



ARTICLE

# Spatial-Temporal Graph Fusion with Dual-Scale Convolution for Traffic Flow Prediction

Dan Wang<sup>1</sup>, Mengyi Cui<sup>1</sup>, Zhenhua Yu<sup>1,\*</sup> and Yukang Liu<sup>2</sup>

<sup>1</sup>College of Artificial Intelligence & Computer Science, Xi'an University of Science and Technology, Xi'an, China

<sup>2</sup>China Petroleum Engineering & Construction Corp. Beijing Company, Beijing, China

\*Corresponding Author: Zhenhua Yu. Email: zhenhuayu@xust.edu.cn

Received: 08 October 2025; Accepted: 23 December 2025; Published: 09 April 2026

**ABSTRACT:** Traffic flow prediction is of great importance in traffic planning, road resource management, and congestion mitigation. However, existing prediction have significant limitations in modeling multi-scale spatial-temporal features, particularly in capturing temporal periodicity and spatial dependency in dynamically evolving traffic networks. This paper proposes a novel framework of traffic flow prediction, referred to as Adaptive Graph Fusion Dual-scale Convolutional Network (AGFDCN), which integrates spatial-temporal dynamic graphs with dual-scale convolutional networks. Specifically, we introduce a Dual-Scale Temporal Network, which combines long- and short-term dilated causal convolutions with a temporal decay-aware attention mechanism to efficiently capture traffic patterns across multiple temporal scales. Furthermore, we design a Dynamic Adaptive Graph Module, which models complex spatial dependencies in traffic networks through an adaptive graph fusion mechanism and a dual-path attention-gated module. Finally, the temporal and spatial representations are integrated by employing a gated fusion mechanism, enhancing the overall prediction performance. Experimental results obtained based on three highway datasets (i.e., PEMS04, PEMS07 and PEMS08) verify that the proposed model outperforms several state-of-the-art baselines in various evaluation metrics. Compared to the spatial-temporal graph model AGCRN with best performance in the baseline models, the proposed model exhibits significant improvements across all datasets: it achieves reduces of MAE by 42.07% and RMSE by 35.43% on PEMS04; MAE by 28.35% and RMSE by 29.28% on PEMS07; and MAE by 30.52% and RMSE by 30.73% on PEMS08, respectively, validating its effectiveness in modeling complex spatial-temporal traffic data and its robustness in handling sudden traffic changes.

**KEYWORDS:** Dual-scale convolution; dual-path attention-gated module; adaptive graph fusion; spatial-temporal dynamic graph; traffic flow prediction

## 1 Introduction

The escalating pace of urbanization and sustained growth in transportation demand have made traffic congestion a critical global challenge. Intelligent Transportation Systems (ITS) have become integral to modern urban traffic management. These systems employ progressive technologies—such as large-scale data analytics, artificial intelligence (AI), and the Internet of Things (IoT)—to enable real-time traffic monitoring and management [1,2]. Among the fundamental roles played by ITS, fine-grained traffic flow forecasting has a critical impact on enhancing transportation efficiency, reducing operational costs, and mitigating environmental impacts.

Traffic flow prediction is fundamentally a spatial-temporal sequence forecasting challenge. It requires the concurrent modeling of both dynamic temporal variations and complex spatial dependencies within

transportation networks. Deep learning techniques have shown excellent success in various fields such as computer vision, anomaly detection and natural language processing. Specifically, architectural frameworks such as Convolutional Neural Networks (CNNs) [3], Gated Recurrent Units (GRUs) [4], and Long Short-Term Memory (LSTM) [5] networks have been extensively employed to analyze spatial-temporal correlations in traffic data. Nevertheless, conventional neural architectures face inherent limitations when processing traffic network data, primarily due to: Non-uniform node distributions violating Euclidean spatial assumptions; Irregular topological structures compromising translation invariance [6]. These structural constraints significantly impair the predictive capacity of standard deep learning approaches in transportation applications.

Researchers have developed hybrid spatial-temporal architectures by integrating Graph Convolutional Networks (GCNs) with temporal modeling frameworks such as CNNs and GRUs. These integrated approaches perform well in practice as they effectively capture the dynamic interplay between spatial and temporal dependencies, which is inherent in complex transportation networks. For example, Zhao et al. [7] present a hybrid framework that combines temporal modeling and graph convolution, successfully tackling the difficulties caused by irregular structures and non-uniform node distributions in traffic networks, and producing a meaningful improvement in terms of the accuracy of traffic flow predictions. Furthermore, Zhu et al. [8] adopted an approach integrating graph convolution with multi-scale fusion, which enhances the model's adaptability in dynamic traffic environments by capturing traffic flow characteristics across different temporal scales. In spite of these advancements, existing techniques continue to struggle with modeling multi-scale information, particularly in capturing the inherent complex temporal dependencies and evolving spatial relationships within traffic data.

Specifically, most existing methods rely on static adjacency matrices to represent traffic network topology. However, static matrices are inadequate for capturing time-varying spatial correlations, which are usually induced by traffic fluctuations or unexpected events. In terms of temporal modeling, existing frameworks usually focus on long-term global trends while overlooking fine-grained variations across multiple temporal scales, resulting in delayed responses to sudden changes in traffic conditions. Additionally, spatial dependencies are typically defined by physical proximity, failing to capture latent correlations between geographically distant yet functionally similar regions. The proper depiction of intricate transportation networks is impeded by these constraints.

In order to overcome the difficulties associated with multi-scale dynamic spatial-temporal modeling in forecasting traffic flow, this paper proposes a novel framework called Adaptive Graph Fusion Dual-scale Convolutional Network (AGFDCN), which integrates spatial-temporal dynamic graph construction with dual-scale convolution to effectively capture multi-scale and time-varying spatial-temporal dependencies in complex traffic systems and improve prediction accuracy. The key innovation of this research lies in proposing the Dual-stream Attention Gating (DSAG) spatial modeling module, which addresses the critical challenge of adaptively adjusting road network spatial topology structures—specifically by dynamically updating adjacency matrices based on real-time traffic states. This study is significant because it shows that it performs better at capturing both short-term and long-term variations in traffic flow patterns while being resilient in challenging traffic situations.

The novelty of this study lies in both its conceptual formulation and architectural design. At the conceptual level, we propose an explicit temporal-decay perspective to better characterize the influence of temporal distance—an aspect insufficiently addressed by conventional recurrent or attention-based models. At the architectural level, we develop AGFDCN, a comprehensive spatiotemporal learning framework that integrates dual-scale temporal modeling, a dual-stream spatial gating mechanism, and multi-source graph fusion with adaptive feature balancing. These innovations collectively enhance the model's ability to

represent complex and non-stationary traffic dynamics, resulting in superior predictive performance relative to existing methods.

The following is a summary of this work's primary contributions:

- (1) **A Novel Temporal Modeling Framework with Explicit Temporal Decay Awareness:** we propose a Dual-Scale Temporal Network (DSTN) combined with a Temporal Decay-Aware Attention (TDA) mechanism. DSTN captures multi-scale temporal dependencies through parallel dilated convolutions, while TDA introduces learnable decay factors to explicitly model temporal distance. This design strengthens the model's ability to represent periodic patterns and react to sudden traffic changes—capabilities often lacking in conventional attention-based temporal models.
- (2) **A Hierarchical Spatial Modeling Mechanism via Dual-Stream Attention Gating:** to move beyond the unified spatial modeling used in many dynamic graph approaches, we introduce a Dual-Stream Attention Gating (DSAG) module within the Dynamic Adaptive Graph Module (DAGM). DSAG separately handles local structural correlations through graph convolution and global functional dependencies through graph attention, enabling adaptive and interpretable fusion of spatial information. This hierarchical design improves the model's adaptability to irregular network topologies and time-varying traffic conditions.
- (3) **An Integrated Multi-Source Graph Fusion and Adaptive Spatiotemporal Gating Framework:** we develop a comprehensive graph fusion mechanism that unifies static topological, dynamic similarity, and functional correlation matrices, yielding a richer and more flexible spatial representation than single adaptive graph designs. A learnable gating fusion module further balances temporal and spatial features in real time, enhancing training stability and prediction robustness, particularly under non-stationary or rapidly changing traffic conditions.

This paper is structured in the following manner: [Section 2](#) systematically reviews the related works, including traditional prediction methods and the spatial-temporal graph neural networks. [Section 3](#) introduces the proposed AGFDCN model, with detailed explanations of the design principles behind its key components. [Section 4](#) evaluates the model's performance through comparative experiments, ablation studies, and visual analysis. [Section 5](#) outlines the findings and planned research for this topic.

## 2 Related Works

### 2.1 Traditional Methods in Traffic Flow Prediction

It is a great challenge to model complex spatial dependencies embedded in traffic networks and capture nonlinear temporal dependencies caused by constantly changing road conditions. In traditional methods, statistical learning and machine learning techniques are widely applied. For instance, Historical Average (HA) [9], Vector Auto-Regressive (VAR) [10], and Auto-Regressive Integrated Moving Average (ARIMA) [11] are commonly used based on linear dependency and data stationarity. However, real-world traffic often exhibits significant nonlinearity and instability, which limits the prediction accuracy in statistical models.

To address these challenges, machine learning algorithms such as Support Vector Regression (SVR) [12] and K-Nearest Neighbors (KNN) [13] are applied to handle high-dimensional feature spaces and capture local patterns of data, thus relaxing the assumptions of linearity and stationarity. Nonetheless, these models lack ability to simulate spatial structures and temporal dynamics simultaneously, rendering them ineffective in developing the spatial-temporal relationships, limiting their predictive accuracy.

Deep learning algorithms excel at capturing nonlinear characteristics and spatial-temporal correlations in traffic data. For instance, Recurrent Neural Networks (RNNs) [14], including Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU), are commonly used to extract temporal relationships from

traffic flow sequences. These models have shown promising prediction results. Given the complicated spatial connections in road networks—where the traffic situation of one road segment is frequently modified by its neighbors, Convolutional Neural Networks (CNNs) are utilized to improve spatial feature extraction.

Building upon these foundations, researchers have proposed various deep learning models that integrate temporal modeling with spatial structure perception. To simulate intricate spatial-temporal connections in traffic data, Luo et al. [15] offer GT-LSTM, which combines a self-adaptive graph convolutional network, temporal convolutional network, and bidirectional LSTM. Liu et al. [16] develop a grey convolutional neural network (G-CNN), which leverages grey clustering to extract traffic accident variables and integrates them with spatial-temporal traffic data via a third-order tensor, obtaining improved prediction accuracy in accident scenarios. For long-sequence traffic speed forecasting, Luo et al. [17] create STGIN, a spatial-temporal graph-informer network. It models complicated geographical topology and long-term temporal connections at the same time by combining a Transformer-based Informer module with a Graph Attention Network (GAT). To extract and fuse multi-dimensional features in parallel, Li et al. [18] suggest a multi-stream feature fusion (MFF) model that combines GCN, GRU, and a fully connected neural network (FNN) within a soft attention mechanism. Although deep learning Methods have contributed to higher accuracy in predictions, they generally suffer from insufficient dynamic adaptability and inadequate multi-scale feature extraction, particularly demonstrating limited responsiveness to sudden traffic incidents.

## **2.2 Prediction Based on Spatial-Temporal Graph Neural Networks**

Spatial-temporal graph neural network research has significantly increased in the field of traffic flow prediction. Conventional convolutional neural networks (CNNs) are quite good at extracting spatial characteristics from grid-based data in Euclidean space, but they have a lot of trouble processing features in non-Euclidean regions. Graph Convolutional Networks (GCNs) [19,20] have been widely used to anticipate traffic flow to address the problems. By merging GCNs with time-series deep learning techniques, numerous spatial-temporal feature fusion models have been created.

Several studies focus on attention mechanisms and adaptive graph structures—where graph connectivity is learned or adjusted from data rather than fixed—to enhance feature extraction. Shi et al. [21] propose the Graph-Talk Attention Network (GTHAT) and Bidirectional Enhanced Attention Gated Recurrent Unit (Bi-EAGRU) for multi-scale traffic trend fusion. Hu et al. [22] introduce a multi-head attention spatial-temporal graph neural network that leverages adaptive graph structures. Lei et al. [23] develop a multi-graph spatial-temporal convolution model (MSGCNN) that fuses multi-source data. Huang et al. [24] present a Gated spatial-temporal Graph Network (GGNN) combined with Temporal Convolution Networks (TCN) to capture long-term dependencies. Zhang et al. [25] design a multi-scale spatial-temporal Transformer-based model (MSS-STT) that integrates relative position information. Wang and Oktomy Noto Susanto [26] propose a CNN-LSTM model that captures heterogeneous spatiotemporal correlations and improves prediction performance by integrating multi-source external features and temporal dependencies. Zhao et al. [27] propose a Position-Aware Graph Convolutional Network (STPGCN) with trainable spatial-temporal embeddings. Li et al. [28] develop a multi-task model (MTL-GGA) that fuses single-task features via attention mechanisms. Zhang et al. [29] propose a Convolutional Neural Network-Bi-directional Long Short-Term Memory-Attention Mechanism (CNN-BiLSTM-Attention) model based on Kalman-filtered data processing, which enhances prediction stability and accuracy by extracting spatial-temporal features and assigning adaptive weights through an attention mechanism.

Other models emphasize hybrid architectures integrating convolutional, recurrent, and attention layers. Li et al. [30] propose STEGCN combining TransD embeddings with GCN and convolutional modules. Xi et al. [31] propose a Dynamic Graph and Multi-dimensional Attention Neural Network (DGMAN)

model, which captures time-varying spatial correlations and complex spatiotemporal dependencies through dynamic multi-graph structures and multi-dimensional attention mechanisms. Cao et al. [32] present a Spatial-Temporal Hypergraph Convolutional Network (STGHCN) to capture higher-order spatial relationships. Xia et al. [33] develop a Dynamic Spatial-Temporal Graph Recurrent Neural Network (DSTGRNN) integrating dynamic graph generation and fusion. Zhao et al. [34] propose an interpretable multi-scale spatiotemporal neural network (MSTNN), which constructs a multi-scale time segmentation mechanism and uses spatial attention KAN and temporal channel mixed KAN to solve the problem of insufficient multi-scale feature extraction and model interpretability in traffic prediction. Yu et al. [35] introduce the MSSTGCN model, which extracts spatiotemporal dependencies through a graph attention residual layer and a T-GCN module, and uses a position-embedded Transformer to model long-term temporal patterns. Ma et al. [36] propose a hybrid spatiotemporal model that integrates a Transformer-based self-supervised mask mechanism with a recurrent neural ODE module to jointly capture long-term heterogeneous dependencies and adaptively explore short-term spatiotemporal characteristics. Liu and Zhang [37] propose a spatiotemporal interactive dynamic graph convolutional network for traffic forecasting, named STIDGCN, which leverages an interactive spatial-temporal framework and a data-driven dynamic graph constructed from a traffic pattern library to capture heterogeneous dependencies. Han et al. [38] propose SAGCRN, which enhances input sequences using spatiotemporal context and a traffic pattern library, and adaptively learns graph structures for prediction.

While recent studies have improved adaptability through data-driven graph construction, several fundamental challenges remain in achieving robust and efficient traffic forecasting. Spatial-temporal graph neural networks have advanced traffic flow prediction by better modeling complex topological structures and dynamic dependencies. However, important limitations persist: models with static graph structures struggle to accommodate real-time traffic changes; sophisticated attention mechanisms and graph operations often introduce high computational overhead; and maintaining an effective balance between local details and global trends becomes difficult, especially during sudden traffic disruptions. Nevertheless, emerging techniques such as dynamic graph generation and multi-scale feature fusion provide promising directions for developing more adaptive and resilient traffic prediction systems.

### 3 Development of AGFDCN

#### 3.1 Problem Definition

Traffic flow prediction uses historical data gathered from various sensors in the context of the road network to anticipate future transportation scenarios, including flow and speed. The road network can be modeled as a weighted directed graph  $G = (V, E, A_{pre})$  for modeling the spatial relationships between sensors;  $V$  denotes a collection of nodes that represent  $N$  sensors;  $E$  denotes a collection of edges that represent the connectivity between sensors; and  $A_{pre} \in R^{N \times N}$  is the adjacency structure matrix that measures the proximity between nodes (e.g., distance, traffic flow correlation, etc.). At any time  $t$ , the traffic state can be formulated as a graph signal matrix  $X_t \in R^{N \times D}$ ,  $D$  is the measure of traffic properties (like flow, speed, occupancy rate, and more).

Given historical traffic flow information data  $X_{t-H+1:t} \in R^{N \times D \times H}$  over  $H$  time steps, the traffic prediction problem can be formulated as learning a mapping function  $\Phi(\cdot)$ , which combines the structure of the network topology  $G$  and historical data to forecast the traffic state  $X_{t+1:t+T} \in R^{N \times D \times T}$  over next  $T$  time steps. This procedure can be expressed as:

$$X_{t+1:t+T} = \Phi(X_{t-H+1:t}, G) \quad (1)$$

### 3.2 General Framework of AGFDCN

This study presents the AGFDCN traffic flow prediction model, which combines a double-scale convolution mechanism with spatial-temporal dynamic graphs.

As shown in Fig. 1, the model adopts a multi-layer cascading structure. First of all, the input data is initially preprocessed through channel convolution; Subsequently, the data is passed through three core modules for collaborative processing: 1) The time feature extraction module (DSTN), which models long and short-period temporal features using double-scale diffusion causal convolution (LS-TCN) and temporal decay-aware attention mechanism (TDA); 2) The spatial feature extraction module (DAGM) leverages adaptive graph fusion (DF-Matrix) to model spatial dependencies at multiple levels, while incorporating the dual-stream attention gating mechanism (DSAG) to dynamically modulate the contribution of spatial features; 3) The gated fusion module, which integrates spatial-temporal features dynamically using a learnable weight distribution mechanism. The model uses residual and skip connections to enhance gradient propagation, and the final prediction is output through a  $1 \times 1$  convolution. Table 1 summarizes the correspondence between architectural components, their functional roles, and the sections where they are detailed.

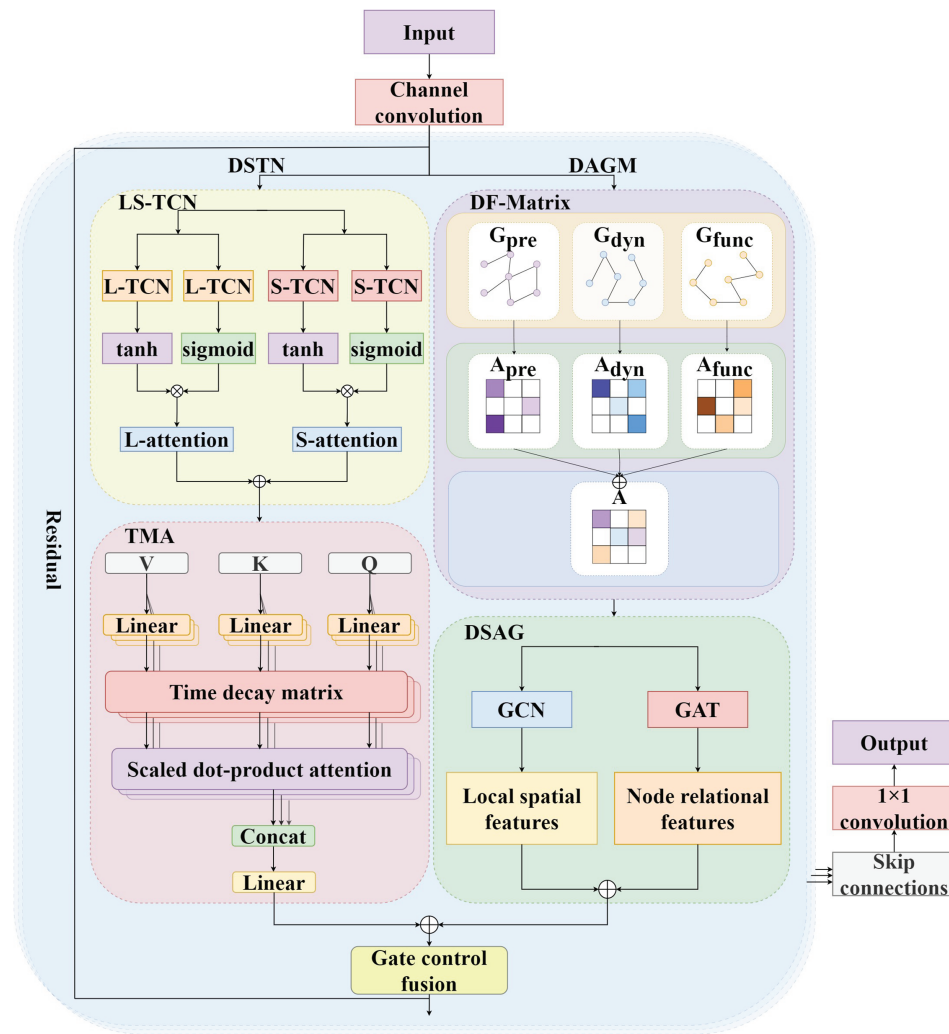


Figure 1: The structure of AGFDCN.

### 3.3 Dual-Scale Temporal Network (DSTN)

To effectively model multi-scale temporal dependencies and complex dynamic changes, this paper proposes a feature extraction framework that combines a double-scale temporal convolution network (TCN) with a temporal decay-aware attention mechanism. The framework captures both fine-grained and coarse-grained temporal features through a global-local collaborative structure and enhances the representation ability using a feed-forward network, thus boosting the model's ability to perceive temporal dynamics.

**Table 1:** Correspondence between AGFDCN components and their functional roles.

Component	Main Function	Section
DSTN	Dual-scale temporal network	<a href="#">Section 3.3</a>
LS-TCN	Long/short-term dilated convolution	<a href="#">Section 3.3.1</a>
TDA	Temporal decay-aware attention	<a href="#">Section 3.3.2</a>
DAGM	Dynamic adaptive graph module	<a href="#">Section 3.4</a>
Adaptive Graph Fusion	Multi-source graph integration	<a href="#">Section 3.4.1</a>
DSAG	Dual stream attention gating mechanism	<a href="#">Section 3.4.2</a>
Gated Fusion	Spatio-temporal feature integration	<a href="#">Section 3.5</a>

#### 3.3.1 Double-Scale TCN

Traditional RNN commonly suffer from vanishing gradient problems, while standard TCNs is difficult to effectively balance global and local features. This paper develops a double-scale TCN architecture, LS-TCN, that uses parallel global (L-TCN) and local (S-TCN) branches to model temporal dependencies across varying time scales, this approach leads to improved accuracy in capturing intricate spatial-temporal relationships within sequential data.

- (1) Global Temporal Feature Extraction (L-TCN): For the input feature  $X$ , two parallel TCNs are applied, with  $\tanh$  (controlling feature range) and sigmoid (for gating and selection) performing as the activation functions:

$$H_{\tanh}^L = \tanh(W_{\tanh}^L * X + b_{\tanh}^L) \quad (2)$$

$$H_{\text{sigm}}^L = \text{sigm}(W_{\text{sigm}}^L * X + b_{\text{sigm}}^L) \quad (3)$$

where  $W_{\tanh}^L$  and  $W_{\text{sigm}}^L$  are learnable convolutional kernel parameters;  $b_{\tanh}^L$  and  $b_{\text{sigm}}^L$  are corresponding bias parameters, respectively.

After feature fusion, the global temporal attention mechanism enhances the feature representation of important time steps:

$$H^L = H_{\tanh}^L \odot H_{\text{sigm}}^L \quad (4)$$

The global temporal attention weights  $A^L$  is calculated as follows:

$$A^L = \text{softmax}\left(\frac{Q^L(K^L)^T}{\sqrt{d_k}}\right)V^L \quad (5)$$

where  $Q^L$ ,  $K^L$ , and  $V^L$  are the Query, Key, and Value matrices, respectively, which are obtained through linear transformations from  $H^L$ ;  $\sqrt{d_k}$  is the scaling factor, which is used to control the magnitude of attention scores.

- (2) Local Temporal Feature Extraction (S-TCN): The structure is as the same as L-TCN, but the convolutional kernel is smaller, focusing on local details:

$$H^S = H_{\tanh}^S \odot H_{\text{sigm}}^S \quad (6)$$

Local temporal attention weights  $A^S$  are calculated as follows:

$$A^S = \text{softmax} \left( \frac{Q^S (K^S)^T}{\sqrt{d_k}} \right) V^S \quad (7)$$

Dual-scale features are integrated through a learnable weight  $\alpha$ :

$$Z = \alpha A^L + (1 - \alpha) A^S \quad (8)$$

where  $A^L$  is the global temporal attention weight,  $A^S$  is the local temporal attention weight, and  $Z$  is the output dual-scale feature.

### 3.3.2 Temporal Decay-Aware Attention (TDA)

The multi-head attention mechanism primarily emphasizes the similarity between features while overlooking the influence of relative positions along the temporal dimension. In this work, we introduce a Time Decay-aware Attention (TDA) mechanism. By incorporating a learnable temporal decay factor, TDA explicitly models the relative positional relationships across different time steps, thereby enhancing the model's ability to capture temporal dependencies. Moreover, it can adaptively adjust the importance of different temporal stages. Compared with existing approaches—such as GRU-based AGCRN or the standard Transformer architecture—TDA can more effectively learn the temporal dynamics in traffic flow data, making it particularly suitable for long-range dependency modeling scenarios where sensitivity to temporal distance is crucial.

The input to this module is the weighted integrated features  $Z \in R^{B \times T \times d}$  output from the previous level, batch size is denoted by  $B$ , time step count by  $T$ , and feature dimension by  $d$ . The core innovation of temporal attention lies in the introduction of learnable temporal decay factors. For each attention head, the query-key interaction matrix not only considers feature similarity but also incorporates the decay weights of temporal intervals, with the specific formula being:

$$A_i = \text{softmax} \left( \frac{Q_i K_i^T}{\sqrt{d_k}} + \lambda D \right) \quad (9)$$

where  $D \in R^{T \times T}$  is the temporal decay matrix, and the element  $D_{t,t'}$  is defined as:

$$D_{t,t'} = \exp(-\gamma |t - t'|) \quad (10)$$

where  $\gamma$  is the learnable decay coefficient, controlling the strength of temporal decay;  $\lambda$  is the super parameter for decay weighting, and it is used to adjust the impact of temporal decay factors in attention calculations, making the feature relationships between adjacent time steps stronger, while interactions at distant time are steps appropriately suppressed, which is consistent with the characteristics of most sequential data.

Concatenating each head's results and applying linear transformations yields the multi-head attention's final output:

$$TMA(Z) = \text{Concat}(\text{head}_1, \dots, \text{head}_h) W^O \quad (11)$$

The output of each attention head is:  $\text{head}_i = A_i V_i$ ,  $A_i$  is the attention weight of the  $i$ -th attention head;  $V_i$  is the corresponding value vector;  $W^o$  represents a learnable output weight matrix, which is used to map the concatenated multi-head output to the required output dimension.

The final temporal feature representation is:

$$Z' = TMA(Z) \quad (12)$$

To further process temporal features and enhance the model's stability and learning capabilities, a residual regression front-end network is employed. First, the temporal feature  $Z'$  is normalized to stabilize the training process and accelerate convergence:

$$Z_{\text{norm}} = \text{LayerNorm}(Z') \quad (13)$$

Next, the front-end network layer further extracts features; this layer typically includes one or more fully connected layers to capture deeper feature relationships:

$$Z_{\text{ff}} = \text{ReLU}(W_{\text{ff}}Z_{\text{norm}} + b_{\text{ff}}) \quad (14)$$

Then, by residually connecting the original features with the output of the front-end network, this helps preserve the original information and promote information flow:

$$\text{Out}_S = Z + Z_{\text{ff}} \quad (15)$$

where  $\text{Out}_S$  represents the temporal feature output after residual connection and normalization processing, which can be considered as the final temporal feature output.

### 3.4 Dynamic Adaptive Graph Module (DAGM)

To capture the time-varying, non-uniform spatial dependencies of traffic flow, this paper proposes a dynamic spatial modeling module. This module builds dynamic adjacency matrices and functional similarity adjacency matrices, combining Graph Attention Networks (GAT) and Dynamic Graph Convolution Networks (DGCN) to form a dual-stream structure for extracting multi-level spatial features. The module effectively combines local and global spatial information via a gating mechanism, boosting the model's capacity to respond to variations in complex traffic networks.

#### 3.4.1 Adaptive Graph Fusion

This module integrates the static topological matrix, dynamic adaptive matrix, and functional correlation matrix with learnable parameters to construct a unified and adaptive graph structure representation. This fusion mechanism greatly enhances the model's ability to capture the changing relationships between road segments over time, allowing for dynamic adaptation to structural shifts in the traffic network and ensuring accurate predictions during abrupt traffic fluctuations.

- (1) **Dynamic Adaptive Matrix:** To capture the temporal correlation of node features, a dynamic adaptive matrix generation module is designed. Given the input feature tensor  $Q \in R^{B \times N \times T \times C}$  ( $B$ ,  $N$ ,  $T$ ,  $C$  represent batch size, count of nodes, time steps, and feature dimension, respectively). First, we calculate the mean feature values of nodes along the temporal dimension:

$$\bar{F} = \frac{1}{BT} \sum \sum Q_{b,:,t,:} \quad (16)$$

where  $1/BT$  is a normalization factor, ensuring that the mean feature value is not affected by batch size  $B$  and time step  $T$ , and equal consideration of each node's features at all time steps.

Second, the dynamic adaptive matrix  $A_{dyn} \in R^{N \times N}$  is calculated using the cosine similarity, utilizing the SoftMax function to enhance the sparsity of the adjacency matrix as follows:

$$A_{dyn} = \text{softmax} \left( \frac{\bar{F} \cdot \bar{F}^T}{\|\bar{F}\| \|\bar{F}^T\|} \right) \quad (17)$$

where  $\bar{F} \cdot \bar{F}^T$  calculates the inner product similarity between nodes.  $\|F\|$  denotes the norm of  $F$  (typically the L2 norm). The SoftMax function is utilized along the row, which helps to highlight connections with higher similarity between nodes while suppressing those with lower similarity, thus forming a clearer and more meaningful topological structure in the network.

- (2) **Functional Correlation Matrix:** To measure the functional relevance of various nodes, a functional correlation matrix is built using feature correlation coefficients. First, we calculate the mean feature values of different nodes  $\bar{F} \in R^{N \times C}$ , serving as the central position of each node in the feature space. Next, the correlation coefficient between nodes is calculated with the use of covariance matrix as follows:

$$R_{i,j} = \frac{Cov(\bar{F}_i \bar{F}_j)}{\sigma_{\bar{F}_i} \sigma_{\bar{F}_j}} \quad (18)$$

where  $Cov(\cdot)$  represents covariance, and  $\sigma$  is the standard deviation. This formula represents the Pearson correlation coefficient, which quantifies the linear relationship between two node feature vectors. This method can quantify the functional similarity between nodes, providing a basis for constructing the functional correlation matrix.

Next, we construct the functional correlation matrix  $A_{func}$  by applying the SoftMax function to increase its sparsity, which helps to highlight connections with higher similarity between nodes while suppressing those with lower similarity. It is calculated as follows:

$$A_{func} = \text{softmax}(R) \quad (19)$$

- (3) **Adaptive Fusion:** To create a comprehensive adjacency matrix  $A$ , learnable parameters are integrated with the static topological matrix  $A_{pre}$ , dynamic adaptive matrix  $A_{dyn}$ , and functional correlation matrix  $A_{func}$ . Here is how this step is accomplished:

$$A = A_{pre} + \beta A_{dyn} + \gamma A_{func} \quad (20)$$

where  $\beta$  and  $\gamma$  are learnable scaling factors, enabling the model to automatically modify the various adjacent matrices' contributions throughout the training phase.

The aggregated adjacency matrix is preprocessed using a symmetric normalization technique to increase numerical stability. Specifically: the degree matrix  $D = \text{diag}(A1_N)^{-1/2}$  is first calculated, then we obtain the normalized adjacency matrix by  $\hat{A} = DAD$ . This normalization process mitigates issues like numerical instability or explosion during calculations, while maintaining the model's steadiness and accelerating the convergence process during training.

### 3.4.2 Dual Stream Attention Gating Mechanism (DSAG)

A dual stream attention gating mechanism (DSAG) is proposed to address the limitations of existing dynamic graph models (e.g., AGCRN), in which a single graph convolution operator is insufficient for comprehensively modeling spatial dependencies. DSAG adopts a dual-stream parallel architecture: the graph convolution branch is responsible for capturing local spatial correlations among nodes in the traffic network, while the graph attention branch focuses on modeling complex global dependencies and can adaptively adjust the importance weights of different nodes. This dual-stream design enables the model to flexibly adapt to dynamic changes in road network structures, preserving fine-grained local details while simultaneously accounting for global relationships. As a result, the model maintains stronger robustness under complex scenarios—such as sudden traffic incidents—and achieves more accurate extraction of multi-dimensional spatiotemporal features.

- (1) Graph Convolution Branch: Spatial features are extracted from the traffic network using graph convolution methods. By adaptively fusing to obtain the adjacency matrix  $\hat{A}$ , it comprehensively reflects the connection relationships between nodes. The following is the representation of the graph convolution operation:

$$X_G^{(t)} = \text{ReLU}(W_g \hat{A} X_{:, :, t, :}) \quad (21)$$

The graph convolution operation at the  $t$ -th time step generates the output feature  $X_G^{(t)} \in R^{B \times N \times C}$  through a ReLU-activated transformation of the product with the learnable weight matrix  $W_g$ , the normalized adjacency matrix  $\hat{A}$  encoding graph structure, and the input feature matrix  $X_{:, :, t, :}$ , and all time-step graph convolution features by  $H$ .

- (2) Graph Attention Branch: Node dependencies are captured by the graph attention branch using the graph attention technique. The graph attention can be represented as:

$$e_{ij} = \text{LeakyReLU}(a^T [WX_i || WX_j]) \quad (22)$$

$$\alpha_{ij} = \text{softmax}(e_{ij}) \quad (23)$$

$$H'_i = \sigma\left(\sum_{j \in N(i)} \alpha_{ij} WX_j\right) \quad (24)$$

where  $e_{ij}$  denotes the preliminary attention score between nodes  $i$  and  $j$ , computed through a LeakyReLU-activated transformation of their linearly projected features ( $WX_i$  and  $WX_j$ ) using learnable parameter vector  $a$ ; this score is subsequently normalized via SoftMax to obtain the attention weight  $\alpha_{ij}$ , which governs feature aggregation from neighboring nodes  $N_{(i)}$  to generate the refined representation  $H'_i$  through an element-wise nonlinear activation  $\sigma$ , with  $W$  serving as the shared weight matrix for feature transformation across all nodes.

- (3) Dual-Stream Attention Gating Mechanism: Finally, the dual-branch features are integrated through the attention gating mechanism:

$$g = \sigma(W_s H'_s + W_g X_G) \quad (25)$$

$$\text{Out}_T = g \odot H'_i + (1 - g) X_G \quad (26)$$

where  $H'_i$  is the graph attention branch's output,  $X_G$  is the graph convolution's output,  $g$  is the gating signal, which is produced by linearly combining the graph attention and graph convolution outputs using the sigmoid function, and  $\text{Out}_T$  is the model's overall output in terms of node dependency relationship modeling and spatial feature extraction.

### 3.5 Gating Fusion

In the designed model, spatial features  $\text{Out}_S$  and temporal features  $\text{Out}_T$  are extracted separately through graph convolution and temporal attention multi-head mechanisms. To further improve the model's capacity to represent spatial-temporal data, these two feature categories are weighted and integrated using a gating fusion process. The integrated features are then passed through a  $1 \times 1$  convolution for the final prediction.

The gating mechanism adaptively modifies the fusion weights according on each feature's significance. The specific calculating process is as follows:

$$g = \sigma(W_T \cdot \text{Out}_T + W_S \cdot \text{Out}_S) \quad (27)$$

where  $g$  is the gating signal obtained through the sigmoid activation function, controlling the fusion ratio of temporal and spatial features;  $W_S$  and  $W_T$  are learnable parameter matrices that correspond to the weights of temporal and spatial features, respectively.

The final integrated output is calculated as follows:

$$\text{Out}_{\text{fused}} = g\text{Out}_T + (1 - g)\text{Out}_S \quad (28)$$

## 4 Experimental Studies

### 4.1 Datasets

Three highway traffic datasets—PEMS04, PEMS07, and PEMS08—provided by the California Department of Transportation's PEMS system are used in the analysis. The datasets are collected from highway networks in the San Francisco Bay Area, the Los Angeles/Ventura County area, and the San Bernardino Area. Specifically, PEMS04 covers 01 January to 28 February 2018; PEMS07 spans 01 May to 30 June 2017; and PEMS08 includes 01 July to 31 August 2016. All datasets record traffic flow and sensor geographic information over a two-month period. For this investigation, the raw data is aggregated into 5-min intervals (288 points per day per sensor) after being collected at 30-s intervals. Each data point contains three features: flow, average speed, and occupancy rate.

In the data preprocessing stage, all feature data is first normalized using Z-score standardization to eliminate dimensional differences between different features. The datasets are then divided in a 6:2:2 ratio into training, validation, and test sets. To prevent data leaking, the time sequence is closely adhered to during the split. Specific statistical data for the datasets are given in [Table 2](#).

**Table 2:** Datasets specific statistical information.

Datasets	Nodes	Edges	Time Steps
PEMS04	307	340	16,992
PEMS07	883	866	28,224
PEMS08	170	295	17,856

### 4.2 Experimental Setups

A Linux server (CPU: 6 vCPUs Intel<sup>®</sup> Xeon<sup>®</sup> Gold 6430, GPU: RTX 4090 (24 G)) is used for all investigations. To forecast the traffic situation for the upcoming 12 time-steps (60 min), a sliding window technique is used, utilizing historical data from the previous 12 time-steps (60 min) as input. The model is trained with the aid of the Adam optimizer with a starting learning rate of 0.0001. There are 250 training

iterations, and the training batch size is 16. Table 3 displays the configuration of the remaining fundamental model parameters.

**Table 3:** Core model parameters.

Parameter	Value
Embedding Dimension	64
Number of Attention Heads	4
Global Time Convolution Kernel Size	(12, 1)
Local Time Convolution Kernel Size	(4, 1)
Prediction Layer Convolution Kernel Size	(1, 1)
Forward Expansion Coefficient	4

### 4.3 Evaluation Metrics

This study assesses the model's performance using three assessment metrics to ensure the objectivity and accuracy of the experimental data: Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Mean Absolute Percentage Error (MAPE). The model performs better at making predictions when the evaluation measures have a smaller value.

$$MAE = \frac{1}{n} \sum_{i=1}^n |y - y_i| \quad (29)$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y - y_i)^2} \quad (30)$$

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{y - y_i}{y} \right| \quad (31)$$

where  $y$  is the true value,  $y_i$  is the predicted value.

### 4.4 Baseline Models

As indicated in Table 4, a number of baseline models are chosen for comparative analytics in order to evaluate the performance of the suggested model.

### 4.5 Experiment Results

#### 4.5.1 Comparative Experiments and Visualization

To validate the performance of the proposed model, all applied models are experimented with the same parameter settings as the proposed model, and the performance is evaluated under the same testing environment for 12 time-steps. The experiments are quantitatively assessed using three evaluation metrics, namely (MAE, RMSE, and MAPE).

Across all datasets and evaluation metrics in Table 5, the AGFDCN model consistently outperforms the baseline methods, with the best-performing results highlighted in bold for clarity. On PEMS04, it reduces MAE by over 42% and RMSE by more than 35% compared with traditional models such as ARIMA, highlighting the advantages of deep learning in capturing nonlinear traffic dynamics. On the larger and more complex PEMS07 dataset, AGFDCN achieves strong results (MAE = 21.42, RMSE = 34.04), surpassing dynamic graph models like AGCRN and demonstrating its adaptability to high-dimensional network structures.

**Table 4:** Description of the baseline models.

Models	Description
ARIMA [11]	A time series analysis technique called auto-regressive integrated moving average is used to forecast future values from historical data.
LSTM [5]	Recurrent neural networks (RNNs) of the Long Short-Term Memory kind use cell state control to solve the problem of long-term dependencies in time series data.
GRU [4]	A condensed form of RNN with a gating mechanism, the Gated Recurrent Unit uses update and reset gates to extract time series features.
STSGCN [39]	Spatial-Temporal Synchronous Graph Convolutional Network builds local spatial-temporal graphs and designs synchronous graph convolution modules. It captures spatial-temporal heterogeneity through a multi-module structure and performs dynamic spatial-temporal representation via parallel processing.
AGCRN [40]	The Adaptive Graph Convolutional Recurrent Networks provide end-to-end traffic flow prediction without a predetermined graph structure by establishing node-adaptive parameter learning and data-adaptive graph creation modules.
ASTGCN [41]	Using a spatial-temporal attention mechanism with graph convolution, the Spatial-Temporal Graph Convolutional Network with Attention Mechanism models current, daily, and weekly traffic flow dependencies separately while dynamically capturing spatial-temporal correlations.
STGCN [42]	Spatial-temporal Graph Convolutional Network models the spatial-temporal dependencies of traffic networks through spatial-temporal graph convolution blocks, using a fully convolutional structure to achieve efficient parallel training.

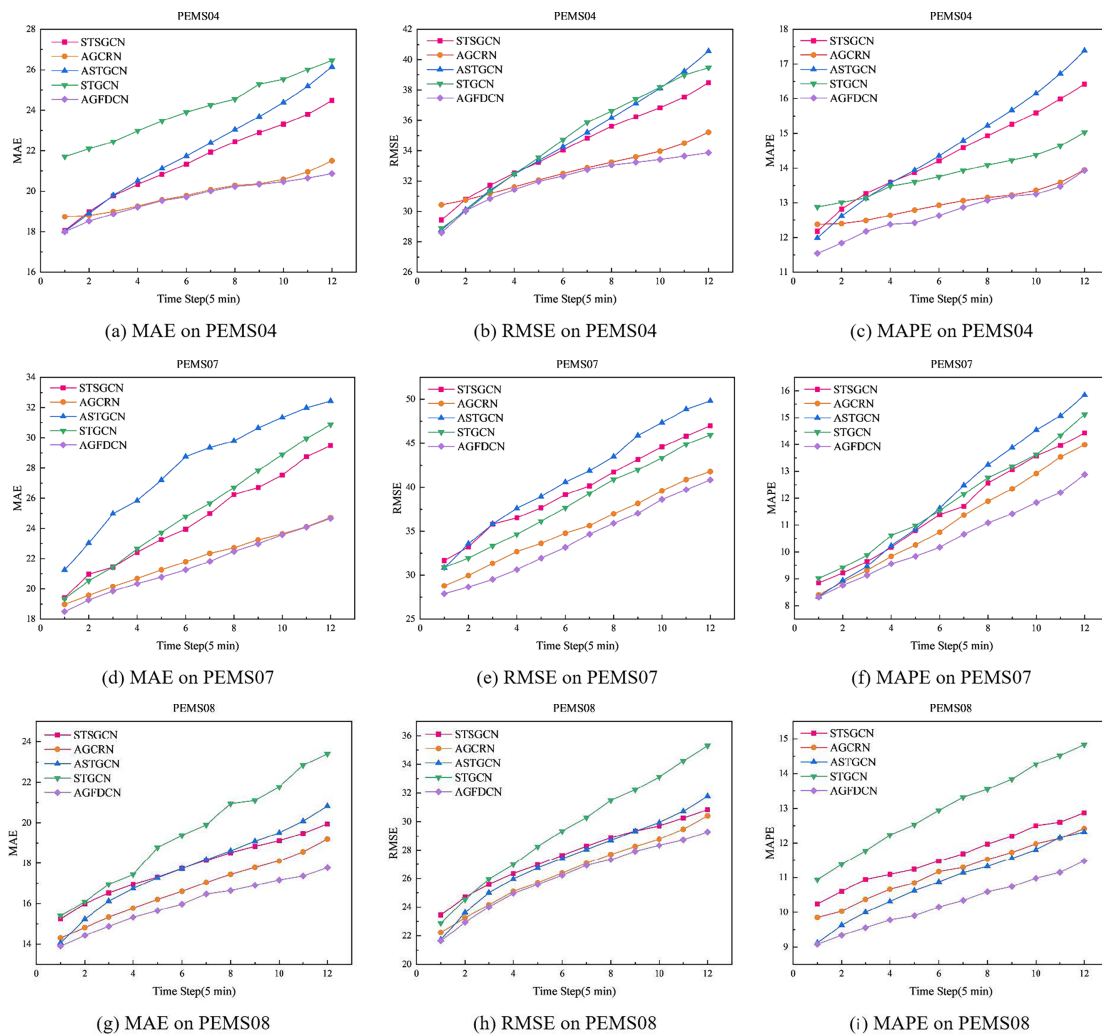
**Table 5:** Comparative results of different models on datasets PEMS04, PEMS07 and PEMS08.

Model	PEMS04			PEMS07			PEMS08		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
ARIMA	33.73	48.80	24.18	38.17	59.27	19.46	31.09	44.32	22.73
LSTM	25.77	40.16	17.34	27.99	43.53	12.59	28.81	40.80	29.61
GRU	25.81	40.18	17.40	28.24	43.78	12.77	20.54	32.15	13.92
STSGCN	21.52	34.03	14.39	24.60	39.75	11.61	17.82	27.67	11.62
AGCRN	19.90	32.71	12.99	21.74	35.65	11.12	16.76	26.64	11.16
ASTGCN	22.81	35.58	16.02	28.05	42.48	12.02	19.15	28.96	10.90
STGCN	24.05	37.65	19.48	25.30	38.58	11.88	19.41	30.06	14.64
<b>AGFDCN</b>	<b>19.54</b>	<b>31.51</b>	<b>12.73</b>	<b>21.42</b>	<b>34.04</b>	<b>10.48</b>	<b>16.21</b>	<b>26.35</b>	<b>10.25</b>

Although LSTM- and GRU-based models learn temporal dependencies, they neglect spatial correlations and thus yield higher errors. Graph-based models such as STGCN and STSGCN incorporate spatial information but are limited by fixed adjacency matrices, reducing their responsiveness to traffic incidents or peak-hour variations. Both AGCRN and AGFDCN employ dynamic graph structures; however, AGFDCN further enhances feature extraction through the DF-Matrix, DSAG, and TDA modules. While the absolute MAE improvement over AGCRN is modest (e.g., 0.36 on PEMS04), the relative gains of 1.8%–3.3% are

consistent across all datasets. In traffic forecasting—where errors are highly correlated and models often plateau near performance ceilings—such cross-dataset, steady improvements reflect meaningful architectural advances. Moreover, AGFDCN achieves these gains with a more interpretable and modular design, including explicit multi-scale temporal modeling (DSTN + TDA) and hierarchical spatial fusion (DSAG), providing clearer mechanisms for capturing complex traffic dynamics than the more monolithic adaptive graph strategy of AGCRN. On PEMS08, AGFDCN attains a low MAPE of 10.25%, further confirming its effectiveness in smaller and structurally regular networks.

The results in Fig. 2 show AGFDCN as the best performer across all metrics, proving the effectiveness of its multi-module design and strong stability. The trend indicates improved performance with increasing structural complexity, highlighting the positive impact of graph structures and attention mechanisms. Compared to fixed-graph models like STGCN and ASTGCN, AGFDCN’s dynamic graph modeling is more effective in capturing time-varying traffic patterns. It also shows the least fluctuation in error curves, confirming its superior long-term forecasting capability in complex scenarios.

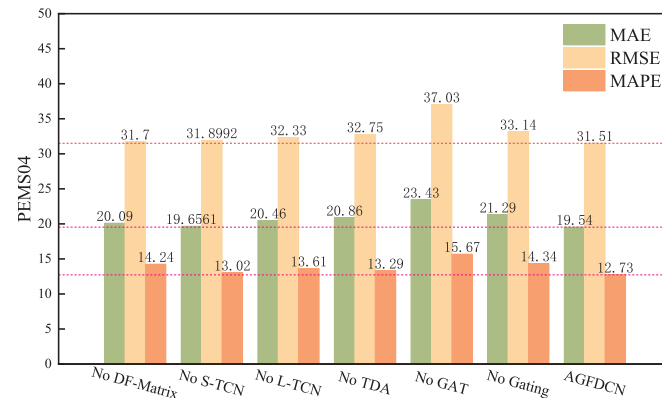


**Figure 2:** Performance comparison of AFGDCN with each benchmark model on the PEMS04, PEMS07 and PEMS08 datasets.

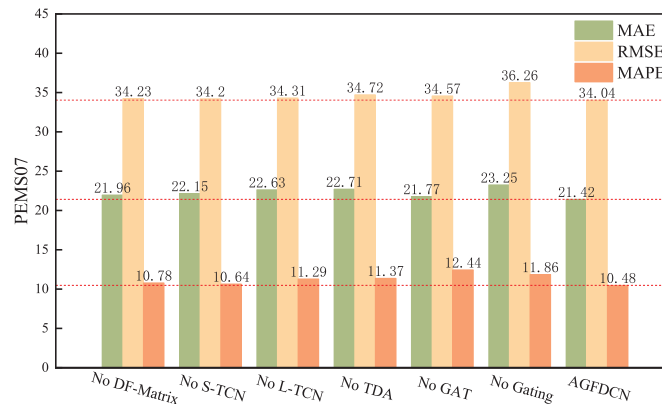
#### 4.5.2 Ablation Study and Visualization

A number of systematic ablation experiments are carried out in order to assess each core module's contribution to the AGFDCN model's performance. To ensure reproducibility, each module removal is implemented as follows: removing the dynamic adaptive matrix and functional similarity matrix is achieved by using only the static adjacency matrix  $A_{pre}$  in Eq. (20); removing the dual-scale temporal convolution is performed by disabling either the global (L-TCN) or local (S-TCN) branch via setting  $\alpha = 1$  or 0 in Eq. (8), respectively; removing TDA is implemented by replacing Eq. (9) with standard scaled dot-product attention; removing the graph attention mechanism in DSAG is done by retaining only the graph convolution branch in Eq. (25) with  $g = 1$ ; and removing the gating fusion module is achieved by replacing Eqs. (27) and (28) with direct feature concatenation and linear projection. By progressively removing these key components, the impact of each on overall model performance is quantitatively analyzed.

Consistently outperforms all variants. Removing the dynamic graph fusion module increases RMSE by 0.13 on PEMS04, confirming its importance in modeling dynamic spatial relationships. Eliminating GAT results in a 3.89 MAE increase on PEMS04, highlighting its role in capturing global node dependencies. Without the gated fusion module, MAPE degrades by 2.67% on PEMS08, emphasizing its necessity for spatiotemporal integration. The dual-scale temporal branches offer complementary benefits: S-TCN captures short-term fluctuations (0.92 MAE increase when removed), while L-TCN models long-term trends (0.82 RMSE increase when removed). As visualized in Fig. 3, the radar chart visually demonstrates that all ablated variants are enclosed within the full model's curve, illustrating the synergy and necessity of each component across diverse traffic scenarios.

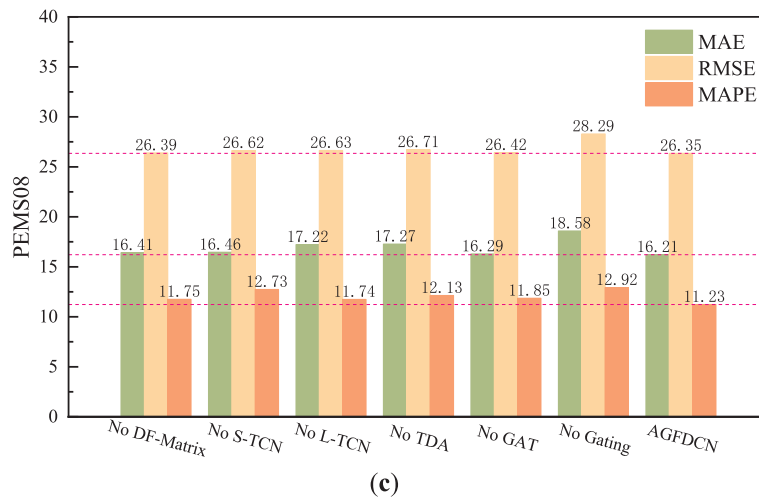


(a)



(b)

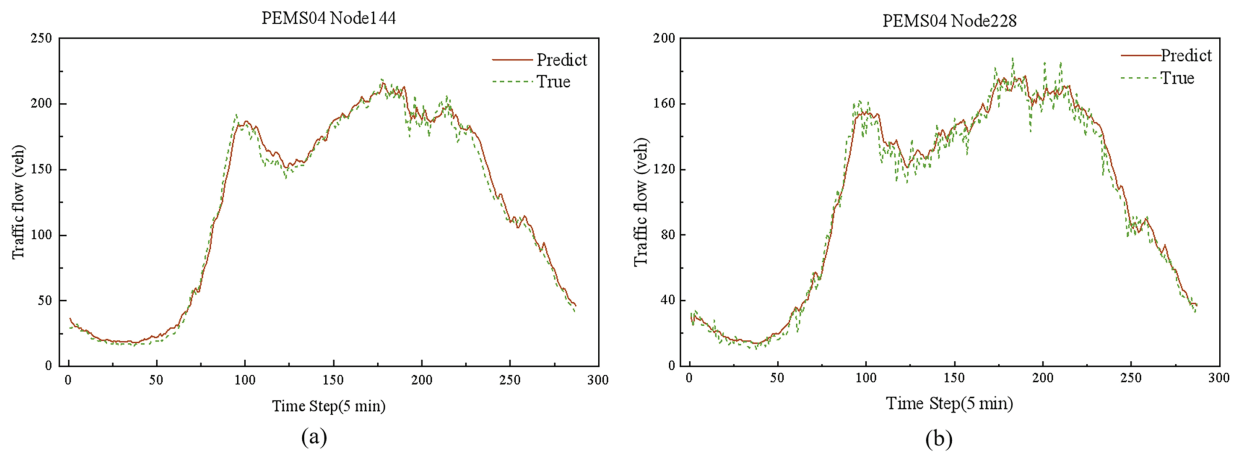
Figure 3: (Continued)



**Figure 3:** Visualization of Ablation Results (a) PEMS04 (b) PEMS07 (c) PEMS08.

#### 4.5.3 Visualization of Prediction Results

To visually demonstrate the predictive performance of the AGFDCN model, this study selects traffic flow data from Sensor 144 and Sensor 228 in the PEMS04 dataset for prediction and visualization (Fig. 4a,b). These two sensors are located in areas with high and stable traffic flow, respectively, providing rich traffic data. The solid line in the pictures reflects the ground truth, while the dashed line shows the model’s projections for the traffic flow at these two nodes throughout the course of the following 12 time-steps (60 min).



**Figure 4:** Visualization of Prediction Results on the PEMS04 Datasets (a) Sensor 144 (b) Sensor 228.

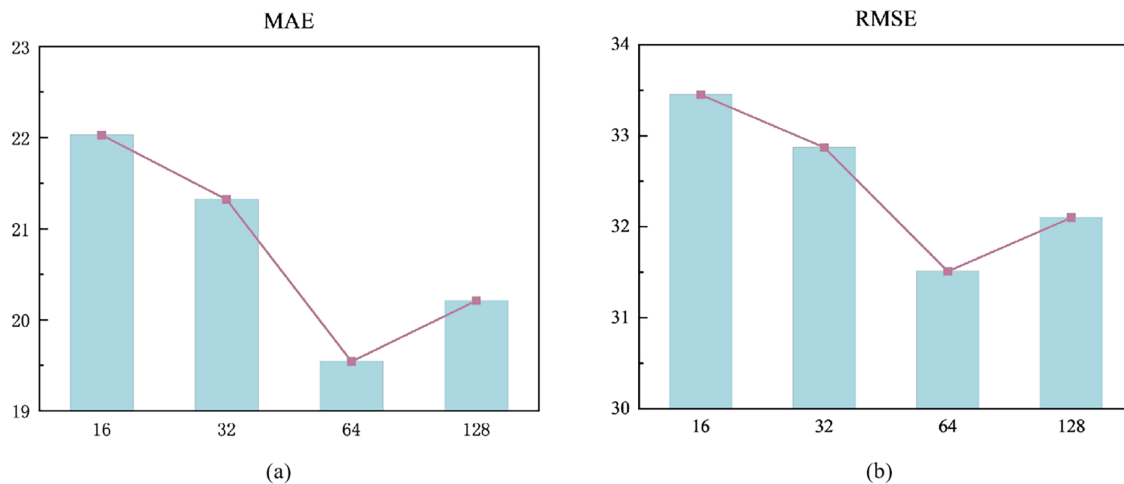
Overall, the predicted values closely match the measured data, indicating that the model effectively captures the dynamic behavior of traffic flow. Even during large fluctuations, the predictions generally follow the ground truth with only minor delays, demonstrating strong robustness and adaptability under complex conditions. However, noticeable deviations remain at certain peak points, suggesting that prediction accuracy under extreme scenarios still requires improvement.

To quantify prediction stability, we compute the standard deviation over five independent runs. The average relative standard deviation stays below 3% across all time steps, showing that the model produces consistent results despite inherent training randomness. During sudden congestion events, however, the

model exhibits delayed responses and reduced accuracy compared with stable flow periods. This implies that, despite the combination of dynamic graph fusion and decay-aware attention, the model's sensitivity to abrupt nonlinear changes is still constrained by factors such as temporal receptive fields and graph update frequency. Future work may therefore focus on real-time graph updates or anomaly-aware attention modules to enhance responsiveness to sudden traffic abnormalities.

#### 4.5.4 Hyperparameter Analysis

With an emphasis on the influence of embedding dimension on prediction performance in the PEMS04 dataset, Fig. 5 examines how hyperparameters affect the AGFDCN model. As the embedding dimension rises, the model's prediction accuracy first improves before declining, as shown in the figure. Lower dimensions lead to poor feature representation, making it difficult to capture intricate traffic flow patterns. However, beyond a certain threshold (e.g., dimension  $> 64$ ), further increasing the dimension degrades test performance. To validate whether this degradation stems from overfitting, we compare training and test MAE across dimensions (see Fig. 5). When the dimension increases from 64 to 128, training MAE decreases from 18.32 to 17.45, while test MAE rises from 19.54 to 20.81. This widening gap (from 1.22 to 3.36) clearly indicates overfitting: the model memorizes training data but fails to generalize. Therefore, excessively high dimensions increase model complexity and lead to overfitting, compromising generalization to unseen data.



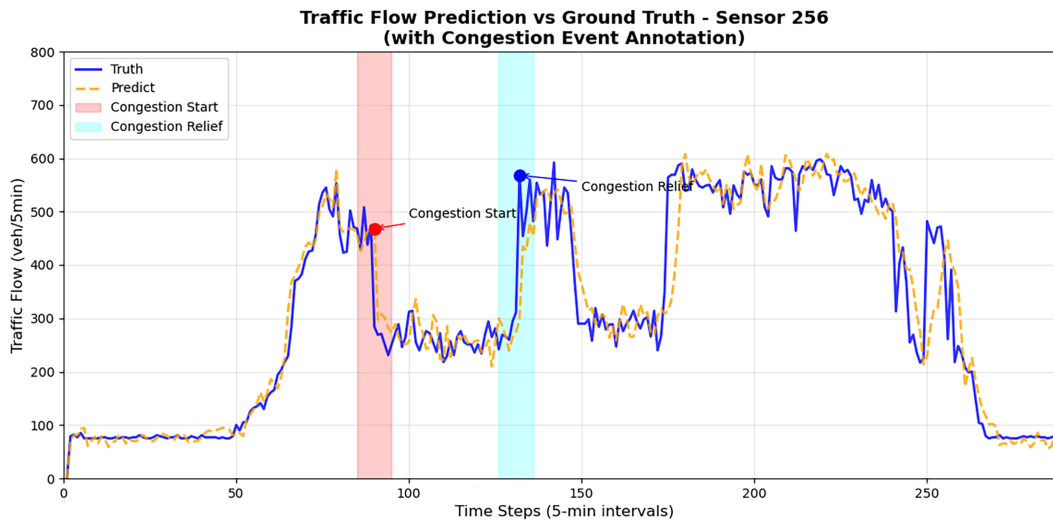
**Figure 5:** Hyperparameter analysis results on the PEMS04 dataset (a) MAE (b) RMSE.

#### 4.5.5 Case Study

To verify the effectiveness of the proposed AGFDCN model in traffic flow forecasting, we selected the actual observation data of sensor 256 within a specific time period and compared it with the model's prediction results, as illustrated in Fig. 6. The figure also marks the start and relief times of the congestion event to facilitate examination of the model's performance under congested conditions.

From the overall trend, the AGFDCN model is able to closely fit the variations of real traffic flow. In particular, during non-congestion periods, the predicted curve nearly coincides with the actual curve, demonstrating high prediction accuracy. Around the peak traffic period, the model also accurately captures the rising and falling dynamics of the flow, indicating strong capability in extracting temporal features. Notably, during the annotated congestion event, the actual traffic flow exhibits a sharp decline followed by persistent fluctuations at low-flow levels. Although the AGFDCN model slightly overestimates the flow in this stage, it still clearly reflects the overall pattern of the abrupt drop and subsequent recovery. This suggests

that the model possesses a certain degree of responsiveness to sudden congestion events, further validating its robustness and applicability under complex traffic conditions.



**Figure 6:** Prediction result on a pattern with abrupt change.

## 5 Conclusion

In this work, we propose AGFDCN, a traffic flow prediction model that integrates a dual-scale temporal framework with spatiotemporal dynamic graph modeling to effectively capture complex, multi-scale dependencies in evolving traffic networks. The model's dual-scale temporal module combines long-short range dilated causal convolutions with a temporal-aware multi-head attention mechanism to learn both short-term fluctuations and long-term trends. For spatial modeling, a dynamic fusion mechanism adaptively integrates physical, dynamic, and functional similarity-based adjacency matrices, while a dual-stream attention gating module unifies graph convolution and graph attention to capture both local and global spatial relationships. A gating fusion module further enhances performance by adaptively integrating temporal and spatial representations.

Extensive experiments on three benchmark datasets—PEMS04, PEMS07, and PEMS08—demonstrate that AGFDCN consistently outperforms classical and state-of-the-art baselines in MAE, RMSE, and MAPE, showcasing its strong predictive accuracy and generalization capability. Ablation studies further validate the necessity of each component and the synergistic effectiveness of the overall architecture.

Despite its strong performance, AGFDCN still faces limitations. Under extreme traffic conditions—such as sudden congestion or incident-induced breakdowns—prediction accuracy can degrade due to the latency of adaptive graph updates and the limited receptive field of temporal convolutions. Moreover, while the model shows robustness on freeway datasets, its generalizability to urban road networks, multimodal traffic systems, or regions with significantly different topologies requires further investigation. Future work will focus on improving real-time adaptability, incorporating external contextual factors such as weather and special events, and evaluating the model across a broader spectrum of spatiotemporal forecasting tasks.

**Acknowledgement:** The authors are deeply grateful to all team members involved in this research.

**Funding Statement:** This work was supported in part by the National Nature Science Foundation of China under Grants 62476216 and 62006184, in part by the Key Research and Development Program of Shaanxi Province under

Grant 2024GX-YBXM-146, in part by the Scientific Research Program Funded by Education Department of the Shaanxi Provincial Government under Grant 23JP091, and the Youth Innovation Team of Shaanxi Universities.

**Author Contributions:** Conceptualization: Dan Wang, Mengyi Cui, Zhenhua Yu, Yukang Liu; Methodology: Dan Wang, Mengyi Cui; Formal analysis and investigation: Dan Wang, Mengyi Cui, Zhenhua Yu; Writing—original draft preparation: Dan Wang, Mengyi Cui; Writing—review and editing: Dan Wang, Mengyi Cui; Funding acquisition: Dan Wang, Zhenhua Yu; Resources: Dan Wang, Mengyi Cui, Yukang Liu; Supervision: Dan Wang, Zhenhua Yu. All authors reviewed and approved the final version of the manuscript.

**Availability of Data and Materials:** The data that support the findings of this study are available from the corresponding author upon reasonable request.

**Ethics Approval:** Not applicable.

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

## References

1. Nagy AM, Simon V. Improving traffic prediction using congestion propagation patterns in smart cities. *Adv Eng Inform.* 2021;50(1):101343. doi:10.1016/j.aei.2021.101343.
2. Medina-Salgado B, Sánchez-DelaCruz E, Pozos-Parra P, Sierra JE. Urban traffic flow prediction techniques: a review. *Sustain Comput Inform Syst.* 2022;35(7):100739. doi:10.1016/j.suscom.2022.100739.
3. Zhang W, Yu Y, Qi Y, Shu F, Wang Y. Short-term traffic flow prediction based on spatio-temporal analysis and CNN deep learning. *Transp A Transp Sci.* 2019;15(2):1688–711. doi:10.1080/23249935.2019.1637966.
4. Shu W, Cai K, Xiong NN. A short-term traffic flow prediction model based on an improved gate recurrent unit neural network. *IEEE Trans Intell Transp Syst.* 2022;23(9):16654–65. doi:10.1109/TITS.2021.3094659.
5. Ma C, Dai G, Zhou J. Short-term traffic flow prediction for urban road sections based on time series analysis and LSTM\_BILSTM method. *IEEE Trans Intell Transp Syst.* 2022;23(6):5615–24. doi:10.1109/TITS.2021.3055258.
6. Kashyap AA, Raviraj S, Devarakonda A, Nayak KSR, Santhosh KV, Bhat SJ. Traffic flow prediction models—a review of deep learning techniques. *Cogent Eng.* 2022;9(1):2010510. doi:10.1080/23311916.2021.2010510.
7. Zhao L, Song Y, Zhang C, Liu Y, Wang P, Lin T, et al. T-GCN: a temporal graph convolutional network for traffic prediction. *IEEE Trans Intell Transp Syst.* 2020;21(9):3848–58. doi:10.1109/TITS.2019.2935152.
8. Zhu J, Wang Q, Tao C, Deng H, Zhao L, Li H. AST-GCN: attribute-augmented spatiotemporal graph convolutional network for traffic forecasting. *IEEE Access.* 2021;9:35973–83. doi:10.1109/ACCESS.2021.3062114.
9. Smith BL, Williams BM, Keith Oswald R. Comparison of parametric and nonparametric models for traffic flow forecasting. *Transp Res Part C Emerg Technol.* 2002;10(4):303–21. doi:10.1016/S0968-090X(02)00009-8.
10. Miao K, Phillips PCB, Su L. High-dimensional VARs with common factors. *J Econom.* 2023;233(1):155–83. doi:10.1016/j.jeconom.2022.02.002.
11. Williams BM, Hoel LA. Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: theoretical basis and empirical results. *J Transp Eng.* 2003;129(6):664–72. doi:10.1061/(asce)0733-947x(2003)129:.
12. Wu CH, Ho JM, Lee DT. Travel-time prediction with support vector regression. *IEEE Trans Intell Transp Syst.* 2004;5(4):276–81. doi:10.1109/tits.2004.837813.
13. Liu Y, Yu H, Fang H. Application of KNN prediction model in urban traffic flow prediction. In: 2021 5th Asian Conference on Artificial Intelligence Technology (ACAIT); 2021 Oct 29–31; Haikou, China. p. 389–92. doi:10.1109/acait53529.2021.9731348.
14. He Y, Huang P, Hong W, Luo Q, Li L, Tsui KL. In-depth insights into the application of recurrent neural networks (RNNs) in traffic prediction: a comprehensive review. *Algorithms.* 2024;17(9):398. doi:10.3390/a17090398.
15. Luo Y, Zheng J, Wang X, Tao Y, Jiang X. GT-LSTM: a spatio-temporal ensemble network for traffic flow prediction. *Neural Netw.* 2024;171(3):251–62. doi:10.1016/j.neunet.2023.12.016.
16. Liu Y, Wu C, Wen J, Xiao X, Chen Z. A grey convolutional neural network model for traffic flow prediction under traffic accidents. *Neurocomputing.* 2022;500:761–75. doi:10.1016/j.neucom.2022.05.072.

17. Luo R, Song Y, Lu Y, Zhao N, Su R. STGIN: a spatial temporal graph-informer network for long sequence traffic speed forecasting. In: Proceedings of the 2024 IEEE 27th International Conference on Intelligent Transportation Systems (ITSC); 2024 Sep 24–27; Edmonton, AB, Canada. p. 3221–6. doi:10.1109/ITSC58415.2024.10919917.
18. Li Z, Xiong G, Tian Y, Lv Y, Chen Y, Hui P, et al. A multi-stream feature fusion approach for traffic prediction. *IEEE Trans Intell Transp Syst.* 2022;23(2):1456–66. doi:10.1109/TITS.2020.3026836.
19. Rahmani S, Baghbani A, Bouguila N, Patterson Z. Graph neural networks for intelligent transportation systems: a survey. *IEEE Trans Intell Transp Syst.* 2023;24(8):8846–85. doi:10.1109/TITS.2023.3257759.
20. Liang G, Kintak U, Ning X, Tiwari P, Nowaczyk S, Kumar N. Semantics-aware dynamic graph convolutional network for traffic flow forecasting. *IEEE Trans Veh Technol.* 2023;72(6):7796–809. doi:10.1109/TVT.2023.3239054.
21. Shi X, Cao F, Ji Y, Ma J. Traffic flow prediction method based on multi-scale spatio-temporal features and soft attention mechanism. *Comput Eng.* 2024;50(12):346–57. doi:10.19678/j.issn.1000-3428.0068659.
22. Hu X, Yu E, Zhao X. Multi-head attention spatial-temporal graph neural networks for traffic forecasting. *J Comput Commun.* 2024;12(3):52–67. doi:10.4236/jcc.2024.123004.
23. Lei B, Li JL, Zhang P, Li W, Chen C. Long term prediction on urban traffic flow based on multi-source spatio-temporal graph convolutional neural network model. *J Highw Transp Res Dev.* 2024;41(4):204–13. doi:10.3969/j.issn.1002-0268.2024.04.021.
24. Huang H, Xie JY, Li ZH, Sun X, Peng T. Traffic flow forecasting method based on gated spatial-temporal graph network and TCN. *Transp Sci Technol.* 2024;6:126–31. doi:10.3963/j.issn.1671-7570.2024.06.024.
25. Zhang Y, Zhang L, Liu B, Liang Z, Zhang X. Multi-spatial scale traffic prediction model based on spatio-temporal transformer. *Comput Eng Sci.* 2024;46(10):1852–63. doi:10.3969/j.issn.1007-130X.2024.10.014.
26. Wang JD, Oktomy Noto Susanto C. Traffic flow prediction with heterogenous data using a hybrid CNN-LSTM model. *Comput Mater Contin.* 2023;76(3):3097–112. doi:10.32604/cmc.2023.040914.
27. Zhao Y, Lin Y, Wen H, Wei T, Jin X, Wan H. Spatial-temporal position-aware graph convolution networks for traffic flow forecasting. *IEEE Trans Intell Transp Syst.* 2023;24(8):8650–66. doi:10.1109/TITS.2022.3220089.
28. Li HH, Cao QX, Shan ZY, Zhao ZX. Multi-task traffic flow prediction based on attention spatial-temporal graph neural network. *Traffic Sci Eng.* 2025;41(6):65–74. doi:10.16544/j.cnki.cn43-1494/u.20230302002.
29. Zhang H, Yang G, Yu H, Zheng Z. Kalman filter-based CNN-BiLSTM-ATT model for traffic flow prediction. *Comput Mater Contin.* 2023;76(1):1047–63. doi:10.32604/cmc.2023.039274.
30. Li W, Liu X, Tao W, Zhang L, Zou J, Pan Y, et al. Location and time embedded feature representation for spatiotemporal traffic prediction. *Expert Syst Appl.* 2024;239(5):122449. doi:10.1016/j.eswa.2023.122449.
31. Xi Y, Yan X, Jia ZH, Yang B, Su R, Liu X. A dynamic multigraph and multidimensional attention neural network model for metro passenger flow prediction. *Concurr Comput Pract Exp.* 2024;36(18):e8140. doi:10.1002/cpe.8140.
32. Cao S, Wu L, Zhang R, Chen Y, Li J, Liu Q. A spatial-temporal gated hypergraph convolution network for traffic prediction. *IEEE Trans Veh Technol.* 2024;73(7):9546–59. doi:10.1109/TVT.2024.3365213.
33. Xia Z, Zhang Y, Yang J, Xie L. Dynamic spatial-temporal graph convolutional recurrent networks for traffic flow forecasting. *Expert Syst Appl.* 2024;240(3):122381. doi:10.1016/j.eswa.2023.122381.
34. Zhao W, Yuan G, Zhang Y, Liu X, Liu S, Zhang L. An interpretable and efficient multi-scale spatio-temporal neural network for traffic flow forecasting. *Expert Syst Appl.* 2026;296(24):128961. doi:10.1016/j.eswa.2025.128961.
35. Yu F, Chen Z, Xia X, Zong X. MSSTGCN: multi-head self-attention and spatial-temporal graph convolutional network for multi-scale traffic flow prediction. *Comput Mater Contin.* 2025;82(2):3517–37. doi:10.32604/cmc.2024.057494.
36. Ma L, Wang Y, Lv X, Guo L. Construction of a traffic flow prediction model based on neural ordinary differential equations and Spatiotemporal adaptive networks. *Sci Rep.* 2025;15(1):9787. doi:10.1038/s41598-025-92859-z.
37. Liu A, Zhang Y. Spatial-temporal dynamic graph convolutional network with interactive learning for traffic forecasting. *IEEE Trans Intell Transp Syst.* 2024;25(7):7645–60. doi:10.1109/TITS.2024.3362145.
38. Han S, Lee H, Lee DY, Kim SS, Yoon S, Lim S. Sequence-aware adaptive graph convolutional recurrent networks for traffic forecasting. *Knowl Based Syst.* 2025;330(4):114533. doi:10.1016/j.knosys.2025.114533.

39. Song C, Lin Y, Guo S, Wan H. Spatial-temporal synchronous graph convolutional networks: a new framework for spatial-temporal network data forecasting. *Proc AAAI Conf Artif Intell.* 2020;34(1):914–21. doi:10.1609/aaai.v34i01.5438.
40. Bai L, Yao L, Li C, Wang X, Wang C. Adaptive graph convolutional recurrent network for traffic forecasting. *Adv Neural Inf Process Syst.* 2020;33:17804–15. doi: 10.3390/math13244003.
41. Guo S, Lin Y, Feng N, Song C, Wan H. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. *Proc AAAI Conf Artif Intell.* 2019;33(1):922–9. doi:10.1609/aaai.v33i01.3301922.
42. Yu B, Yin H, Zhu Z. Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting. *arXiv:1709.04875.* 2017. doi:10.48550/arXiv.1709.04875.