuations.
3. We conduct extensive experiments on four widely-used real-world traffic datasets. Results demonstrate that FISTGCN achieves superior predictive performance while maintaining competitive computational efficiency.

The rest of this paper is organized as follows. In Section 2, we review related works in the field of traffic flow prediction and spatial-temporal graph neural networks. Section 3 presents the problem formulation and introduces a frequency-aware and interactive spatial-temporal graph convolutional network for traffic flow forecasting. In Section 4, we conduct experiments on real-world datasets and compare the prediction performance with several existing methods. The discussion of the results is presented in Section 5. Finally, Section 6 concludes the paper and outlines promising directions for future research.

2. Related Works

2.1. Traffic Flow Prediction

Early research on traffic flow prediction primarily relied on statistical approaches such as historical averages (HA) [23], ARIMA, and vector autoregressive (VAR) [24]. While these methods offer certain advantages in handling time series data, they are fundamentally based on linearity assumptions and thus struggle to capture the complex nonlinear patterns inherent in traffic flow data. To overcome these limitations, various traditional machine learning methods have been progressively introduced into the traffic flow-prediction domain. These approaches better model the nonlinear relationships in data, thereby improving both accuracy and robustness. Typical examples include support vector regression (SVR) [25] and k-nearest neighbors (KNN) [26], which predict future traffic flow by learning data distributions and patterns. However, conventional machine learning methods generally rely on manual feature selection and construction, a process that is not only time-consuming but may also restrict the generalizability of the models.

2.2. Graph Convolution Network

In recent years, deep learning-based methods have made remarkable progress in traffic flow prediction, owing to their capabilities for automatic feature extraction and efficient representation learning. Compared with traditional machine learning models, deep learning frameworks can more effectively capture the complex spatial-temporal dependencies in traffic data and substantially reduce reliance on manual feature engineering. Early studies often combined CNNs with RNNs or LSTMs to jointly extract spatial-temporal features from traffic data. CNNs, with their local receptive field mechanism, excel at capturing spatial dependencies among adjacent road segments; however, their local convolution operations limit their capacity to model long-range spatial dependencies. RNNs and LSTMs, while suitable for modeling dynamic temporal dependencies, are prone to issues such as vanishing gradients and difficulty in learning long-term dependencies, resulting in diminished performance on long-range sequence modeling tasks. Therefore, conventional deep models that integrate CNNs and RNNs still face challenges in predicting traffic flow over long horizons and in scenarios with complex spatial relationships.

To address these issues, graph neural networks (GNNs) have been widely adopted for traffic flow prediction in recent years. Leveraging their strengths in modeling non-Euclidean data, GNNs effectively capture spatial dependencies among nodes and propagate information through graph convolution operations. For example, Yu et al. proposed STGCN [13], combining temporal and spatial convolutions to capture spatial-temporal correlations in traffic data. This model applies 1D convolution for temporal modeling and employs the Chebyshev graph convolution network (ChebNet) for spatial modeling, thereby effectively extracting spatial-temporal features. STSGCN [27] employs graph convolution to capture spatial information related to each node while simultaneously modeling temporal dependencies, thus directly capturing heterogeneity in local spatial-temporal graphs and significantly improving predictive accuracy. DCRNN [14] models spatial correlations in traffic data as a diffusion process on directed graphs and replaces the original GRU's linear layers with GCNs to further enhance the modeling of spatial-temporal dependencies.

However, early STGNN models such as STGCN and DCRNN rely on predefined graph structures, which limit their capacity to capture potential latent spatial dependencies within traffic networks. To overcome this, researchers have introduced methods based on learnable adaptive adjacency matrices. For instance, Graph WaveNet [15] and AGCRN [16] dynamically learn and update the graph structure during training, thereby more effectively capturing hidden spatial correlations. Nonetheless, these adaptive adjacency matrices typically remain fixed after training, rendering them incapable of adapting to dynamic changes during inference. To address this limitation, several data-driven dynamic graph generation methods have been developed, which generate dynamic graph structures based on the input traffic data. For example, STDE-DGCN [28] constructs dynamically weighted adjacency matrices by fusing similarity and distance graphs via Gaussian kernels, thereby effectively capturing the dynamic spatial-temporal correlations among sensors and improving GCN performance. Subsequent works, such as ADCT-Net [29], DDSTGCN [30], and GDGCN [31], further investigate the dynamics of spatial graphs by adopting various approaches to model dynamic relationships, emphasizing the critical importance of capturing time-varying spatial relationships for achieving accurate traffic flow prediction.

In addition to GCN-based models, attention-based graph modeling approaches have also demonstrated outstanding performance and have been widely applied to traffic flow forecasting. For instance, DSAN-ST [32] addresses spatial and temporal dependencies using a multi-spatial attention (MSA) mechanism and a dynamic attention encoder (DAE), where MSA separates spatial features and captures key relationships, and DAE dynamically extracts temporal information relevant to the target location. Bi-STAT [33] leverages

an adaptive transformer to model both spatial and temporal dependencies. GMAN [34] employs a pure attention mechanism to independently capture spatial-temporal correlations from both temporal and spatial perspectives. PDFormer [35] uses two masking matrices to obtain spatial dependencies for both short-term and long-term views, introducing propagation delay awareness into the attention design and achieving notable results. ProSTformer [36] proposes a progressive spatial-temporal self-attention mechanism with a parallel transformer architecture to jointly fuse spatial-temporal features, further enhancing predictive performance.

As shown in Table 1, we categorize the representative STGNNs based on their graph relation states outlined in the related work and detail the methods employed in their spatial and temporal modules. Despite the significant progress made by existing models, they are either constrained by static graph structures or neglect the interaction of multi-scale spatial features, typically focusing on spatial characteristics at a single scale. Moreover, in terms of temporal modeling, these models do not explicitly differentiate between high-frequency and low-frequency components for multi-scale modeling. To address these issues, this paper proposes a new frequency-aware and interactive spatial-temporal graph convolutional network, aiming to more effectively capture dynamic spatio-temporal correlations and fill the gap in existing models in this regard.

Table 1. Classification We have reviewed the tables and confirm the removal of vertical lines as per the guidelines. Thank you! of spatial-temporal modeling methods.

Model	Temporal Module	Spatial Module	Graph Relations
STGCN	CNN	GNN	Static
STSGCN	GCN	GCN	Static
DCRNN	RNN	GNN	Static
GraphWaveNet	TCN	GCN	Static
AGCRN	RNN	GCN	Static
STDE-DGCN	CNN	GCN	Static
ADCT-Net	Attention	GCN	Static
DDSTGCN	TCN	GCN	Dynamic
GDGCN	TGCN	GCN	Dynamic
Bi-STAT	Attention	Attention	Dynamic
GMAN	Attention	Attention	Dynamic
PDFormer	Attention	Attention	Dynamic
DSAN-ST	Attention	Attention	None
ProSTformer	Attention	CNN	None

3. Notations, Definitions and Preliminaries

The notation commonly used throughout this paper is summarized in Table 2.

3.1. Problem Statement

Traffic flow forecasting aims to predict future traffic states by utilizing historical data together with the spatial structure of road networks. Specifically, a set of N sensor nodes deployed across the road network records traffic flow measurements over the past T time intervals, aggregated as $X = [x_1, x_2, \dots, x_T] \in \mathbb{R}^{T \times N \times C}$, where $x_t \in \mathbb{R}^{N \times C}$ denotes the sensor observations at time t , and each measurement contains $C = 1$ channel representing traffic flow. At the same time, the road system is represented as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, A)$, in which $\mathcal{V}$ denotes the set of sensor nodes, $\mathcal{E}$ denotes the set of edges between nodes, and $A \in \mathbb{R}^{N \times N}$ is a learnable adjacency matrix.

Table 2. Frequently used notations.

Notations	Definitions
$\mathcal{G}$	The traffic spatial graph.
V	The researched road segments sensors set.
N	Number of sensor nodes.
E	The connectivity among road sensors.
C	The traffic features dimension.
A	The adjacency matrix of the network $\mathcal{G}$.
T	The number of input historical steps.
T'	The number of output predictable steps.
D	The corresponding degree matrix of the adjacency matrix.
Δ	The normalized Laplacian matrix.
PE	The position embedding of positions in the series.
E_1, E_2	The learnable node embedding.
d	The hidden dimension.
a	The feature dimensions of learnable node embedding.
L	Number of layers in the spatial-temporal encoder.
f	The traffic flow-prediction function.

The forecasting task is thus formalized as predicting future traffic flow states based on both the previous T time steps of traffic data and a road graph $\mathcal{G}$. This can be mathematically formulated as learning a function f that maps the combined input of historical observations and the road graph $\mathcal{G}$ to the traffic flow at the subsequent T' time steps:

$$[\mathbf{X}_{(t-T):t}, \mathcal{G}] \xrightarrow{f} \mathbf{X}_{(t+1):(t+T')} \quad (1)$$

3.2. Preliminaries: Fast Fourier Transform

The Fourier Transform is a classical mathematical technique that converts a signal from its original time domain to the frequency domain, thereby enabling the decomposition of a complex signal into sinusoidal components at different frequencies. For a continuous signal $\mathbf{X}(t)$, the Fourier Transform is defined as follows:

$$\mathbf{X}_f = \mathcal{F}(\mathbf{X}) = \int_{-\infty}^{\infty} \mathbf{X}(t) e^{-j2\pi ft} dt \quad (2)$$

where $\mathcal{F}(\cdot)$ denotes the Fourier Transform operator, and $\mathbf{X}_f$ represents the signal in the frequency domain. Correspondingly, the Inverse Fourier Transform provides a mechanism to reconstruct the original time-domain signal from its frequency-domain representation. The inverse operation is defined as follows:

$$\mathbf{X}(t) = \mathcal{F}^{-1}(\mathbf{X}_f) = \int_{-\infty}^{\infty} \mathbf{X}_f(f) e^{j2\pi ft} df \quad (3)$$

where $\mathcal{F}^{-1}(\cdot)$ denotes the Inverse Fourier Transform operator.

In practical deep learning applications, especially for discrete time-series data, the Discrete Fourier Transform and its inverse are usually employed. These transformations allow for the analysis and manipulation of sequential signals in the frequency domain, making it possible to extract periodicity, distinguish high-frequency fluctuations, and recover the processed features back to the temporal domain for further modeling.

4. Methodology

The overall architecture of FISTGCN is illustrated in Figure 2. After the gated embedding layer aggregates the traffic flow data with spatial and temporal embeddings, multiple spatial-temporal encoder layers are employed to extract spatial-temporal patterns, which are then passed to the output layer for final prediction. The detailed implementation of each module will be comprehensively introduced in the following subsections.

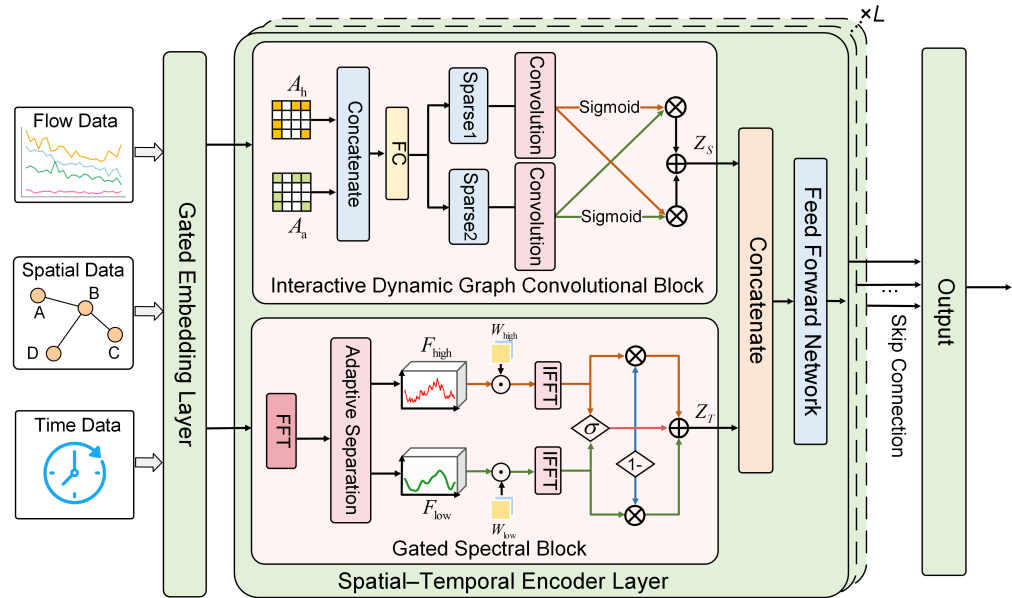


Figure 2. The overall architecture of the FISTGCN, which contains L identical layers.

4.1. Gated Embedding Layer

The gated embedding is responsible for integrating various types of initial traffic information by introducing spatial and temporal embeddings and subsequently reducing redundancy through the gating mechanism, thereby enhancing the model's ability to capture spatial-temporal features of traffic data. First, the spatial embedding learns the structural correlations among nodes within the road network by employing the graph Laplacian matrix, thereby characterizing the spatial coupling of traffic flow across different nodes. Specifically, we first compute the normalized Laplacian matrix as follows:

$$\Delta = I - D^{-\frac{1}{2}} A D^{-\frac{1}{2}}, \quad (4)$$

where A denotes the adjacency matrix, and D represents the degree matrix. Next, we perform eigenvalue decomposition, $\Delta = U \Lambda U^T$. The spatial graph Laplacian embedding, $E_{spe} \in \mathbb{R}^{N \times d_{st}}$, is obtained by selecting the k eigenvectors in U associated with the smallest non-trivial eigenvalues in Λ , followed by a linear projection to preserve the global structure of the graph.

Next, the temporal embedding emphasizes the periodic and trend-based characteristics inherent in traffic data. On a weekly scale, traffic flow patterns show noticeable differences between weekdays and weekends. On a daily scale, traffic flow typically exhibits significant peaks during morning and evening rush hours, while remaining relatively stable during off-peak periods. We construct two matrices representing weekly and daily periodicities, denoted as $E_w(t)$ and $E_d(t)$, where the functions $w(t)$ and $d(t)$ map the timestamp t to weekly indices (1 to 7) and minute indices (1 to 1440), respectively. After applying linear transformations to $E_w(t)$ and $E_d(t)$, the temporal periodic embeddings $E_p \in \mathbb{R}^{T \times d_{st}}$ are generated.

Finally, the raw traffic flow input X is transformed into E_f via a linear layer and combined with spatial-temporal embeddings. Subsequently, a gating mechanism is employed to integrate multi-dimensional spatial-temporal information, thereby enhancing the model's capacity to effectively represent complex traffic patterns. The resulting representation $X_e \in \mathbb{R}^{T \times N \times d}$, where $d = d_f + d_{st}$, is computed as follows:

$$X_{fus} = \oplus(E_f, (E_{spe} + E_p)), \quad (5)$$

$$X_e = \text{Conv}(\sigma(\text{Conv}(X_{fus})) \odot \text{GELU}(\text{Conv}(X_{fus}))), \quad (6)$$

$$X_e = X_e + \text{PE}(X_e), \quad (7)$$

where Conv denotes 1×1 convolution, PE represents temporal positional encoding, $\oplus$ denotes the concatenation operation, σ is the sigmoid activation function, and $\odot$ denotes the Hadamard product, which refers to an element-wise multiplication of two matrices of the same dimensions, where each element in the resulting matrix is the product of the corresponding elements in the original matrices.

4.2. Spatial-Temporal Encoder Layer

The spatial-temporal encoder layer primarily consists of two components: an interactive dynamic graph convolutional block for spatial features and a gated spectral block for temporal features.

4.2.1. Interactive Dynamic Graph Convolutional Block

We design an Interactive Dynamic Graph Convolutional Block (IDGCB) as the spatial modeling module, which constructs a dynamically adjacency matrix by integrating adaptive and dynamic adjacency matrices, thereby enabling dynamic modeling of compound spatial dependencies and supporting the interactive extraction of multi-scale spatial features to further uncover deeper spatial correlations.

Dynamic Graph Convolution

Considering that traffic networks exhibit structural stability in the long term but are affected by external disturbances such as traffic volume and weather in the short term, the spatial relationships between nodes exhibit significant time-varying. To address this, we construct a time-evolving, dynamically fused adjacency matrix that balances the long-term stability and short-term dynamics of the traffic network. The static attributes of each node (e.g., type, points of interest (POIs)) are encoded as high-dimensional vectors $E_1 \in \mathbb{R}^{N \times a}$ and $E_2 \in \mathbb{R}^{N \times a}$, serving as adaptive embeddings to represent the basic spatial features of the nodes. In addition, dynamic traffic signals are integrated with spatial-temporal embeddings to form X_e , which characterizes the real-time state of each node. Subsequently, by computing the similarity between node representations, we generate the adaptive adjacency matrix and the dynamic adjacency matrix separately, as defined below:

$$A_{a,ij} = \frac{\exp(E_{1,i}E_{2,j}^\top)}{\sum_{j=1}^N \exp(E_{1,i}E_{2,j}^\top)} \quad (8)$$

$$A_{h,ij} = \frac{\exp(X_{e,i}X_{e,j}^\top)}{\sum_{j=1}^N \exp(X_{e,i}X_{e,j}^\top)} \quad (9)$$

where i denotes the row index. The resulting two adjacency matrices, $A_a \in \mathbb{R}^{N \times N}$ and $A_h \in \mathbb{R}^{N \times N}$, represent the spatial associations between nodes from the perspectives of long-term structural stability and short-term dynamics, respectively. To integrate information

from both matrices, we concatenate them and apply a fully connected layer to obtain the final dynamically fused adjacency matrix:

$$A_f = \oplus(A_a, A_h)W_F \quad (10)$$

Based on this dynamically fused adjacency matrix, the dynamic graph convolution operation is defined as follows:

$$DGCN(X_e) = A_f X_e W_D \quad (11)$$

Spatial Interaction Learning

To jointly model multi-scale spatial features, we propose a spatial-interaction learning strategy. As illustrated in Figure 2, the IDGCB comprises dual layers of sparse dynamic graph convolutions with different sparsity levels, enabling the effective capture of spatial dependencies across multiple scales. The first layer SDGCN employs higher sparsity to focus on fine-grained, local spatial patterns, while the second layer adopts lower sparsity to capture broad, long-range spatial dependencies. The core computation of SDGCN is given by:

$$SDGCN1 = \mathcal{T}k_1(A_f)X_e W_{G1} \quad (12)$$

$$SDGCN2 = \mathcal{T}k_2(A_f)X_e W_{G2} \quad (13)$$

$$[\mathcal{T}k(A_f)]_{ij} = \begin{cases} (A_f)_{ij}, & (A_f)_{ij} \geq t_i \\ 0, & \text{otherwise} \end{cases} \quad (14)$$

where $\mathcal{T}k(\cdot)$ denotes a learnable top- k selection operator, with t_i denoting the k -th largest value in the i -th row of A_f , and k dynamically controlling the level of sparsity. Furthermore, we introduce a feature interaction mechanism, allowing the output of each graph convolution layer to influence the feature-extraction process of the other layer and promote information exchange across different scales. The interaction process is as follows:

$$Z_S = \sigma(SDGCN1(X_e)) \odot SDGCN2(X_e) + \sigma(SDGCN2(X_e)) \odot SDGCN1(X_e) \quad (15)$$

where $Z_S \in \mathbb{R}^{T \times N \times d/2}$ represents the output of IDGCB, $\odot$ denotes the Hadamard product, and σ denotes the sigmoid activation function. This spatial interaction learning strategy enables multi-scale spatial information to mutually reinforce the extraction of spatial features, thereby enhancing the overall feature representation.

4.2.2. Gated Spectral Block

The Gated Spectral Block is designed based on the spectral characteristics of traffic flow signals, which moves beyond conventional single-scale temporal modeling by introducing dual frequency-domain branches and a gated fusion mechanism. Leveraging the Fast Fourier Transform (FFT), the original time series is projected into the frequency domain. A learnable threshold adaptively separates the spectrum into low- and high-frequency components, followed by two dedicated filters that learn features at specific temporal scales. The gated fusion strategy dynamically integrates multi-scale spectral features, enabling effective multi-scale spectral modeling of temporal data. By separating the temporal components into low- and high-frequency bands, GSB facilitates targeted learning of both long-term trends (such as periodic traffic patterns) and short-term fluctuations (such as sudden traffic events or accidents). This component-based approach allows the model to adjust its focus dynamically,

depending on the dominant temporal patterns at any given moment, thereby significantly enhancing prediction ability across varying traffic conditions.

Fast Fourier Transformations

Given a discrete time series $x[n]$, the frequency-domain representation $x[k]$ is obtained by performing an FFT along the temporal dimension, as shown in Equation (2). This process effectively reveals the energy distribution of traffic signals across different frequency components, facilitating the capture of variations at multiple temporal scales. Similarly, for X_e , its representation is computed as follows:

$$F = \mathcal{F}[X_e] \in \mathbb{C}^{N \times F' \times d} \quad (16)$$

where $\mathcal{F}$ denotes the 1D FFT operation, and F' is the length of the resulting frequency-domain representation. The FFT is applied to decompose the traffic flow signals into frequency components, allowing the model to focus on short-term variations (high-frequency components) during incidents or rush hours, while capturing long-term trends (low-frequency components) such as daily or weekly patterns. This separation improves the model's ability to predict diverse traffic conditions by focusing on relevant frequencies at different times.

Adaptive Separation of Low and High Frequencies

In the frequency domain, high-frequency components typically correspond to short-term fluctuations and abrupt events in traffic flow, while low-frequency components reflect long-term trends and periodic variations. To enable modeling across different temporal scales, GSB incorporates a learnable threshold for adaptive separation of low- and high-frequency components. Specifically, based on the squared magnitude of frequency components $P = |F|^2$, a threshold τ is automatically learned, generating a binary mask M that partitions the frequency spectrum into low- and high-frequency segments:

$$M = \begin{cases} 0, & \text{if } P > \tau \\ 1, & \text{otherwise} \end{cases} \quad (17)$$

The threshold τ is optimized automatically during model training via backpropagation, allowing the separation criterion to adapt to the non-stationarity of different datasets and temporal signals.

$$F_{\text{low}} = F \cdot M \quad (18)$$

$$F_{\text{high}} = F \cdot (1 - M) \quad (19)$$

For the separated low-frequency components F_{low} and high-frequency components F_{high} , GSB employs two sets of learnable filters to fully exploit the sequential characteristics in each frequency band. Let W_{low} and W_{high} denote the learnable filters for the low- and high-frequency components, respectively. The filtering process can be formulated as follows:

$$F_L = W_{\text{low}} \odot F_{\text{low}} \quad (20)$$

$$F_H = W_{\text{high}} \odot F_{\text{high}} \quad (21)$$

Next, the low- and high-frequency components are mapped back to the temporal domain via the inverse fast Fourier transform (IFFT), yielding time-domain feature representations for different frequency bands.

$$S_L = \mathcal{F}^{-1}[F_L] \in \mathbb{R}^{T \times N \times d} \quad (22)$$

$$S_H = \mathcal{F}^{-1}[F_H] \in \mathbb{R}^{T \times N \times d} \quad (23)$$

Gated Fusion Unit

The contribution of low- and high-frequency features to prediction results dynamically varies across time and scenarios. For example, high-frequency components are more critical for short-term forecasting during rush hours or sudden events, while low-frequency components are more informative for overall trend estimation during stable periods. To address this, GSB incorporates a gated fusion unit that adaptively aggregates low- and high-frequency features through dynamic weighting. The fusion process is formulated as follows:

$$gate = \sigma(S_L W_L + S_H W_H) \quad (24)$$

$$Z_T = (gate \odot S_L + (1 - gate) \odot S_H) W_Z \quad (25)$$

where $Z_T \in \mathbb{R}^{T \times N \times d/2}$ represents the output of GSB, while $W_L \in \mathbb{R}^{d \times d}$, $W_H \in \mathbb{R}^{d \times d}$, and $W_Z \in \mathbb{R}^{d \times d/2}$ are learnable parameters.

4.2.3. Fusion

The spatial features extracted by the interactive dynamic graph convolution block and the temporal features captured by the gated spectral block are concatenated along the feature dimension and fused via a linear mapping. This process enables the model to integrate spatial and temporal information, fully leveraging their complementarity.

Specifically, the spatial-temporal fusion process can be formalized as follows:

$$Z_{ST} = \oplus(Z_S, Z_T) W_Z \quad (26)$$

$$Z = Z_{ST} + \text{FFN}(\text{LayerNorm}(Z_{ST})) \quad (27)$$

where $Z \in \mathbb{R}^{T \times N \times d}$ represents the output of the spatial-temporal encoder, and FFN is a feed-forward neural network composed of two fully connected layers with GELU activation functions, which enhance the model's capacity for nonlinear representation.

4.3. Output Layer

We employ skip connections composed of 1×1 convolutions to aggregate the outputs of each spatial-temporal encoder, transforming them into the final hidden state $Z_{sk} \in \mathbb{R}^{T \times N \times d_{sk}}$, where d_{sk} denotes the feature dimension of the skip connection. Subsequently, the hidden state Z_{sk} is transformed by two successive 1×1 convolutional layers to produce the multi-step prediction results:

$$\hat{X} = \text{Conv}_2(\text{Conv}_1(Z_{sk})) \quad (28)$$

where $\hat{X} \in \mathbb{R}^{T' \times N \times C}$ denotes the predicted values for the next T' time steps.

4.4. Loss Function

For training FISTGCN, we adopt the Huber loss function, which is defined as follows:

$$\mathcal{L}(\hat{y}, y) = \begin{cases} \frac{1}{2}(\hat{y} - y)^2, & |\hat{y} - y| \leq \delta \\ \delta(|\hat{y} - y| - \frac{1}{2}\delta), & |\hat{y} - y| > \delta \end{cases} \quad (29)$$

where $\hat{y}$ and y are the predicted and actual traffic flow values, respectively, and δ is the hyperparameter to balance squared error.

5. Experimental Results and Analysis

5.1. Datasets

Four publicly available datasets from the California Traffic Management System, PeMS03, PeMS04, PeMS07, and PeMS08 [27], are used in this study to validate the predictive performance of the proposed model. Each dataset contains 288 data points per day, collected in real time at 5-minute intervals by measuring freeway traffic in California. These datasets were selected since they cover multiple freeway networks and capture diverse traffic conditions. This provides a comprehensive representation of traffic patterns. The detailed statistics of the datasets are presented in Table 3. Each dataset was split into training (60%), validation (20%), and test (20%) subsets for model evaluation.

Table 3. The detailed information of datasets.

Dataset	Sensors	Edges	Time Range	Time Steps
PeMS03	358	547	09/2018–11/2018	26,208
PeMS04	307	340	01/2018–02/2018	16,992
PeMS07	883	866	05/2017–08/2017	28,224
PeMS08	170	295	07/2016–08/2016	17,856

5.2. Experimental Setup

5.2.1. Experimental Settings

Experiments were conducted on a workstation equipped with an NVIDIA A800 GPU. The time step interval was set to five minutes, with both input and output sequence lengths fixed at 12 time steps. Model hyperparameters, including the hidden dimension d and the number of encoder layers L , were selected from $\{16, 32, 64, 128\}$ and $\{4, 5, 6, 7\}$, respectively, based on validation set performance. In the final configuration, the hidden dimension d was set to 64 and the number of spatial-temporal encoder layers L to 6. For the top- k sparsity parameters in the IDGCB, specific values were set for each dataset: for PeMS08, the large and small sparsity values were set to 0.6 and 0.9, respectively; for PeMS07, they were set to 0.7 and 0.9; and for both PeMS03 and PeMS04, the values were set to 0.5 and 0.8, respectively. A batch size of 16 was utilized during training. The model was trained for 300 epochs using the AdamW optimizer. The version of the AdamW optimizer used is from the PyTorch library (version 1.10.1), which includes the AdamW implementation. To mitigate overfitting, an early stopping strategy was employed, where training would automatically terminate if no performance improvement on the validation set was observed over 50 consecutive epochs. Additionally, random seeds were fixed for all experiments to ensure reproducibility. Model performance was evaluated using the following three metrics: Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Mean Absolute Percentage Error (MAPE).

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (30)$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (31)$$

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right| \times 100\% \quad (32)$$

Here, y_i and $\hat{y}_i$ denote the actual and predicted traffic flow values at node i for a given time step, respectively.

5.2.2. Baselines

We compare FISTGCN with a series of previous baseline models, which can be broadly categorized as follows:

Traditional time series-prediction models: including HA [23], VAR [24], and SVR [25].

GNN-based models: such as DCRNN [14], STGCN [13], ASTGCN [37], GraphWaveNet [15], MTGNN [38], AGCRN [16], STSGCN [27], STFGNN [39], DGCRN [17], DSTAGNN [40], D²STGNN [41], MegaCRN [42], and PDG2Seq [43].

Attention-based models: including GMAN [34], ASTGNN [44], and PDFormer [35].

- HA method forecasts traffic by using the average of historical traffic data.
- VAR is a statistical model that captures temporal dependencies between multiple traffic-related variables to predict future traffic flow.
- SVR uses a linear support vector machine to predict traffic conditions based on past data patterns.
- DCRNN models traffic flow as a diffusion process, utilizing diffusion GCN combined with GRU to effectively capture the spatial-temporal dependencies inherent in traffic data.
- STGCN integrates GCN with 1D convolution to model both spatial and temporal dependencies in traffic data.
- ASTGCN introduces spatial and temporal attention mechanisms to effectively capture the spatial-temporal correlations in traffic flow.
- GraphWaveNet employs gated TCN stacked with GCN to model spatial-temporal dependencies.
- MTGNN proposes a novel mix-hop propagation layer and a dilated inception layer to effectively capture spatial and temporal dependencies in time series data.
- AGCRN employs adaptive graph convolution in place of the GRU's original linear transformation layer to more effectively capture spatial-temporal correlations in traffic flow.
- GMAN leverages distinct attention modules along the temporal and spatial dimensions to effectively capture spatial-temporal correlations.
- STSGCN employs a meticulously designed synchronous spatial-temporal modeling mechanism to effectively capture complex, localized spatial-temporal correlations.
- STFGNN proposes a robust hierarchical method that captures local spatial relations by leveraging temporal context from traffic states and trends.
- ASTGNN employs a temporal self-attention mechanism to capture sequence dynamics with global receptive fields and a dynamic graph convolution module to adaptively model spatial correlations.
- DGCRN employs hyper-networks to extract dynamic features from node attributes, generating filter parameters at each time step.
- DSTAGNN constructs a data-driven dynamic graph and uses a GNN with multi-head attention and multi-scale gated convolutions to model spatial-temporal dependencies.
- D²STGNN decouples diffusion and inherent traffic patterns and captures spatial-temporal dependencies through dynamic graph learning.
- MegaCRN employs a meta-graph learner with a Meta-Node Bank to dynamically generate spatial-temporal graph structures for traffic modeling.
- PDFormer captures dynamic spatial dependencies and models time delay with dual graph masks and a delay-aware module.
- PDG2Seq extracts periodic features using time points as indices, and combines these with dynamic traffic features to construct a Periodic Dynamic Graph for enhanced spatial-temporal feature extraction.

5.3. Experimental Results

5.3.1. Overall Comparison

Table 4 presents the average evaluation metrics for multi-step prediction tasks across four benchmark datasets, comparing the proposed FISTGCN model with multiple mainstream baselines. Overall, FISTGCN consistently outperforms other methods on most metrics and datasets, demonstrating its effectiveness.

Table 4. Performance comparison between FISTGCN and other models on four datasets. Bold: best; underline: second best.

Method	PeMS03			PeMS04			PeMS07			PeMS08		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
HA	30.08	46.22	28.64%	38.51	55.75	28.21%	45.32	65.74	21.56%	31.99	46.49	20.28%
VAR	23.65	38.26	24.51%	24.54	38.61	17.24%	50.22	75.63	32.22%	19.19	29.81	13.10%
SVR	21.97	35.29	21.51%	28.70	44.56	19.20%	32.49	50.22	14.26%	23.25	36.16	14.64%
DCRNN	15.53	27.18	15.62%	19.63	31.26	13.59%	21.16	34.14	9.02%	15.22	24.17	10.21%
STGCN	15.65	27.31	15.39%	19.57	31.38	13.44%	21.74	35.27	9.24%	16.08	25.39	10.60%
ASTGCN	17.34	29.56	17.21%	22.93	35.22	16.56%	24.01	37.87	10.73%	18.25	28.06	11.64%
GraphWaveNet	14.80	25.88	14.92%	18.54	30.09	12.71%	19.84	32.86	8.44%	14.54	23.67	9.41%
MTGNN	14.88	25.24	15.47%	18.96	31.05	13.65%	20.98	34.40	9.31%	15.12	24.23	9.65%
AGCRN	15.29	26.95	15.15%	19.83	32.26	12.97%	20.57	34.40	8.74%	15.95	25.22	10.09%
GMAN	16.52	27.18	17.36%	18.84	30.75	13.25%	20.97	34.20	9.05%	14.57	24.71	9.98%
STSGCN	17.48	29.21	16.78%	21.19	33.65	13.90%	24.26	39.03	10.21%	17.13	26.80	10.96%
STFGNN	16.77	28.34	16.30%	19.83	31.88	13.02%	22.07	35.80	9.21%	16.64	26.22	10.60%
ASTGNN	14.78	<u>25.00</u>	<u>14.79%</u>	18.60	30.91	12.36%	20.62	34.00	8.86%	15.00	24.70	9.50%
DGCRN	14.80	25.94	15.04%	18.80	30.65	12.82%	20.48	33.25	9.06%	14.60	24.16	9.33%
DSTAGNN	15.57	27.21	14.68%	19.30	31.46	12.70%	21.42	34.51	9.01%	15.67	24.77	9.94%
D ² STGNN	14.88	26.01	15.12%	18.34	29.93	12.81%	19.68	33.19	8.43%	14.35	24.18	9.33%
MegaCRN	14.84	26.25	15.16%	18.70	30.52	12.76%	19.89	33.12	8.47%	14.68	23.68	9.53%
PDFormer	14.76	25.56	15.51%	18.32	<u>29.96</u>	12.10%	19.83	<u>32.87</u>	8.52%	<u>13.58</u>	23.51	9.04%
PDG2Seq	<u>14.62</u>	25.47	14.88%	<u>18.24</u>	30.08	<u>12.09%</u>	19.28	33.04	8.07%	13.60	<u>23.37</u>	<u>8.99%</u>
FISTGCN	14.43	24.94	14.76%	18.06	29.92	12.05%	<u>19.46</u>	32.72	<u>8.24%</u>	13.23	22.76	8.75%

Traditional time series forecasting methods (e.g., HA, VAR) generally underperform compared to spatial-temporal neural networks, primarily due to their inability to model spatial dependencies. This underscores the necessity of spatial feature modeling. GNN models that capture dynamic spatial correlations (e.g., PDG2Seq, MegaCRN) usually outperform those based on static spatial assumptions (e.g., DCRNN, AGCRN). This is because dynamic graph modeling can adaptively adjust inter-node relationships in response to changing traffic flow. As a result, it better accommodates changes in network structure and traffic patterns, capturing more complex spatial dependencies. Notably, FISTGCN achieves superior performance across all four public datasets, largely due to its gated mechanism at the embedding stage. This mechanism effectively integrates multi-dimensional spatial-temporal information and enhances the representation of spatial-temporal features and periodicity. In its architecture, FISTGCN balances long-term network stability and short-term dynamics by fusing adaptive and dynamic adjacency matrices, allowing for more flexible spatial modeling. The spatial interaction mechanism further facilitates the extraction of deeper dynamic spatial correlations. Additionally, for temporal modeling, FISTGCN moves beyond single-scale approaches by employing an adaptive threshold to separate low- and high-frequency components. It also leverages learnable filtering and gated fusion to further enhance its ability to model multi-scale temporal features.

To comprehensively evaluate the models' performance across different prediction horizons, the future 12-step predictions of the PDG2Seq, PDFormer, DSTAGNN, and FISTGCN models are compared, with the corresponding MAE, RMSE, and MAPE reported and plotted,

as shown in Figure 3. As observed, FISTGCN achieves the best results across all prediction lengths and evaluation metrics. As the prediction horizon increases, the MAE, RMSE, and MAPE for all models increase, reflecting the greater difficulty of long-term forecasting. While FISTGCN performs comparably to the state-of-the-art PDG2Seq in short-term predictions, it demonstrates a clear advantage in long-term forecasting scenarios.

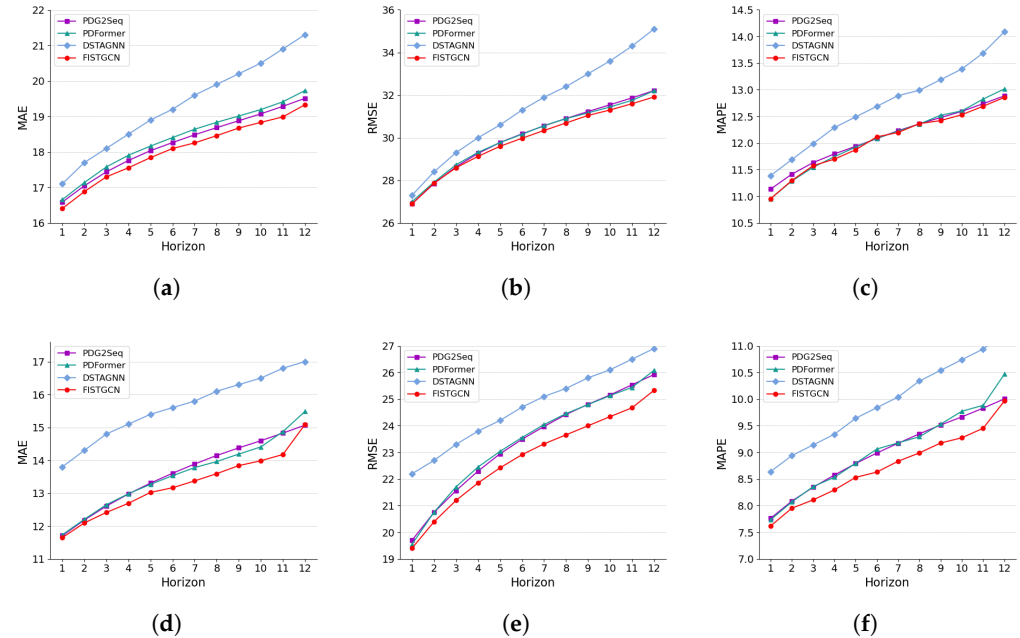


Figure 3. Comparison of Single-Step Prediction on the PeMS04 and PeMS08 Datasets. (a) MAE on PeMS04; (b) RMSE on PeMS04; (c) MAPE on PeMS04; (d) MAE on PeMS08; (e) RMSE on PeMS08; (f) MAPE on PeMS08.

5.3.2. Ablation Study

To further validate the effectiveness of each component in FISTGCN, we conduct systematic ablation experiments on the PeMS04 and PeMS08 datasets. Specifically, five variants of FISTGCN are designed by removing or replacing key modules:

w/o GE: The gating mechanism is removed from the embedding stage.

w/o DG: The dynamic graph convolution network (DGCN) is replaced by an adaptive GCN, where the adjacency matrix is set to the adaptive adjacency matrix.

w/o IL: The spatial interaction mechanism is replaced by directly adding the outputs of two SDGCN layers.

w/o HF: The high-frequency information is removed from the Gated Spectral Block.

w/o LF: The low-frequency information is removed from the Gated Spectral Block.

Table 5 provides a detailed comparison of FISTGCN and its ablated variants. Specifically, removing the gating mechanism in the embedding stage (*w/o* GE) results in higher errors. The gating mechanism enables the model to adaptively balance and integrate different information channels, which is crucial for filtering noise and enhancing spatial-temporal representations. Replacing the dynamic graph convolution with a static adaptive GCN (*w/o* DG) further reduces accuracy. This underscores the importance of modeling dynamic spatial relationships. Dynamic graph convolution allows the model to adapt flexibly to changing traffic network conditions. Similarly, eliminating the spatial interaction mechanism by simply stacking multi-scale spatial features (*w/o* IL), without explicit cross-scale interaction, weakens the model's performance. This demonstrates that cross-scale feature interaction is essential for capturing the multi-scale spatial dependencies inherent in real-world traffic networks. Finally, removing either the high-frequency (*w/o* HF) or low-frequency (*w/o* LF) components from the

Gated Spectral Block increases prediction errors. This emphasizes the necessity of modeling both short-term fluctuations and long-term trends. Each temporal component provides complementary information. The joint modeling of these components enables the network to capture both short-term disturbances and persistent temporal patterns.

Table 5. Performance comparison of different methods on PeMS04 and PeMS08.

Methods	PeMS04			PeMS08		
	MAE	RMSE	MAPE (%)	MAE	RMSE	MAPE (%)
<i>w/o</i> GE	18.31	30.29	12.16	13.31	23.04	8.87
<i>w/o</i> DG	18.31	30.60	12.15	13.43	23.05	8.85
<i>w/o</i> IL	18.15	30.25	12.10	13.39	23.06	8.81
<i>w/o</i> HF	18.15	30.18	12.18	13.32	22.94	8.81
<i>w/o</i> LF	18.40	30.87	12.34	13.49	23.17	8.89
FISTGCN	18.06	29.92	12.05	13.23	22.75	8.75

The ablation study clearly demonstrates that each component of FISTGCN plays a pivotal role in capturing different aspects of traffic flow. The gating mechanism enables the effective integration of multi-dimensional features, the dynamic graph convolution models time-varying spatial relationships, and the spatial interaction mechanism facilitates multi-scale spatial feature extraction. The frequency-domain separation (low and high-frequency components) allows the model to adaptively focus on different temporal components, improving its predictive accuracy in diverse traffic scenarios.

5.4. Computation Cost

Table 6 provides a detailed comparison of the computational costs between FISTGCN and two other competitive baseline models. To ensure fairness, all models were evaluated using a consistent batch size of 16. GPU costs include only the training phase and exclude unrelated processes such as data loading.

Table 6. Comparison of model performance and GPU memory usage.

Dataset	Model	Params (Total)	Training Time (s/epoch)	Inference (s)	GPU Memory (GB)
PeMS04	PDFormer	531,165	118.2	13.2	5.24
	PDG2Seq	1,153,053	63.8	6.3	2.59
	FISTGCN	486,909	52.8	7.5	3.95
PeMS08	PDFormer	531,165	50.5	5.4	2.07
	PDG2Seq	1,151,957	54.1	5.4	1.82
	FISTGCN	473,973	35.1	3.9	1.94

Experimental results demonstrate that FISTGCN is the most lightweight in terms of parameter count, having the smallest number of parameters among all compared models. In terms of training efficiency, FISTGCN achieved the shortest training time on both the PeMS04 and PeMS08 datasets, significantly outperforming the baselines. For inference efficiency, FISTGCN achieved the fastest inference speed on the PeMS08 dataset and ranked second, just after PDG2Seq, on PeMS04. Regarding GPU memory usage, FISTGCN's usage on PeMS08 was comparable to that of PDG2Seq. FISTGCN achieves superior predictive performance without a significant increase in computational cost, highlighting its advantage in cost-effectiveness and practical application.

5.5. Analysis of Component Effects of Gated Spectral Block

Figure 4 presents experiments on the PeMS04 and PeMS08 datasets to demonstrate the interpretability of the learned low- and high-frequency components in the GSB. By

comparing the predictions of the full FISTGCN model, its variants retaining only high or low-frequency components, and the actual traffic flow values, we reveal the distinct contributions of each spectral component.

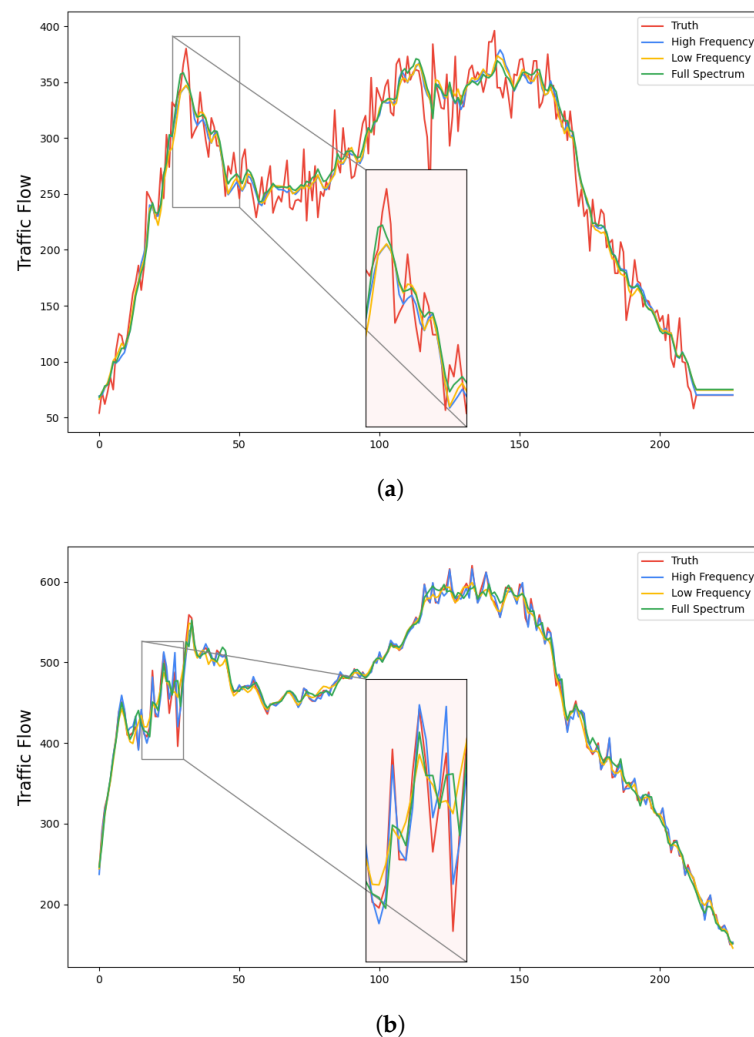


Figure 4. Visualization of visual comparison of prediction results for different inputs. (a) Visualization of real and predicted traffic flow at node 10 on PeMS04; (b) visualization of real and predicted traffic flow at node 49 on PeMS08.

From the visualizations, the full-spectrum predictions (green line) consistently align most closely with the ground truth (red line), highlighting the effectiveness of the GSB in fusing both low- and high-frequency information. High-frequency-only predictions (blue line) are sensitive to abrupt changes and short-term fluctuations but are volatile and tend to overfit noise, especially during stable periods. In contrast, low-frequency-only predictions (yellow line) produce smoother outputs and capture long-term trends, but they fail to reflect rapid local changes and often lag behind sudden events. By adaptively integrating both components, the full-spectrum model achieves balanced predictions, accurately capturing both rapid short-term variations and overall trends.

6. Conclusions

In conclusion, we introduce FISTGCN, a frequency-aware and interactive spatial-temporal graph convolutional network for traffic flow prediction. FISTGCN leverages dynamic adjacency matrices and a multi-scale spatial interaction mechanism to enhance spatial feature extraction, while a gated spectral block enables the adaptive separation and

fusion of low- and high-frequency components to model temporal features. Comprehensive experiments on four real-world datasets demonstrate that FISTGCN delivers state-of-the-art predictive performance while maintaining competitive computational efficiency.

However, several important directions for future research remain. First, scalability remains a critical challenge. While FISTGCN performs well on standard benchmark datasets, its scalability to larger datasets and more complex road networks needs further exploration. This includes optimizing the model for higher-dimensional and multi-modal data, such as integrating real-time traffic signal information or weather data. Second, while FISTGCN outperforms several baselines in terms of prediction accuracy, it does so at the cost of increased model complexity and training time. The trade-off between accuracy and computational efficiency should be quantified in future studies, considering both the model's training time and inference speed. Exploring more efficient architectures, such as pruning or distillation, could lead to improvements in these aspects. Finally, the practical implications of this research are significant for intelligent transportation systems (ITS). By accurately forecasting traffic conditions, FISTGCN can support real-time decision-making in urban traffic management, such as dynamic traffic signal control, congestion mitigation, and route optimization.

In future work, we plan to extend FISTGCN's framework to handle larger-scale traffic networks and multi-modal data, improve its scalability and computational efficiency, and investigate the integration of FISTGCN with real-time control systems in urban environments.

Author Contributions: Methodology, G.T.; Investigation, J.C.; Resources, M.Z.; Writing—original draft, G.T. and H.W.; Visualization, H.W.; Project administration, M.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Research Foundation of Chongqing University of Science and Technology under Grant ckrc2019032.

Informed Consent Statement: Not applicable.

Data Availability Statement: For this research, the datasets employed are publicly accessible real-world datasets. All code and scripts to reproduce the experiments are available at: <https://github.com/userTGQ/FISTGCN> (accessed on 5 May 2025).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Xiao, Z.; Fu, X.; Zhang, L.; Goh, R.S.M. Traffic Pattern Mining and Forecasting Technologies in Maritime Traffic Service Networks: A Comprehensive Survey. *IEEE Trans. Intell. Transp. Syst.* **2020**, *21*, 1796–1825. [CrossRef]
2. Xie, P.; Li, T.; Liu, J.; Du, S.; Yang, X.; Zhang, J. Urban flow prediction from spatiotemporal data using machine learning: A survey. *Inf. Fusion* **2020**, *59*, 1–12. [CrossRef]
3. Williams, B.M.; Hoel, L.A. Modeling and Forecasting Vehicular Traffic Flow as a Seasonal ARIMA Process: Theoretical Basis and Empirical Results. *J. Transp. Eng.* **2003**, *129*, 664–672. [CrossRef]
4. Defferrard, M.; Bresson, X.; Vandergheynst, P. Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering. In *Proceedings of the Advances in Neural Information Processing Systems 29 (NIPS 2016), Barcelona Spain, 5–10 December 2016*; Lee, D., Sugiyama, M., Luxburg, U., Guyon, I., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2016; Volume 29.
5. Xu, D.W.; Wang, Y.D.; Jia, L.M.; Qin, Y.; Dong, H.H. Real-time road traffic state prediction based on ARIMA and Kalman filter. *Front. Inf. Technol. Electron. Eng.* **2017**, *18*, 287–302. [CrossRef]
6. Jozefowicz, R.; Zaremba, W.; Sutskever, I. An empirical exploration of recurrent network architectures. In *Proceedings of the 32nd International Conference on Machine Learning—Volume 37, ICML'15, Lille, France, 7–9 July 2015*; pp. 2342–2350.
7. Hochreiter, S. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *Int. J. Uncertain. Fuzziness Knowl. Based Syst.* **1998**, *6*, 107–116. [CrossRef]
8. Chung, J.; Gulcehre, C.; Cho, K.; Bengio, Y. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. *arXiv* **2014**, arXiv:1412.3555. [CrossRef]

9. Lv, Y.; Duan, Y.; Kang, W.; Li, Z.; Wang, F.Y. Traffic Flow Prediction With Big Data: A Deep Learning Approach. *Trans. Intell. Transport. Syst.* **2015**, *16*, 865–873. [\[CrossRef\]](#)
10. Bai, L.; Yao, L.; Kanhere, S.S.; Yang, Z.; Chu, J.; Wang, X. Passenger Demand Forecasting with Multi-Task Convolutional Recurrent Neural Networks. In Proceedings of the Advances in Knowledge Discovery and Data Mining: 23rd Pacific-Asia Conference, PAKDD 2019, Macau, China, 14–17 April 2019; Part II; Springer: Berlin/Heidelberg, Germany, 2019; pp. 29–42. [\[CrossRef\]](#)
11. Han, L.; Du, B.; Sun, L.; Fu, Y.; Lv, Y.; Xiong, H. Dynamic and Multi-faceted Spatio-temporal Deep Learning for Traffic Speed Forecasting. In Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, KDD'21, Virtually, 14–18 August 2021; ACM: New York, NY, USA, 2021; pp. 547–555. [\[CrossRef\]](#)
12. Zhao, L.; Song, Y.; Zhang, C.; Liu, Y.; Wang, P.; Lin, T.; Deng, M.; Li, H. T-GCN: A Temporal Graph Convolutional Network for Traffic Prediction. *IEEE Trans. Intell. Transp. Syst.* **2020**, *21*, 3848–3858. [\[CrossRef\]](#)
13. Yu, B.; Yin, H.; Zhu, Z. Spatio-temporal Graph Convolutional Neural Network: A Deep Learning Framework for Traffic Forecasting. *arXiv* **2017**, arXiv:1709.04875.
14. Li, Y.; Yu, R.; Shahabi, C.; Liu, Y. Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting. *arXiv* **2018**, arXiv:1707.01926. [\[CrossRef\]](#)
15. Wu, Z.; Pan, S.; Long, G.; Jiang, J.; Zhang, C. Graph WaveNet for Deep Spatial-Temporal Graph Modeling. *arXiv* **2019**, arXiv:1906.00121.
16. Bai, L.; Yao, L.; Li, C.; Wang, X.; Wang, C. Adaptive Graph Convolutional Recurrent Network for Traffic Forecasting. In Proceedings of the Advances in Neural Information Processing Systems 33 (NeurIPS 2020), Virtual, 6–12 December 2020; Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2020; Volume 33, pp. 17804–17815.
17. Li, F.; Feng, J.; Yan, H.; Jin, G.; Yang, F.; Sun, F.; Jin, D.; Li, Y. Dynamic Graph Convolutional Recurrent Network for Traffic Prediction: Benchmark and Solution. *ACM Trans. Knowl. Discov. Data* **2023**, *17*, 9:1–9:21. [\[CrossRef\]](#)
18. Lin, J.; Li, Z.; Li, Z.; Bai, L.; Zhao, R.; Zhang, C. Dynamic Causal Graph Convolutional Network for Traffic Prediction. In Proceedings of the 2023 IEEE 19th International Conference on Automation Science and Engineering (CASE), Auckland, New Zealand, 26–30 August 2023; pp. 1–8. [\[CrossRef\]](#)
19. Qin, Y.; Tao, X.; Fang, Y.; Luo, H.; Zhao, F.; Wang, C. DMGSTCN: Dynamic Multigraph Spatio-Temporal Convolution Network for Traffic Forecasting. *IEEE Internet Things J.* **2024**, *11*, 22208–22219. [\[CrossRef\]](#)
20. Ju, W.; Zhao, Y.; Qin, Y.; Yi, S.; Yuan, J.; Xiao, Z.; Luo, X.; Yan, X.; Zhang, M. COOL: A Conjoint Perspective on Spatio-Temporal Graph Neural Network for Traffic Forecasting. *Inf. Fusion* **2024**, *107*, 102341. [\[CrossRef\]](#)
21. Li, Y.; Shao, Z.; Xu, Y.; Qiu, Q.; Cao, Z.; Wang, F. Dynamic Frequency Domain Graph Convolutional Network for Traffic Forecasting. In Proceedings of the ICASSP 2024—2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Seoul, Republic of Korea, 14–19 April 2024; pp. 5245–5249. [\[CrossRef\]](#)
22. Guo, C.; Chen, C.H.; Hwang, F.J.; Chang, C.C.; Chang, C.C. Multi-view spatiotemporal learning for traffic forecasting. *Inf. Sci.* **2024**, *657*, 119868. [\[CrossRef\]](#)
23. Tunnicliffe Wilson, G. Time Series Analysis: Forecasting and Control, 5th Edition, by George E. P. Box, Gwilym M. Jenkins, Gregory C. Reinsel and Greta M. Ljung, 2015. Published by John Wiley and Sons Inc., Hoboken, New Jersey, pp. 712. ISBN: 978-1-118-67502-1. *J. Time Ser. Anal.* **2016**, *37*, 709–711. [\[CrossRef\]](#)
24. Lu, Z.; Zhou, C.; Wu, J.; Jiang, H.; Cui, S. Integrating Granger Causality and Vector Auto-Regression for Traffic Prediction of Large-Scale WLANs. *KSII Trans. Internet Inf. Syst.* **2016**, *10*, 136–151. [\[CrossRef\]](#)
25. Dhiman, H.S.; Deb, D.; Guerrero, J.M. Hybrid machine intelligent SVR variants for wind forecasting and ramp events. *Renew. Sustain. Energy Rev.* **2019**, *108*, 369–379. [\[CrossRef\]](#)
26. May, M.; Hecker, D.; Körner, C.; Scheider, S.; Schulz, D. A Vector-Geometry Based Spatial kNN-Algorithm for Traffic Frequency Predictions. In Proceedings of the 2008 IEEE International Conference on Data Mining Workshops, Pisa, Italy, 15–19 December 2008; pp. 442–447. [\[CrossRef\]](#)
27. Song, C.; Lin, Y.; Guo, S.; Wan, H. Spatial-Temporal Synchronous Graph Convolutional Networks: A New Framework for Spatial-Temporal Network Data Forecasting. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; AAAI Press: Palo Alto, CA, USA, 2020; Volume 34, pp. 914–921. [\[CrossRef\]](#)
28. Zhang, W.; Zhu, K.; Zhang, S.; Chen, Q.; Xu, J. Dynamic graph convolutional networks based on spatiotemporal data embedding for traffic flow forecasting. *Knowl. Based Syst.* **2022**, *250*, 109028. [\[CrossRef\]](#)
29. Kong, J.; Fan, X.; Zuo, M.; Deveci, M.; Jin, X.; Zhong, K. ADCT-Net: Adaptive traffic forecasting neural network via dual-graphic cross-fused transformer. *Inf. Fusion* **2024**, *103*, 102122. [\[CrossRef\]](#)
30. Sun, Y.; Jiang, X.; Hu, Y.; Duan, F.; Guo, K.; Wang, B.; Gao, J.; Yin, B. Dual Dynamic Spatial-Temporal Graph Convolution Network for Traffic Prediction. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 23680–23693. [\[CrossRef\]](#)
31. Xu, Y.; Han, L.; Zhu, T.; Sun, L.; Du, B.; Lv, W. Generic Dynamic Graph Convolutional Network for traffic flow forecasting. *Inf. Fusion* **2023**, *100*, 101946. [\[CrossRef\]](#)

32. Lin, H.; Bai, R.; Jia, W.; Yang, X.; You, Y. Preserving Dynamic Attention for Long-Term Spatial-Temporal Prediction. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Virtual, 6–10 July 2020; pp. 36–46. [\[CrossRef\]](#)
33. Chen, C.; Liu, Y.; Chen, L.; Zhang, C. Bidirectional Spatial-Temporal Adaptive Transformer for Urban Traffic Flow Forecasting. *IEEE Trans. Neural Netw. Learn. Syst.* **2023**, *34*, 6913–6925. [\[CrossRef\]](#)
34. Zheng, C.; Fan, X.; Wang, C.; Qi, J. GMAN: A Graph Multi-Attention Network for Traffic Prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020*; AAAI Press: Palo Alto, CA, USA, 2020; Volume 34, pp. 1234–1241. [\[CrossRef\]](#)
35. Jiang, J.; Han, C.; Zhao, W.X.; Wang, J. PDFormer: Propagation Delay-Aware Dynamic Long-Range Transformer for Traffic Flow Prediction. In Proceedings of the AAAI Conference on Artificial Intelligence, Washington, DC, USA, 7–14 February 2023; AAAI Press: Palo Alto, CA, USA, 2023; Volume 37, pp. 4365–4373. [\[CrossRef\]](#)
36. Yan, X.; Gan, X.; Tang, J.; Zhang, D.; Wang, R. ProSTformer: Progressive Space-Time Self-Attention Model for Short-Term Traffic Flow Forecasting. *IEEE Trans. Intell. Transp. Syst.* **2024**, *25*, 10802–10816. [\[CrossRef\]](#)
37. Guo, S.; Lin, Y.; Feng, N.; Song, C.; Wan, H. Attention Based Spatial-Temporal Graph Convolutional Networks for Traffic Flow Forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019*; AAAI Press: Palo Alto, CA, USA, 2019; Volume 33, pp. 922–929. [\[CrossRef\]](#)
38. Wu, Z.; Pan, S.; Long, G.; Jiang, J.; Chang, X.; Zhang, C. Connecting the Dots: Multivariate Time Series Forecasting with Graph Neural Networks. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD’20, Virtual, 6–10 July 2020; pp. 753–763. [\[CrossRef\]](#)
39. Li, M.; Zhu, Z. Spatial-Temporal Fusion Graph Neural Networks for Traffic Flow Forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence, Virtually, 19–21 May 2021*; AAAI Press: Palo Alto, CA, USA, 2021; Volume 35, pp. 4189–4196. [\[CrossRef\]](#)
40. Lan, S.; Ma, Y.; Huang, W.; Wang, W.; Yang, H.; Li, P. DSTAGNN: Dynamic Spatial-Temporal Aware Graph Neural Network for Traffic Flow Forecasting. In Proceedings of the 39th International Conference on Machine Learning, PMLR, Baltimore, MA, USA, 17–23 July 2022; Proceedings of Machine Learning Research (PMLR); Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., Sabato, S., Eds.; PMLR: Baltimore, MD, USA, 2022; Volume 162, pp. 11906–11917.
41. Shao, Z.; Zhang, Z.; Wei, W.; Wang, F.; Xu, Y.; Cao, X.; Jensen, C.S. Decoupled Dynamic Spatial-Temporal Graph Neural Network for Traffic Forecasting. *arXiv* **2022**, arXiv:2206.09112. [\[CrossRef\]](#)
42. Jiang, R.; Wang, Z.; Yong, J.; Jeph, P.; Chen, Q.; Kobayashi, Y.; Song, X.; Fukushima, S.; Suzumura, T. Spatio-Temporal Meta-Graph Learning for Traffic Forecasting. In Proceedings of the AAAI Conference on Artificial Intelligence, Washington, DC, USA, 7–14 February 2023; AAAI Press: Palo Alto, CA, USA, 2023; Volume 37, pp. 8078–8086. [\[CrossRef\]](#)
43. Fan, J.; Weng, W.; Chen, Q.; Wu, H.; Wu, J. PDG2Seq: Periodic Dynamic Graph to Sequence Model for Traffic Flow Prediction. *Neural Netw.* **2025**, *183*, 106941. [\[CrossRef\]](#)
44. Guo, S.; Lin, Y.; Wan, H.; Li, X.; Cong, G. Learning Dynamics and Heterogeneity of Spatial-Temporal Graph Data for Traffic Forecasting. *IEEE Trans. Knowl. Data Eng.* **2022**, *34*, 5415–5428. [\[CrossRef\]](#)

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.