Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

A Metro Passenger Flow Prediction Method Based on Adaptive Graph Convolution and Hybrid Attention

lu zeng
Jiangxi University of Science and Technology
qingshuo sha
Jiangxi University of Science and Technology
tao gao
986012723@qq.com
Jiangxi University of Science and Technology
qinzhen fang
Jiangxi University of Science and Technology

Research Article

Keywords: Urban Rail Transit, Short-term Passenger Flow Prediction, Bilinear Temporal Attention, Adaptive Graph Convolution, Hybrid Attention

Posted Date: August 3rd, 2026

DOI: https://doi.org/10.21203/rs.3.rs-10414796/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

A Metro Passenger Flow Prediction Method Based on Adaptive Graph Convolution and Hybrid Attention

Lu Zeng[12] • Qingshuo Sha[12] • Tao GAO[12] • Qinzhen Fang[12]

Abstract

Accurate short-term prediction of inbound passenger flow for urban rail transit is of great significance for network planning, smart services, and refined operations. However, existing methods cannot deeply fuse spatial topology, temporal periodicity, and trends, and fail to capture dynamic fluctuations and long-term evolution, causing low accuracy. To address this, we propose AGHA-Net, a method based on adaptive graph convolution and hybrid attention. The method first uses an adaptive graph convolutional network (AdaGCN) to accurately capture dynamically evolving spatial dependencies among stations. Then, a spatiotemporal gated recurrent unit (ST-GRU) deeply couples graph convolution with temporal recurrence, enabling efficient fusion and extraction of local spatiotemporal features. Furthermore, a dual-branch hybrid attention layer is introduced that uses sequential self-attention to capture long-range temporal dependencies. In contrast, a bilinear temporal attention mechanism performs dual-dimensional compression and constructs a temporal attention matrix to model internal correlations within long sequences. Consequently, it comprehensively improves prediction accuracy and stability. Experiments on Hangzhou metro data show AGHA-Net significantly outperforms baselines. Visualization verifies its ability to capture complex patterns, and ablation studies confirm each module's necessity. The proposed method provides reliable decision support for intelligent rail transit management, passenger flow control, and capacity optimization.

Keywords: Urban Rail Transit; Short-term Passenger Flow Prediction; Bilinear Temporal Attention; Adaptive Graph Convolution; Hybrid Attention

1. Introduction

With the rapid development of intelligent transportation systems, accurate, real-time passenger flow prediction is crucial. It boosts the operational efficiency of urban rail transit, eases congestion, and enables smart cities. It provides key data for station flow control, dynamic capacity dispatch, and the optimization of operational plans. It ensures safe and efficient rail transit and enhances passenger experience. Therefore, it is valuable for refined management and smart upgrading of urban rail transit.

Metro passenger flow data is spatiotemporally coupled. Spatially, station passenger flows are correlated and dynamic. Temporally, flows show nonlinearity, multiple periods, and long-term trends. The core challenge in short-term passenger flow prediction is twofold. First, to effectively capture dynamic spatial dependencies. Second, to accurately model complex temporal patterns. Traditional prediction paradigms rely mainly on statistical models, such as logistic regression(Apronti et al. 2016), ARIMA(Williams and Hoel 2003), and Kalman filtering(Okutani and Stephanedes 1984). They use linear time-series modeling to characterize patterns of passenger flow evolution. Such models have simple structures and are easy to deploy. However, because of

Qingshuo Sha
372987739@qq.com
1 School of Electrical Engineering and Automation, Jiangxi University of Science and Technology, Ganzhou 341000, China
2 Maglev College, Jiangxi University of Science and Technology, Ganzhou 341000, China

their linear assumptions, they cannot adequately capture the nonlinear dynamics of passenger flow sequences. Without spatial topological correlations, they cannot capture inter-node structural dependencies. Thus, they cannot capture the data’ s inherent spatiotemporal dependencies. Compared with traditional statistical methods, machine learning models have greater advantages in fitting nonlinear relationships. For example, K-NN(Zhang et al. 2013), Bayesian networks(Sun, Zhang, and Yu 2006), and support vector machines(Wu, Ho, and Lee 2004) have been widely applied to short-term traffic prediction tasks. These methods have improved prediction accuracy to a certain extent. Rail transit networks are expanding, and passenger flow data is growing exponentially. The limitations of traditional machine learning are becoming increasingly evident: reliance on manual feature engineering and insufficient generalization ability. Therefore, they cannot meet the demands of high-accuracy, real-time prediction for large-scale networks.

In recent years, deep learning has become a mainstream approach for passenger flow prediction(Teng et al. 2026; Yang et al. 2024). This is due to its powerful ability to automatically extract and represent features. Early studies mainly focused on Euclidean-structured data, using RNN(Ma et al. 2015), LSTM(Yang et al. 2021), and CNN(Ata et al. 2024; Singh, Sahana, and Bhattacharjee 2025) to capture temporal features and local spatial correlations, achieving some progress in short-term traffic prediction. However, these methods cannot effectively model the non-Euclidean topology of rail transit networks, limiting their ability to capture complex spatial relationships between stations(Bao et al. 2023; Guo and Zhang 2020). To overcome this limitation, graph convolutional networks (GCNs) were introduced to the field of traffic prediction. They can learn spatial dependencies directly from the topology of road networks. This provides an effective solution for modeling non-Euclidean structured data(Xiaohui Huang 2023).

Existing graph convolution-based spatiotemporal prediction models have achieved significant results(Bai et al. 2021; Chang, Liu, and Jia 2023) , but three prominent issues remain. First, most methods rely on predefined adjacency matrices to model spatial relationships. The adjacency weights are fixed and cannot be adaptively learned from data. Therefore, they struggle to accurately characterize the nonlinear spatial dependencies between stations. Second, temporal modeling mostly depends on a single recurrent or convolutional structure. Their long-range dependency capture mechanism is relatively limited. This makes it difficult for the models to finely distinguish multi-scale periodic patterns. It also makes it hard to capture long-term trends in passenger flow simultaneously. As a result, long-term predictions are prone to error accumulation. Third, the spatial and temporal modules are often simply connected in series. Spatiotemporal feature interaction is insufficient, and information fusion efficiency is low. This leads to limited model stability and generalization ability.

To address the above problem, this paper proposes a metro passenger flow prediction method based on adaptive graph convolution and hybrid attention (AGHA-Net). This method decomposes the metro network into two parts. One part is the static topological prior. The other part is the data-driven adaptive spatial correlation. Through adaptive graph convolution, spatio-temporal gated units, and a hybrid attention mechanism, this method jointly models dynamic and long-term spatial dependencies. The main contributions of this paper are as follows:

(1) We propose AGHA-Net, a model for predicting metro passenger flows. This model integrates adaptive graph convolution and hybrid attention. It adopts a modeling strategy that separates static topology from data-driven associations. This strategy solves two problems of traditional static graph convolution: reliance on a predefined adjacency matrix and difficulty in adaptively learning spatial dependencies between stations.

(2) We innovatively construct an ST-GRU spatio-temporal gated unit. This unit embeds the adaptive graph convolution module into the GRU gating structure. As a result, spatial aggregation and state updating evolve cooperatively within the same computational graph. This design breaks the separation between spatial and temporal modules in traditional models. It thereby enhances the real-time perception and memory capabilities for spatio-temporally coupled patterns such as peak aggregation and transfer coordination.

(3) We innovatively design a hybrid attention layer. This layer operates in parallel with sequential self-

attention and bilinear temporal attention. The former introduces positional encoding and one-dimensional convolution. It accurately captures long-range temporal dependencies across the globe. The latter compresses both channel and space dimensions. It then constructs a temporal attention matrix. This matrix efficiently models long-term trend correlations.

(4) Experiments were done on a real Hangzhou Metro dataset. They proved the model works well.

The remainder of this paper is organized as follows. Section 2 reviews related work. Section 3 provides a formal definition of the passenger flow prediction problem. Section 4 describes in detail the structure and core modules of the proposed AGHA-Net. Section 5 reports the experimental setup and result analysis. Finally, Section 6 concludes the paper and discusses future work.

2. Related Work

2.1. Traditional statistical and machine learning methods

Early traffic flow prediction was mainly based on traditional statistical methods. The most representative ones are Historical Average (HA)(Hamed and Al-Masaeid 1995) and the ARIMA model. These methods are easy to compute and deploy. They predict future trends by analyzing the time series characteristics of historical data. However, they are linear. They cannot capture the nonlinearity or time-varying spatiotemporal correlations of traffic flow. To address this issue, machine learning methods such as Support Vector Machines (SVM), Fuzzy Logic Systems (FLS)(Shirzadeh et al. 2021), and Particle Swarm Optimization (PSO) (Linchao Li 2019) have been introduced into the field of traffic prediction. SVM has been widely used in short-term traffic flow prediction due to its strong ability to fit nonlinear relationships. For example, Fu et al.(Fu et al. 2016) proposed a hierarchical multi-SVM framework for complex traffic scenarios. This framework improves the robustness of prediction and recognition by combining features and employing hierarchical decision-making. However, these methods heavily rely on manual feature extraction. Their procedures are cumbersome. They cannot also generalize in large-scale network scenarios.

2.2. Deep learning method

Deep learning has been successfully applied to traffic flow prediction tasks. It has powerful feature learning and representation capabilities. LSTM and GRU can effectively capture temporal correlations in traffic flow. CNN efficiently extracts spatial features. It leverages local perception and weight sharing. Combined models of the two have been widely used for traffic flow prediction(Wang et al. 2025). Bogaerts et al.(Bogaerts et al. 2020) proposed a graph CNN-LSTM hybrid model. This model uses graph convolution for spatial feature extraction. It uses an LSTM for modeling temporal dependencies. The model predicts both short-term and long-term traffic using trajectory data. Traffic networks have complex topology and non-Euclidean characteristics. Traditional CNN struggles to effectively model such structures. Graph convolutional networks (GCNs) can learn directly from graph structures. They can better represent the spatial dependencies in traffic data. For example, SSNN proposed by Van et al. (Lint, Hoogendoorn, and Zuylen 2002) combines state-space dynamics with recurrent neural networks. This model models the evolution of traffic states on highway segments. It effectively captures nonlinear spatiotemporal dependencies. CLTFP proposed by Wu et al. (Wu and Tan 2016)integrates one-dimensional convolution with LSTM. This model simultaneously captures spatial features, short-term variations, and long-term periodic patterns of traffic flow. The DeepSTN proposed by Lin et al. (Lin et al. 2019) combines deep convolutional neural networks with semantic perception modules. This model jointly models long-range spatial dependencies, location semantics, and temporal features for citywide crowd flow.

However, these methods typically rely on a predefined Laplacian matrix to capture node correlations. As a result, they struggle to model spatiotemporal correlations that evolve dynamically. To address this issue, related

studies have turned to data-driven approaches. They adaptively generate dynamic correlation matrices. Yu et al. (Yu, Yin, and Zhu 2017) proposed STGCN. It accurately models the spatial correlations of road network topology. It uses graph convolution modules. It also uses gated temporal convolution to extract long-term temporal dependencies. Song et al. (Song and Ren 2024) proposed STACGCN. It uses channel attention and adaptive graph convolution. It captures complex spatiotemporal dependencies. This model targets traffic speed prediction tasks. Dai et al. (Dai, Xu, and Gu 2020)proposed a hybrid spatiotemporal graph convolutional network called H-STGCN. This model can accurately infer future travel times. It does so based on forward-looking traffic volume data. Yao et al.(Yao and Zhou 2025) proposed AG-NCDE. It integrates adaptive graph convolution with neural-controlled differential equations. This model captures the continuous spatiotemporal dependencies in traffic flow. Liu et al.(Liu et al. 2024) proposed ST-DAGCN. It integrates dual adaptive graph convolution with gated recurrent units. This model adaptively learns global static features. It also learns local dynamic features of the road network. It effectively captures nonlinear spatiotemporal dependencies.

2.3. Attention-based method

The aforementioned adaptive graph convolution effectively improves spatiotemporal dependency modeling. However, when attention mechanisms are fused with graph networks, the resulting model demonstrates even stronger global feature capture performance in traffic prediction. Gao et al. (Gao et al. 2024)proposed PMR-GCN. This model uses P-GCN with a learnable diagonal matrix. It improves the accuracy of peak passenger flow prediction. It also employs multi-head self-attention to efficiently capture global spatiotemporal features. The model combines the Fourier transform and KL divergence regularization. It then achieves better prediction of metro passenger flow. Benarmas et al.(Benarmas and Bey 2024)proposed a three-stage deep learning framework for traffic prediction. This framework uses an attention-enhanced LSTM. It models key influencing road segments in the network. It also uses an attention-driven generative adversarial module. This module extracts long-term temporal correlations. Jiang et al.(Jiang et al. 2023)proposed PDFormer. It models dynamic, long-range spatial correlations in road networks via spatial self-attention. It extracts long-term temporal features using temporal self-attention. It also explicitly models traffic propagation delays. It does so using a delay-aware feature transformation. Feng et al. (Feng and Tassiulas 2022)proposed ASTTN. It is a traffic prediction model. This model jointly models spatiotemporal correlations. It stacks local spatiotemporal multi-head self-attention layers. It also optimizes attention weights through adaptive graph learning. Yan et al. (Yan et al. 2024)proposed ProSTformer. It is a pure self-attention model. This model uses progressive spatiotemporal multi-head self-attention, tensor rearrangement, and patch merging techniques to progressively capture spatiotemporal dependencies from local to global scales, thereby significantly reducing the computational cost.

Compared with existing work, AGHA-Net has the following advantages:

(1) Through adaptive graph convolution, a low-dimensional embedding is learned for each station in the metro network. This method adaptively generates an adjacency relation matrix. It does not rely on a predefined Laplacian matrix. It can simultaneously capture both long-term, stable, static topology and time-varying, dynamic spatial correlations. Thereby, it achieves a unified representation of static topological priors and data-driven associations. This effectively avoids the fragmentation of spatiotemporal modeling.

(2) We adopt a dual-branch parallel mechanism. It consists of sequential self-attention and bilinear temporal attention. The former accurately captures long-range temporal dependencies at the global scale. The latter efficiently models long-term trend correlations through dual-dimensional compression. The ST-GRU is used to perform gated pre-compression on temporal information. This reduces the computational complexity of long-sequence attention. It thereby improves model efficiency.

(3) After ST-GRU encoding, residual connections are introduced. A feature alignment mechanism is also introduced. This mechanism deeply fuses two types of features. One is the original spatiotemporal features. The other is the dual-attention enhanced features. This helps preserve low-level spatiotemporal information. It also

alleviates the vanishing gradient problem in deep networks. Finally, it achieves joint optimization of multi-level spatiotemporal representations.

To emphasize the importance of this study, we conducted a comprehensive review of the existing literature. We also systematically compared our proposed model with previous studies. The detailed comparison results are shown in Table 1.

Table 1 Comparison of the similar studies

References	Spatial Evolvability	Spatiotemporal Synergy	Periodic Decoupling	Attribute Dynamic Graph	Modules
(Wang, Yue, et al. 2026)	Flow-Adaptive Dynamic Graph Attention Mechanism	Transformer	×	×	MTS-GAT
(Xing et al. 2025)	Dynamic Multi-Graph Fusion Mechanism	Attention, Transformer	√	×	MGLTN
(Wang, Chen, et al. 2026)	Embedding Inner-Product Mechanism	GRU	×	×	FedCTPM
(Yu et al. 2025)	Dynamic Temporal Adjacency Generation Mechanism	Transformer	×	×	CLEAR
(An et al. 2025)	Dynamic Spatial Graph Generation and Multi-Graph Fusion	Transformer	×	×	STMPNet
(Chen et al. 2025)	Cross-Attention Evolution	Cross Spatial-Temporal Attention	×	√	DTRformer
(Cai et al. 2026)	Semantic Dynamic Graph Evolution	Transformer	×	×	Ada-STSA
(Mao, Han, and Wang 2026)	Dynamic Similarity Hypergraph Evolution	TCN	×	×	HR-DHAN
(Zhang et al. 2026)	Dynamic Masking Global Spatial Attention	Transformer, Attention	×	×	DGSSformer
This Study	Adaptive Graph Convolution, Spatiotemporal Gated Recursive	GRU, Bilinear temporal attention	√	√	AGHA-Net

Notes: ‘x’ denotes the Issue has not been considered, and ‘ √ ’ denotes the issue has been considered.

3. Problem statement and preparation work

Urban rail transit passenger flow prediction is a typical spatiotemporal sequence prediction task. It is based on historical passenger flow observation sequences of metro stations. It also uses the network topology. Then it accurately predicts inbound passenger flow at multiple future time steps. This provides reliable data support for dynamic capacity deployment, collaborative passenger flow control, and scientific operational decision-making.

The rail transit network is abstracted as an undirected graph $G = (V, E, A)$, where V is the set of station nodes, E is the set of physical connection edges between stations, $A \in R^{N \times N}$ is the adjacency matrix describing the topological structure of the network, and N is the total number of stations in the network.

The historical passenger flow observation sequence is defined as $X \in R^{N \times F \times \tau}$,where F is the feature dimension of a single station, and τ is the length of the historical observation time steps. At time step t, the input sequence can be expressed as:

$$X = \{X^{t-\tau}, X^{t-\tau+1}, \dots, X^{t}\} \tag{1}$$

Where $X^{t} \in R^{N \times F}$ is the passenger flow observation vector for all stations at time step t.

The passenger flow prediction task can be formalized as learning a mapping function $f(\cdot)$. This function maps the historical spatiotemporal data. It maps to the predicted passenger flow sequence $\hat{Y} \in R^{N \times T_P}$. The prediction covers the next T_P time steps. That is:

$$\hat{Y} = f(X, A) \tag{2}$$

$$\hat{Y} = \left\{Y^{t+1}, Y^{t+2}, \dots, Y^{t+T_p}\right\} \tag{3}$$

Where $Y^{t+T_p} \in R^{N}$ is the predicted passenger flow result for all stations at time step $t + T_p$.

To help readers better understand the three key modules proposed in Section 4. They are an adaptive graph convolution, a spatiotemporal gated unit, and a hybrid attention mechanism. The following briefly describes three basic models.

(1) Graph Convolutional Network

For a graph $G = (V, E, A)$, with adjacency matrix $A \in R^{N \times N}$ and degree matrix $D_{ii} = \sum_j A_{ij}$, the standard graph convolutional layer is defined as:

$$H^{l+1} = \sigma\left(\hat{D}^{-\frac{1}{2}}\hat{A}\hat{D}^{-\frac{1}{2}}H^{l}W^{l}\right) \tag{4}$$

Where $\hat{A} = A + I_N$, $\hat{D}$ is the degree matrix of $\hat{A}$, H^{l} is the node features of the $l-$th layer, W^{l} is a learnable weight matrix, and σ is an activation function. This operation achieves spatial information propagation through neighbor aggregation.

(2) Gated Recurrent Unit

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z) \tag{5}$$

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r) \tag{6}$$

$$\tilde{h}_t = \tanh(W_h x_t + U_h(r_t \odot h_{t-1}) + b_h) \tag{7}$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \tag{8}$$

GRU can alleviate the vanishing gradient problem in long-term dependencies. Its structure is also simpler than that of LSTM.

(3) Self-Attention Mechanism

Given the query matrix Q, key matrix K, and value matrix V, the self-attention output is:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^{T}}{\sqrt{d}}\right)V \tag{9}$$

By computing the correlation between any two positions within a sequence, self-attention can capture global dependencies. However, it lacks explicit positional information. Therefore, it is usually combined with positional encoding.

Section 4 of this paper extends the above basic models in several ways. Learnable node embeddings are used to construct adaptive graph convolution (AdaGCN). Graph convolution is embedded into the GRU to form a spatiotemporal gated unit (ST-GRU). Positional encoding and convolution enhancement are introduced into self-attention to obtain sequential self-attention (SSAM). Meanwhile, bilinear temporal attention (TABL) is designed. This attention reduces computational overhead and strengthens the modeling of long-range periodic patterns.

4. Methodology

The proposed AGHA-Net architecture consists of three cascaded components. They are the Spatiotemporal Gated Recursive layer, the Hybrid Attention layer, and the Feature Processing layer. This is illustrated in Figure 1. The method models dynamic spatial dependencies. It also extracts local spatiotemporal features. It captures global long-range temporal dependencies. The model can uniformly represent the complex evolution patterns of metro passenger flow. It thereby achieves accurate prediction.

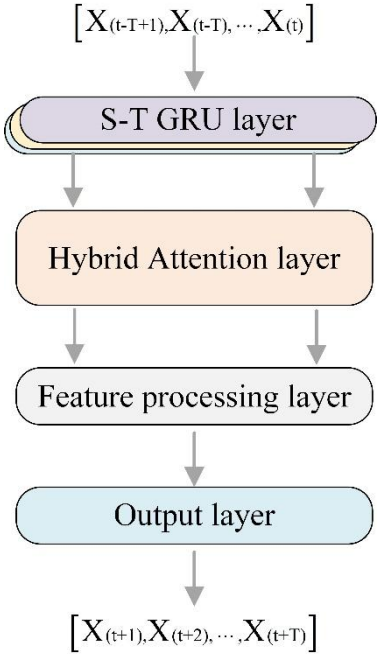


Fig.1 AGHA-Net model architecture

4.1. Spatiotemporal Gated Recursive layer (ST-GRU)

The spatial correlation patterns of urban rail transit networks vary significantly across different time periods. This is especially true during morning and evening peaks. It is also true during transfer gathering and commuting fluctuations. Under these scenarios, passenger flow exhibits time-varying spatial dependencies. Traditional static graph convolution struggles to capture such dynamic characteristics. It cannot effectively model them. Meanwhile, passenger flow sequences have strong temporal continuity. They also have local fluctuation features. These features require memory-capable temporal units. Such units are needed for effective modeling. To address this, we design a spatiotemporal gated recursive layer. We term it ST-GRU. This layer embeds adaptive graph convolution into a gated recurrent unit. The resulting unified structure can simultaneously model spatial correlations. It can also model temporal dependencies. It achieves efficient extraction of local spatiotemporal features. It also enables iterative updates to these features.

4.1.1. Adaptive Graph Convolution Module (AdaGCN)

Traditional graph convolutional networks rely on predefined adjacency and Laplacian matrices for spatial information aggregation. In these networks, the adjacency weights are fixed once set. This prevents the model from adaptively learning the true spatial dependencies between stations from the data. To address this issue, this paper proposes an adaptive graph convolution mechanism (AdaGCN). It is illustrated in Figure 2. This mechanism adaptively generates adjacency relationships through learnable node embeddings.

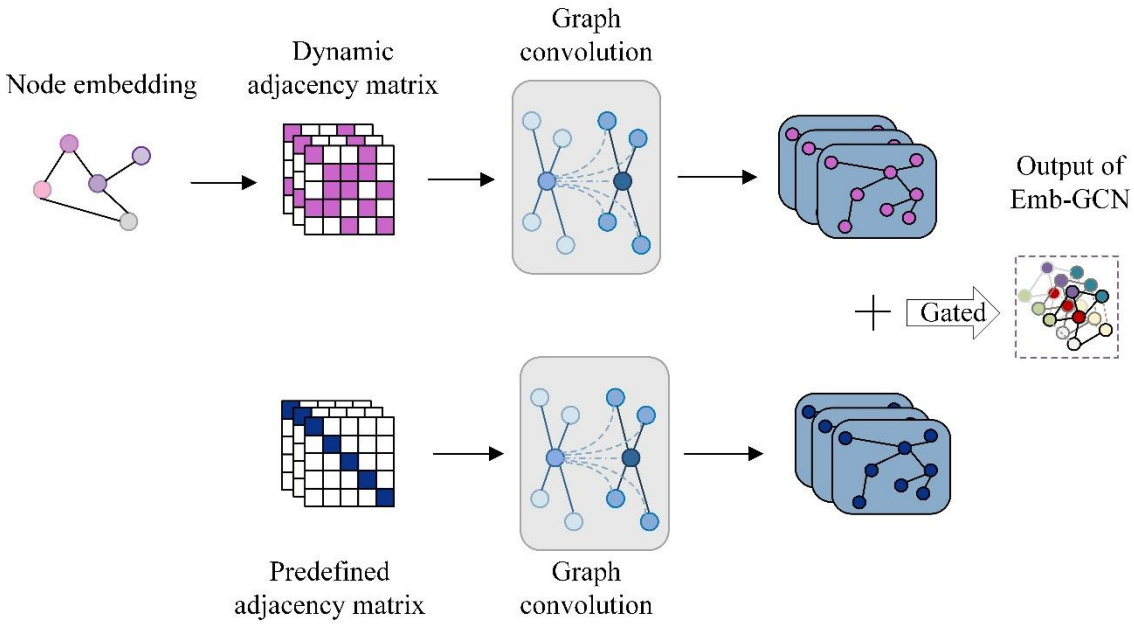


Fig.2 Adaptive Graph Convolutional Module

It retains the network's static topological prior. At the same time, it adaptively learns the strength of spatial correlations between stations from the data. The standard graph convolution operation can be expressed as:

$$\mathrm{GCN}(\mathrm{A},\mathrm{X}) = \sigma\left(\mathrm{I_N} + \mathrm{D}^{-\frac{1}{2}}\mathrm{A}\mathrm{D}^{-\frac{1}{2}}\right)\mathrm{X}\mathrm{W} + \mathrm{b} \tag{10}$$

Here, A is the network adjacency matrix. D is the node degree matrix. $\mathrm{I_N}$ is the identity matrix. X is the station passenger flow feature matrix. W and b are learnable parameters. $\sigma(\cdot)$ is the nonlinear activation function.

To break free from reliance on a fixed adjacency matrix, this paper introduces a learnable node embedding matrix $\mathrm{E_{adj}} \in \mathrm{R}^{\mathrm{N}\times\mathrm{d_e}}$. This matrix computes similarities via the inner product of node embeddings. Then, after nonlinear activation and normalization, we obtain the adaptive adjacency matrix $\tilde{\mathrm{A}}$:

$$\tilde{\mathrm{A}} = \mathrm{softmax}\left(\mathrm{ReLU}\left(\mathrm{E_{adj}} \cdot \mathrm{E_{adj}^{T}}\right)\right) \tag{11}$$

Here, $\mathrm{E_{adj}} \in \mathrm{R}^{\mathrm{N}\times\mathrm{d_e}}$ is the learnable embedding matrix for stations. N is the total number of stations. $\mathrm{d_e}$ is the embedding dimension. Each station is mapped to a low-dimensional dense vector through this matrix. $\mathrm{ReLU}(\cdot)$ is the rectified linear unit function. It performs a nonlinear transformation on the inner product similarities. It sets negative correlation weights to zero. This only retains positive associations. Thereby, it enhances the sparsity of the adjacency matrix. $\mathrm{softmax}(\cdot)$ is the normalized exponential function. It maps the weights along rows to the interval $(0,1)$. This yields a dynamic adjacency relation matrix with probabilistic semantics. This matrix is directly used for spatial information aggregation, which evolves with node embeddings to reflect time-varying correlations. At this point, the graph convolution operation simplifies to:

$$\mathrm{EmbGCN}(\mathrm{X}) = \tilde{\mathrm{A}}\mathrm{X}\mathrm{W} + \mathrm{b} \tag{12}$$

Here, $\tilde{\mathrm{A}}$ incorporates both static topological priors and dynamic data-driven relationships. This matrix is directly used for spatial information aggregation.

This structure can reduce computational overhead. At the same time, it significantly enhances the model's ability to capture time-varying and nonlinear spatial dependencies.

4.1.2 Spatiotemporal Gated Recurrent Unit (ST-GRU Unit)

This paper embeds AdaGCN into the GRU gating mechanism. It thereby constructs a spatiotemporally

coupled ST-GRU unit. This unit simultaneously performs spatial information aggregation at each time step of the temporal update. This is illustrated in Figure 3.

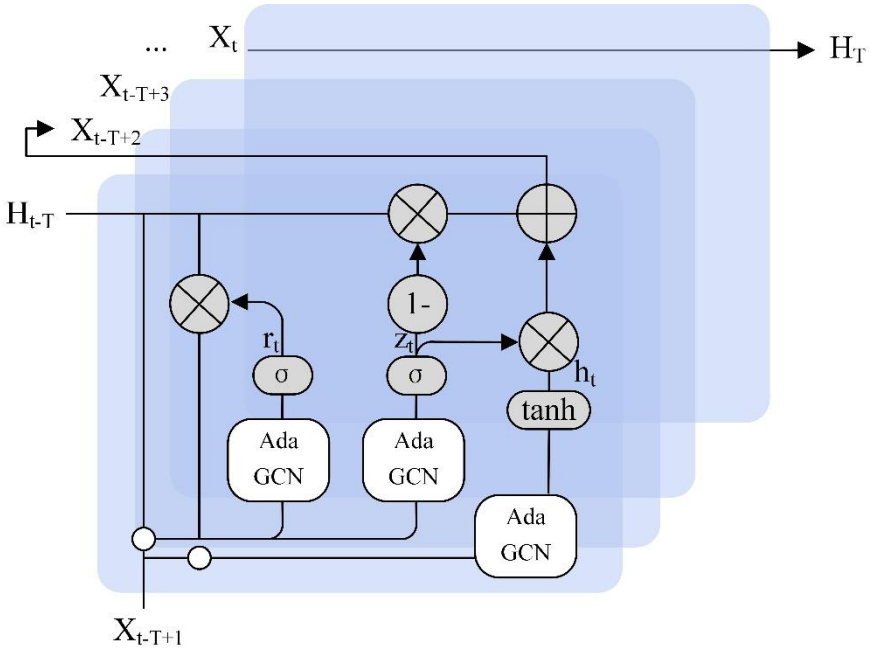


Fig.3 Spatiotemporal Gated Recurrent Unit

It achieves joint updating and propagation of spatial features and temporal states. As a result, it effectively captures local spatiotemporal dependencies in passenger flow sequences. The mathematical formulas are as follows:

$$z_t = \sigma\left(\mathrm{AdaGCN}_z\left(\tilde{A}, X_t \parallel H_{t-1}\right)\right) \tag{13}$$

$$r_t = \sigma\left(\mathrm{AdaGCN}_r\left(\tilde{A}, X_t \parallel H_{t\text{-}1}\right)\right) \tag{14}$$

$$\tilde{h}_t = \tanh\left(\mathrm{AdaGCN}\left(\tilde{A}, X_t \parallel \left(r^t \odot H_{t\text{-}1}\right)\right)\right) \tag{15}$$

$$H_t = z^t \odot H_{t-1} + (1 - z^t) \odot \tilde{h}^t \tag{16}$$

Here, z_t is the update gate, and r_t is the reset gate. The two gates share the same input structur e but use separate weight matrices. $\tilde{h}_t$ is the candidate hidden state. H_t is the output at the current ti me step. $\parallel$ denotes feature concatenation. $\odot$ denotes element-wise multiplication. $\sigma(\cdot)$ is the sigmoid activation function.

By stacking multiple ST-GRU units, the model can progressively deepen its ability to extract and memorize local spatiotemporal patterns. This enables it to better adapt to short-term fluctuations and the continuous evolution of passenger flow.

4.2. Hybrid Attention layer

The ST-GRU layer performs well in extracting local spatiotemporal features. However, when dealing with long temporal sequences, it tends to suffer from insufficient modeling of long-range dependencies. To address this limitation, this paper designs a hybrid attention layer. This layer consists of two parallel branches: sequential self-attention and bilinear temporal attention. The former captures long-range temporal dependencies at the global scale. The latter captures long-term trend correlations. Together, they achieve comprehensive enhancement of temporal features.

4.2.1. Sequential Self-Attention Module (SSAM)

Traditional self-attention mechanisms process sequential data. They tend to lose temporal order information during this process. Therefore, they struggle to model passenger flow's sequential dependencies effectively. To address this, this paper introduces temporal positional encoding into self-attention, as shown in Figure 4.

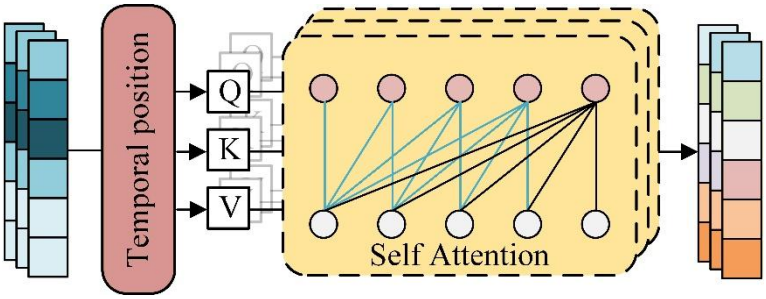


Fig.4 Sequential Self-Attention Module

Meanwhile, one-dimensional convolution is applied to the queries and keys. This enhances local perception. It allows the model to capture local correlations. It also preserves the integrity of the temporal structure. The mathematical formulas are as follows.

$$\mathrm{Att}(\mathrm{Q},\mathrm{K},\mathrm{V}) = \mathrm{softmax}\left(\frac{\mathrm{Q_c}(\mathrm{Q})\cdot\mathrm{K_c}(\mathrm{K})^{\mathrm{T}}}{\sqrt{\mathrm{d}}}\right)\mathrm{V_l}(\mathrm{V}) \tag{17}$$

$$\mathrm{Q_c}(\mathrm{Q}) = \mathrm{Conv}(\mathrm{Q}) \tag{18}$$

$$\mathrm{K_c}(\mathrm{K}) = \mathrm{Conv}(\mathrm{K}) \tag{19}$$

$$\mathrm{V_l}(\mathrm{V}) = \mathrm{Linear}(\mathrm{V}) \tag{20}$$

Here, Q, K, and V present the query, key, and value vectors, respectively d is the feature dimension. $\mathrm{Conv}(\cdot)$ is one-dimensional convolution. $\mathrm{Linear}(\cdot)$ is a linear transformation.

This module uses residual connections and layer normalization. They stabilize the training process. It effectively alleviates the vanishing gradient problem in deep networks. Thereby, it significantly improves the model’s ability to capture long-term dependencies and periodic patterns.

4.2.2. Bilinear Temporal Attention Block (TABL)

Sequential self-attention captures long-range temporal dependencies. It does so through global self-attention. However, its full QKV projection mechanism incurs a high computational cost. This is especially true on long sequences. Meanwhile, its dot-product interaction is limited. It cannot explicitly model multi-scale periodic patterns well. To address this, this paper designs a bilinear temporal attention module. It is called TABL. This module replaces the self-attention projection. It uses dual-perspective convolutional compression and bilinear interaction instead. This reduces computational overhead. At the same time, it enhances the ability to capture long-range periodic patterns. It also captures structured trend correlations. This is illustrated in Figure 5.

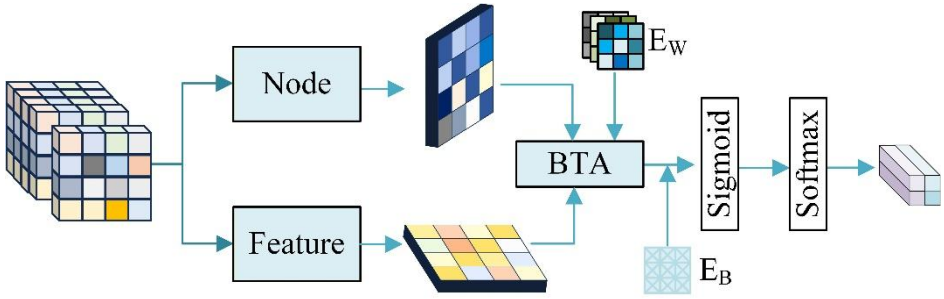


Fig.5 Bilinear Temporal Attention Block

Let the input sequence be $\mathrm{X} \in \mathrm{R}^{\mathrm{B}\times\mathrm{T}\times\mathrm{N}\times\mathrm{D}}$. Here, B is the batch size. T is the number of time steps. N is

the number of nodes. D is the feature dimension. First, its dimensions are rearranged to $S \in R^{B\times D\times N\times T}$:

$$S = \mathrm{Permute}\big(X, (0,3,2,1)\big) \tag{21}$$

Here, $\mathrm{Permute}(\cdot)$ denotes a tensor dimension permutation. It moves the channel dimension D forward. It places the temporal dimension T last. This makes the feature dimension the input channel for convolution.

The tensor S is rearranged to $C_1 \in R^{B\times D\times T\times N}$. Then, a convolution kernel $W_{c1} \in R^{1\times 1\times D\times 1}$ is applied. It compresses the feature dimension to 1. Finally, we obtain the node-level time-step representation $F_1 \in R^{B\times T\times N}$:

$$F_1 = \mathrm{Squeeze}\big(\mathrm{Conv2d}(C_1, W_{c1})\big) \tag{22}$$

Here, $\mathrm{Conv2d}(\cdot)$ denotes two-dimensional convolution. $\mathrm{Squeeze}(\cdot)$ removes channel dimensions of size 1.

The tensor S is rearranged to $C_2 \in R^{B\times N\times D\times T}$. Next, a convolution kernel $W_{c2} \in R^{1\times 1\times N\times 1}$ is applied. It compresses the feature dimension to 1. This yields the channel-level time-step representation $F_2 \in R^{B\times D\times T}$:

$$F_2 = \mathrm{Squeeze}(\mathrm{Conv2d}(C_2, W_{c2}),\mathrm{axis} = 1) \tag{23}$$

Here, F_1 captures the node-level representation of time steps from a feature perspective. F_2 captures the channel-level representation of time steps from a spatial perspective.

Introduce a learnable parameter matrix $W \in R^{N\times D}$. Also, introduce a learnable bias $B \in R^{T\times T}$. Then construct the temporal attention logits matrix $L \in R^{B\times T\times T}$:

$$L = \sigma(F_1 W F_2 + B) \tag{24}$$

Here, $\sigma(\cdot)$ is the Sigmoid activation. During computation, $F_1 W$ is first mapped to $R^{B\times T\times D}$. Then it performs an inner product with F_2 along the feature dimension. This forms a bilinear form. To enhance the model's ability to represent interactions across different time steps, a learnable linear transformation matrix $V \in R^{T\times T}$ is introduced. It performs cross-time-step feature reorganization on L. Then batch normalization is applied:

$$L' = \mathrm{BatchNorm1d}(VL) \tag{25}$$

Here, VL performs a linear reorganization of the attention logits matrix across time steps. $\mathrm{BatchNorm1d}(\cdot)$ normalizes along the temporal dimension. This stabilizes training.

Apply Softmax normalization to L'. This yields the attention weight matrix $A \in R^{B\times T\times T}$:

$$A_{ij} = \frac{\exp(L_{ij})}{\sum_{k=1}^{T} \exp(L_{ik})} \tag{26}$$

Here, A_{ij} is the normalized attention weight from time step i to time step j. It satisfies $\sum_j A_{ij} = 1$.

TABL replaces the full projection of traditional self-attention. It uses convolutional compression and bilinear operations instead. This method reduces computational overhead. At the same time, it retains the ability to model long-range temporal correlations.

4.3. Feature processing layer

The hybrid attention layer can effectively capture global long-range temporal dependencies. However, its output spatiotemporal features may suffer from spatial representation inconsistency. They may also suffer from a bias in attention weight distribution. Directly mapping these features to the prediction space may introduce spatial structure noise. This would weaken prediction accuracy. To address this, we construct a feature processing layer. This layer performs three functions. They are spatial feature alignment, information consistency calibration, and prediction mapping. It uses a predefined adjacency matrix as a spatial inductive bias. It explicitly injects physical topology priors into the feature aggregation process. This guides the model to prioritize real interactions between spatially adjacent stations. It also suppresses spurious responses. These responses are caused by cross-region non-physical correlations. At the same time, it performs structured aggregation of neighborhood information. It refines the spatial consistency of the upstream extracted spatiotemporal features. This ensures that the final feature representation for prediction accounts for physical topology constraints. It also accounts for

data-driven dependencies. Thereby, it improves prediction stability. Finally, one-dimensional convolution is used to map the features to the prediction space. It outputs the passenger flow prediction for the next time step.

5. Experiments

To validate the effectiveness, robustness, generalization capability, and long-term prediction stability of the proposed AGHA-Net model, this chapter conducts comparative experiments. The experiments are carried out from multiple perspectives, including dataset configuration, baseline models, evaluation metrics, overall performance, typical site analysis, ablation studies, and verification of key module effectiveness. Quantitative analysis evaluates the model's advantages in capturing dynamic spatial dependencies, extracting local spatiotemporal features, and modeling long-range temporal dependencies.

5.1. Dataset Overview

This paper uses inbound passenger flow data for the Hangzhou metro from January 1 to January 25, 2019, as the experimental subject. The data covers three metro lines and 80 stations. The operating hours are 6:00 to 23:00. The dataset includes typical passenger flow scenarios, including weekdays and weekends. It encompasses various patterns, including commuting peaks, off-peak periods, and interchange passenger flow. Passenger flow data is aggregated at 15-minute, 30-minute, and 45-minute intervals. This fully accommodates the different forecasting cycle requirements of metro operations. In the experiment, the dataset is split into training, validation, and test sets at 6:2:2. This ensures a reasonable data split and experimental reproducibility. Furthermore, to facilitate fine-grained ablation analysis and multi-step prediction performance evaluation, the original automatic fare collection (AFC) data is additionally aggregated at 5-minute and 20-minute intervals. These intervals, together with the 15-minute and 30-minute intervals, constitute multi-level prediction step sizes.

5.2. Baseline model

To objectively and comprehensively verify the effectiveness of the AGHA-Net model, this paper selects a variety of models to form a complete baseline system. These models include traditional statistical models, classical deep learning models, and mainstream spatiotemporal graph neural network models. The details are as follows:

(1) Traditional methods: Historical Average model (HA);

(2) Classical deep learning models: Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), Graph Neural Network (GCN);

(3) Advanced spatiotemporal graph networks: Attention-based Spatio-Temporal Graph Neural Network (ASTGNN)(Guo et al. 2022), Attention-based Spatio-Temporal Graph Convolutional Network (ASTGCN)(Ye, Xue, and Jiang 2022), Temporal Graph Convolutional Network (T-GCN)(Zhao et al. 2020), Physics-Virtual Cooperative Graph Network (PVCGN)(Liu et al. 2022), Propagation Delay-Aware Dynamic Long-Range Transformer (PDFormer)(Jiang et al. 2023), Long-term Correlations Dual-graph Transformer (LCDFormer)(Cai, Wang, and Hu 2024), Hybrid Spatial-Temporal Graph Neural Network (HSTGNN)(Wang et al. 2025), and Spatial-Temporal Convolutional Long Short-Term Memory Network (ST-ConvLSTMNet)(Peng, Cheng, and Li 2026).

5.3. Experimental setup and evaluation metrics

The experiment uses four historical time steps as input. It predicts the inbound passenger flow for the next time step. This covers the need for short-term, accurate prediction. The hardware environment is an NVIDIA GeForce RTX 4060. The code is implemented based on Python and the PyTorch framework. The key model parameters are as follows: the GRU hidden dimension is 64. The batch size is 4. The Adam optimizer is used with an initial learning rate of 0.001. The number of training epochs is set to 80.

To comprehensively evaluate the model's prediction accuracy, this paper uses three mainstream metrics. They are Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and Root Mean Square Error (RMSE). Smaller values indicate better prediction performance. The calculation formulas are as follows:

$$E_{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - x_i| \tag{27}$$

$$E_{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - x_i}{x_i}\right| \tag{28}$$

$$E_{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - x_i)^2} \tag{29}$$

Where n is the number of samples, y_i is the predicted value, and x_i is the true value.

5.4. Comparison of comprehensive performance of models

This study sets three prediction granularities: 15, 30, and 45 minutes. We compare the model's prediction accuracy across different time scales. The results are shown in Table 2. Experiments show that compared with all baseline models, the proposed AGHA-Net achieves the best performance across all three metrics. The prediction accuracy is significantly improved.

Table 2 Prediction accuracy of AGHA-Net model and baseline model

Model	15min			30min			45min		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
HA	54.23	59.91	50.48%	55.22	62.12	51.23%	57.84	64.77	52.79%
LSTM	29.43	39.61	51.32%	31.32	39.86	65.82%	35.93	44.81	52.61%
GRU	30.21	37.48	55.69%	35.89	46.71	60.93%	37.43	49.92	62.27%
GCN	50.00	35.48	70.71%	53.12	38.71	72.89%	56.25	41.94	75.00%
ASTGNN	44.88	74.81	60.33%	46.26	77.64	60.86%	48.01	81.12	62.05%
ASTGCN	27.78	68.40	26.56%	26.86	71.35	28.93%	35.34	76.49	36.01%
T-GCN	24.85	35.47	40.57%	30.32	44.54	37.06%	31.42	47.09	45.90%
PVGCN	24.71	40.23	46.12%	26.67	44.52	49.19%	30.24	46.81	52.13%
PDFormer	25.31	44.18	26.56%	26.89	47.18	28.56%	28.01	48.76	30.56%
LCDFormer	28.73	44.17	29.60%	29.36	45.38	29.81%	30.06	46.61	31.07%
HSTGNN	25.80	43.10	26.48%	26.10	43.66	26.82%	26.62	44.47	27.18%
ST-ConvLSTMNet	24.01	38.18	30.56%	25.03	39.97	32.01%	26.01	41.98	34.09%
AGHA-Net	**21.65**	**34.68**	**21.83%**	**22.37**	**35.92**	**23.06%**	**22.96**	**36.91**	**24.43%**

The experimental results show that the proposed AGHA-Net significantly outperforms all baseline models. This holds across all prediction granularities and three evaluation metrics. Its core advantages are as follows. The model adaptively captures time-varying spatial dependencies through AdaGCN-based adaptive graph convolution. It combines the ST-GRU spatiotemporal gating unit. This efficiently extracts local spatiotemporal features. Meanwhile, it relies on a hybrid attention layer. This layer consists of sequential self-attention and bilinear temporal attention. It accurately captures long-range temporal dependencies. Thus, the model achieves a unified representation of normal passenger flow, peak passenger flow, and interchange passenger flow. In contrast, HA relies solely on historical averages for prediction. It is completely unable to capture the complex spatiotemporal characteristics of metro passenger flow. Therefore, its performance is the worst across all metrics. LSTM and GRU can capture more temporal correlations through their gating mechanisms. Their prediction accuracy is better than that of HA. However, due to their lack of spatial modeling capability, their performance improvement is limited. GCN can extract the topological features of the road network. But it lacks sufficient modeling of temporal patterns. This makes it difficult to achieve high-precision temporal prediction tasks on its

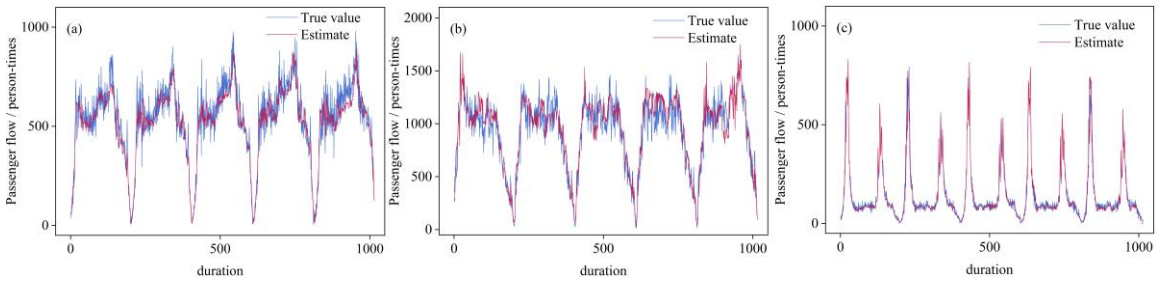
own. Spatiotemporal graph models such as LCDFormer , PDFormer , and HSTGNN further improve prediction performance. They integrate passenger flow features from both temporal and spatial dimensions. HSTGNN also includes optimizations for spatiotemporal feature learning. However, most of these methods rely on static graph structures and single recurrent units, and thus still struggle to effectively capture the time-varying spatial dependencies and long-range temporal dependencies that evolve dynamically with passenger flow in the metro network.

In summary, through the collaborative design of adaptive graph convolution, spatiotemporal gating coupling, and dual-branch hybrid attention, AGHA-Net demonstrates significant advantages in dynamic spatial modeling. It also excels at capturing long-range temporal dependencies. The MAE growth from 15 to 45 min (6.1%) is far below ASTGCN's 27.2%, confirming error accumulation suppression.

5.5. Performance analysis

5.5.1. Adaptability analysis of typical stations

Passenger flow patterns vary significantly across different metro stations. Therefore, prediction performance may differ notably across station types. To comprehensively verify the AGHA-Net model's adaptability to different passenger flow patterns, this section selects three stations with typical flow characteristics for in-depth evaluation. Station A is a commuter station. It exhibits typical morning and evening double-peak features. The morning peak passenger flow is significant, and the commuting pattern is strong. Station B is a commercial area station. It does not have obvious commuter-type double-peak features. However, the passenger flow remains high throughout the day. The overall passenger flow volume is significantly higher than that of other station types. Station C is a residential area station. It does not have obvious commuter-type double-peak features. The evening peak in inbound passenger flow is pronounced. The overall passenger flow volume is at a medium-to-low level throughout the day. Nighttime and weekend passenger flow activity is notably lower.



(a) Site A; (b) Site B; (c) Site C

Fig.6 Single-site prediction results

(1) Figure 6 shows the prediction results of AGHA-Net at three types of stations: A, B, and C. As seen in Figure 6(a), AGHA-Net achieves excellent prediction performance at Station A (a commuter station). On weekdays, passenger flow exhibits clear morning and evening peaks. The morning peak flow is significantly higher than the evening peak. This aligns with typical commuter travel patterns. Weekend passenger flow is relatively flat. It is also significantly lower than weekday flow. The predicted curve closely follows the periodic variations of the true values. The model effectively identifies and captures the tidal patterns of passenger flow at commuter stations. It maintains high prediction accuracy during both weekday peaks and weekend off-peak periods. This fully verifies the model's adaptability to regular passenger flow patterns.

(2) Figure 6(b) shows the passenger flow prediction performance at Station B. Station B is a commercial area station. Unlike other stations, it does not have obvious commuter characteristics. However, its daily passenger flow is significantly higher than that of Stations A and C. The passenger flow volume during peak hours is substantial. AGHA-Net faces challenges of high passenger flow and complex fluctuation patterns.

Despite these challenges, it can still accurately track the overall trend of passenger flow. It also tracks the periodic patterns. The predicted curve maintains good consistency with the true values. This demonstrates that the model is robust. It also demonstrates stable predictive performance when handling high-volume stations.

(3) Figure 6(c) shows the passenger flow prediction performance at Station C. Station C is a residential area station. It is influenced by commuter travel patterns. Its passenger flow distribution exhibits a clear single-evening peak. The overall passenger flow volume is at a medium-to-low level throughout the day. Nighttime and weekend passenger flow activity is notably lower. AGHA-Net can effectively capture the stable passenger flow patterns at residential stations. These patterns are shaped by fixed commuter behavior. In scenarios with gentle fluctuations in passenger flow, the model maintains high prediction accuracy. This further verifies the model's generalization ability to diverse passenger flow patterns.

Based on the experimental results across the three types of stations, AGHA-Net demonstrates strong predictive performance at commuter, commercial, and residential area stations. It can effectively handle variations in passenger flow across different station types. The prediction performance variation across the three station types correlates with their intrinsic temporal volatility, confirming that the model's adaptive mechanisms effectively accommodate diverse passenger flow characteristics without requiring site-specific reconfiguration.

5.5.2 Parameter sensitivity analysis

To clarify the influence patterns of two key spatial hyperparameters in the model, this section conducts sensitivity analyses. It focuses on two hyperparameters. They are the node embedding dimension (embed_dim) and the spatial convolution order (cheb_order). embed_dim is part of the AdaGCN adaptive convolution module. It determines the expressive power of node embeddings. This, in turn, affects the quality of learning from the dynamic adjacency matrix. cheb_order is part of the spatial graph convolution in the feature processing layer. It controls the order of the Chebyshev polynomial based on the static topology. This determines the receptive field of physical neighbor aggregation. These two hyperparameters influence the model's spatial modeling capability. They do so from two perspectives. One is dynamic data-driven. The other is a static topology prior.

We set embed_dim to [4, 6, 8, 10] and cheb_order to [1, 2, 3, 4]. These settings cover comprehensive scenarios, including insufficient dimensionality to potential overfitting and limited receptive fields to possible high-order noise. Figure 7(a) and 7(b) depict the error variation trends of the two parameters at the 15-minute prediction granularity.

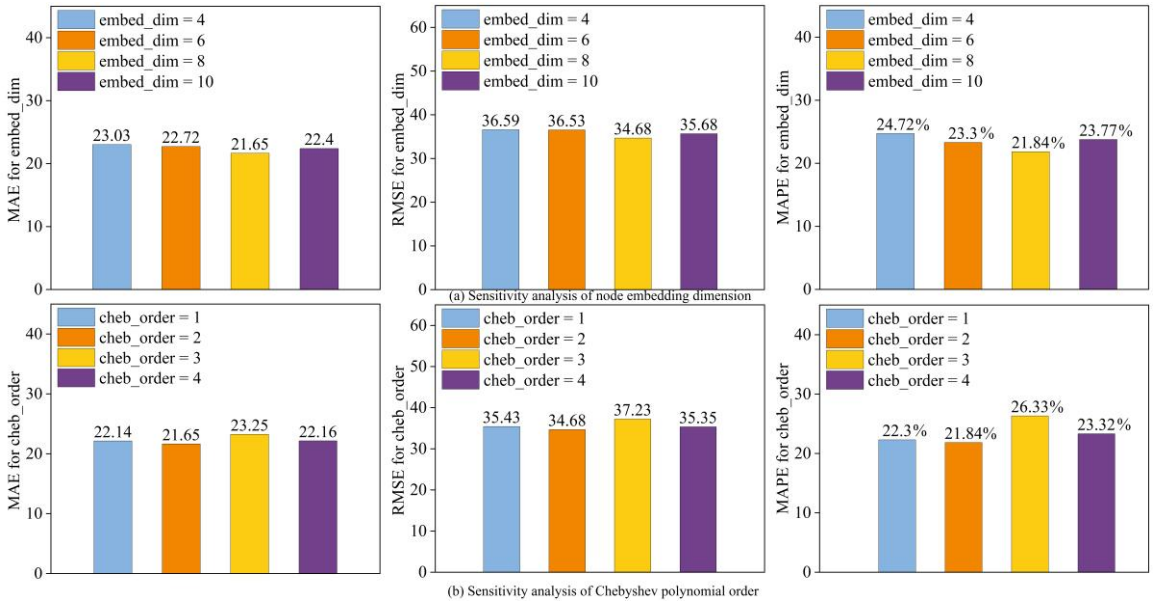


Fig.7 Sensitivity Analysis Result Chart

As shown in Figure 7(a), the model error decreases as embed_dim increases from 4 to 8. This is because a larger embedding dimension better characterizes nonlinear dynamic correlations among stations, enabling dynamic adjacency matrices to precisely learn time-varying spatial dependencies. When embed_dim is further increased to 10, the model has excessive parameters, leading to overfitting with insufficient training data and a slight increase in error. Hence, embed_dim=8 reaches the optimal balance between feature representation capability and model generalization performance.

As shown in Figure 7(b), MAE, RMSE, and MAPE all follow a U-shaped trend as cheb_order increases. At cheb_order=1, graph convolution only captures first-order neighbor information. Its narrow spatial receptive field leads to high prediction errors. When the order reaches 2, the model broadens spatial perception without redundant noise, and all metrics achieve the best results. If the order rises to 3 or 4, high-order neighborhood information causes over-smoothing and excessive noise. It distorts spatial dependency modeling and obviously weakens prediction performance. Therefore, cheb_order=2 realizes the optimal balance between expanding receptive field and restraining over-smoothing.

Overall, AGHA-Net maintains strong robustness within reasonable parameter ranges. Setting embed_dim=8 equips EmbGCN with adequate node-embedding capability. It can accurately capture nonlinear dynamic correlations between stations. Setting cheb_order=2 enables spatial graph convolution to aggregate physical-neighbor information with a proper receptive field. It also effectively avoids the over-smoothing problem. The joint effect of the two parameters provides a solid spatial modeling foundation for stable, high-accuracy prediction.

5.5.3 Ablation experiment

To evaluate the independent contribution of each core component of AGHA-Net and its function in long-term prediction, we perform fine-grained component ablation experiments. We adopt 5-minute and 20-minute prediction steps, in addition to the main 15-minute setting described in Section 5.1. We systematically remove core components and observe changes in error during a 5 to 30-minute multi-step prediction. This reveals module performance and long-range dependency features under different prediction durations. All model settings, training schemes, and data remain consistent without special notes. We establish the following ablation variants:

(1) AGHA-Net (Full): A comprehensive model encompassing all its components.

(2) w/o AdaGCN: Remove the adaptive graph convolution module and replace it with static graph convolution.

(3) w/o ST-GRU: Remove the spatio-temporal gated recurrent unit and use the standard GRU for temporal modeling.

(4) w/o SSAM: Remove the sequential self-attention module and cancel the global long-term temporal dependency modeling.

(5) w/o TABL: Remove the bilinear temporal attention module, retain only the sequential self-attention for temporal modeling, and eliminate the bidimensional compression and bilinear attention mechanism.

Figures 8 and 9 present the error performance of all variants across diverse prediction granularities and calculate their performance degradation relative to the complete model.

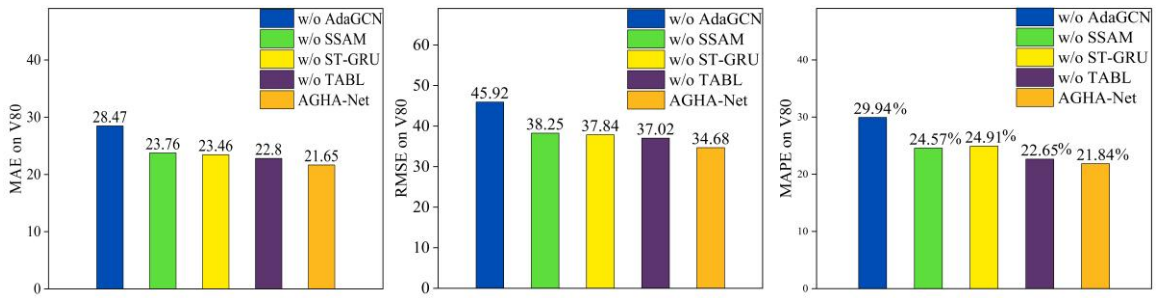


Fig.8 Ablation experiment results of one-step prediction

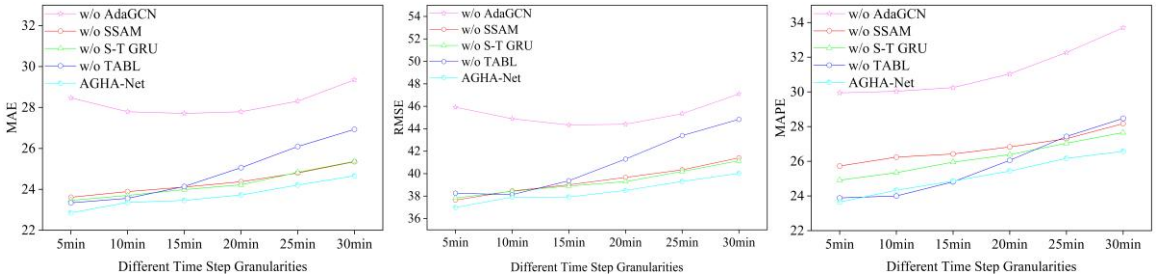


Fig.9 Ablation experiment results at various time granularities

The results clearly demonstrate that the complete AGHA-Net achieves the minimum error at all prediction steps. Removing any core component leads to steady deterioration of prediction performance. Each component plays an indispensable role, especially in long-term prediction and passenger flow peak scenarios:

(1) The removal of AdaGCN causes a sharp drop in model performance. Spatial modeling forms the basic framework of the entire model. Figure 8 shows that the variant without AdaGCN results in an MAE increase of 6.82, an MAPE increase of 8.10 percentage points, and an RMSE increase of 11.24 in single-step prediction. All three metrics degrade by over 35%, ranking first among all variants. Figure 9 further proves that such performance loss is global and irreversible. Its MAE reaches 28.47 at the 5-minute prediction step and fluctuates slightly between 27.71 and 29.35 in subsequent steps. Its RMSE remains high, within 44.34 to 47.10, and its MAPE rises to 33.70% at the 30-minute step. This degradation indicates that the model reverts to static graph convolution. It loses the ability to learn node embedding-based adaptive spatial correlations. No temporal module can compensate for the structural loss of spatial features. Therefore, dynamic spatial dependency modeling is essential to guarantee prediction accuracy.

(2) Removing TABL brings obvious sensitivity to prediction steps, and this module is critical for restraining long-term error accumulation. Figure 9 shows a clear, gradually increasing error trend. In the 5-minute prediction experiment, the model without TABL performs similarly to the complete model, with MAE rising by only 0.49 and MAPE increasing by 0.24 percentage points. Nevertheless, performance degradation accelerates as the number of prediction steps increases. The MAE rises by 0.53 at 15 minutes and 1.45 at 20 minutes, and reaches 2.32 at 30 minutes. Meanwhile, RMSE increases from 38.26 to 44.84. This gradual degradation trend shows that TABL contributes little to short-term local feature extraction. It mainly constructs bilinear temporal attention via channel and spatial compression to capture long-term periodic patterns and internal temporal correlations. For short prediction steps, local temporal modeling in ST-GRU can partially compensate for its absence. With longer prediction steps, the lack of multi-scale periodic information accelerates error accumulation and weakens the model’s ability to track long-term trends in passenger flow.

(3) SSAM and ST-GRU show similar degradation ranges but different patterns of change. They undertake global dependency modeling and local feature fusion, respectively. Figure 8 shows that removing SSAM increases single-step MAE by 2.11 and MAPE by 2.73 percentage points. Removing ST-GRU leads to MAE growth of 1.81 and MAPE growth of 3.07 percentage points. Both modules suffer a performance decline at the same level. Figure 9 further reflects their different degradation rules. The RMSE of the model without SSAM rises steadily from 37.64 at 5 minutes to 41.41 at 30 minutes. Its error accumulates approximately linearly. This shows that missing global long-range dependencies steadily degrade medium- and long-term prediction performance. The model's error trend without ST-GRU is much milder. Its MAPE only rises by 1.09 percentage points at the 15-minute step. It means that the lack of a spatio-temporal coupling mechanism weakens the short-term fluctuation-capturing ability. This defect can be partially offset by temporal modeling of high-level hybrid attention.

In summary, AdaGCN establishes the adaptive spatial correlation framework. ST-GRU realizes local spatio-temporal synchronous updating. SSAM captures global long-range temporal dependencies. TABL achieves long-term trend modeling via dual-dimensional compression and effectively restrains error accumulation. The four modules exhibit distinct functional divisions: AdaGCN provides structural spatial priors, ST-GRU couples local dynamics with temporal transitions, SSAM captures global periodic regularities, and TABL specializes in abrupt fluctuation responses—their parallel integration ensures robust performance across varying traffic conditions.

5.6. Effectiveness analysis of TABL module

To further verify the effectiveness of TABL, this section constructs a comparative model. TABL stands for Temporal Attention with Bilinear module. The module captures long-range temporal dependencies. It also improves multi-step prediction stability. The comparative model is called STAGCN. STAGCN stands for Spatio-Temporal Attention Graph Convolutional Network. STAGCN uses AGHA-Net as its backbone. It replaces the bilinear temporal attention module with a traditional one. The traditional module is called SimplestTATT. The rest of the structure remains unchanged. This includes AdaGCN, ST-GRU, SSAM, and the feature processing layer. The core difference between the two attention modules is as follows. TABL performs compression via convolution along both the feature and node dimensions. It then generates a complete time-time attention matrix. This allows each output time step to adaptively fuse global historical information. The traditional temporal attention module compresses each time step into a scalar. It uses global average pooling. After linear mapping and softmax normalization, it produces independent scaling weights for each time step. There is no information interaction between time steps. As shown in Table 3 and Figure 10, AGHA-Net significantly outperforms STAGCN. This holds across all time granularities.

Table 3 Comparison between the bilinear temporal attention module and the traditional temporal attention module

Model	Time granularity	Evaluation metric		
		MAE	RMSE	MAPE(%)
AGHA-Net	15min	**21.65**	**34.68**	**21.84%**
	30min	22.37	35.92	23.07%
	45min	22.96	36.91	24.43%
STAGCN	15min	23.49	37.95	24.28%
	30min	24.76	40.25	26.29%
	45min	25.02	40.65	26.81%

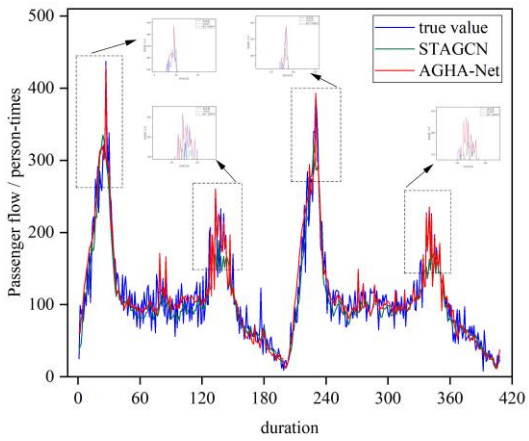


Fig.10 Comparison of Prediction Results

As shown in Table 3, TABL significantly reduces prediction errors compared to the traditional temporal attention module. Taking the 30-minute prediction as an example, AGHA-Net reduces the MAE by 2.39, the

RMSE by 4.33, and the MAPE by 3.32 percentage points. For the 45-minute long-term prediction, the three metrics are reduced by 2.06, 3.74, and 2.38 percentage points, respectively. Looking at the error growth trend, for STAGCN, MAE increases from 23.49 (15 min) to 25.02 (45 min). This is an increase of 1.53. RMSE increases from 37.95 to 40.65, an increase of 2.70. For AGHA-Net, MAE only increases from 21.65 to 22.96, an increase of 1.31. RMSE increases from 34.68 to 36.91, an increase of 2.23. This indicates that TABL effectively suppresses error accumulation in long-term prediction.

The results in Figure 10 further show that TABL can accurately identify long-range temporal dependencies. It does so in passenger flow sequences. It closely follows the true values at both peaks and troughs of passenger flow. In contrast, the traditional temporal attention module shows obvious lag and amplitude deviation. This occurs when facing rapidly changing time periods. The above results fully verify the effectiveness of the dual-view compression and bilinear interaction mechanism. This mechanism excels in improving passenger flow prediction accuracy. It also excels in improving long-term prediction stability.

6. Conclusion

This paper proposes AGHA-Net. It is an adaptive graph convolution and hybrid attention prediction model. It is designed for short-term inbound passenger flow in urban rail transit. The model addresses three issues. These issues exist in short-term inbound passenger flow prediction. They have insufficient dynamic spatial topology. They also have inadequate temporal dependency modeling. And they are inefficient in spatiotemporal feature fusion. The core components of the model are adaptive graph convolution, spatiotemporal gating units, and hybrid attention. The model achieves joint representation of dynamic spatial dependencies, local spatiotemporal features, and long-term temporal trends. This joint representation is achieved within a unified framework.

The model uses AdaGCN adaptive graph convolution. It adaptively generates time-varying adjacency relationships. This effectively captures nonlinear dynamic spatial correlations. These correlations exist between stations. The model employs an ST-GRU unit. This unit has embedded graph convolution. It synchronously updates temporal iteration and spatial feature aggregation. This helps efficiently extract short-term fluctuations. It also helps extract local spatiotemporal dependencies. The hybrid attention layer consists of two parallel parts. They are sequential self-attention and bilinear temporal attention. The former introduces positional encoding and one-dimensional convolution. It enhances global non-local temporal dependencies. The latter uses dual-view compression. This compression operates on feature and node dimensions. It also uses a bilinear interaction mechanism. It effectively captures long-range temporal correlations. It also significantly reduces computational cost. Comparative experiments show that the bilinear temporal attention module plays a key role. It helps suppress the accumulation of long-term prediction error. After feature alignment and nonlinear mapping, the model directly outputs the prediction. The prediction is for the next time step's passenger flow.

We conduct experiments on the real-world passenger flow dataset of Hangzhou Metro. The experimental results show that AGHA-Net outperforms baseline models. These models include LSTM, GCN, and ASTGCN. The comparison is conducted at 15-minute, 30-minute, and 45-minute time granularities. In typical scenarios such as commuter, commercial, and residential area stations, AGHA-Net maintains high accuracy. It also shows strong robustness. Ablation experiments verify the key role of each core module. These modules improve prediction performance. This proves that the model can effectively adapt to the dynamic evolution of passenger flow. It also adapts to complex spatiotemporal patterns.

In real-world operational scenarios, passenger flow is affected not only by spatiotemporal factors. It is also influenced by external factors such as weather and holidays. Future research will incorporate multimodal data.

These include meteorological data, holiday information, and special events. This will enrich the model's input information. It will also improve prediction accuracy and robustness. Meanwhile, we will continue to make improvements in areas such as adaptive graph learning, causal interpretability, and cross-domain transfer. This will support intelligent operation and refined decision-making in rail transit.

Acknowledgements This work was supported by the National Key Research and Development Program of China (Grant No. 2023YFB4302100), the Major Science and Technology Research and Development Project of Jiangxi Province (Grant No. 20232ACE01011), and the Jiangxi Provincial Key Laboratory of Maglev Rail Transit Equipment at Jiangxi University of Science and Technology (Grant No. 2020SSY050).

Author contributions Qingshuo Sha wrote the main manuscript text. Lu Zeng provided guidance on ideas for writing the manuscript and the methods involved in the manuscript. Tao GAO contributed significantly to the analysis and manuscript preparation. Qinzhen Fang helped perform the analysis with constructive discussions. All authors reviewed the manuscript.

Data availability Data will be made availableon on request.

Declarations

Competing interests The authors declare no competing interests.

References

An, Y., Li, Z., Li, X., Liu, W., Yang, X., Sun, H., Chen, M., Zheng, Y., Gong, Y.: Spatio-Temporal Multivariate Probabilistic Modeling for Traffic Prediction. IEEE Transactions on Knowledge and Data Engineering 37, 1-15 (2025).

Apronti, D., Ksaibati, K., Gerow, K., Hepner, J.J.: Estimating traffic volume on Wyoming low volume roads using linear and logistic regression methods(Article). Journal of Traffic and Transportation Engineering (English Edition) 3, 493-506 (2016).

Ata, K. I. M., Hassan, M. K., Ismaeel, A. G., Al-Haddad, S. A. R., Alquthami, T., Alani, S.: A multi-Layer CNN-GRUSKIP model based on transformer for spatial-TEMPORAL traffic flow prediction. Ain Shams Engineering Journal 15 (2024).

Bai, J., Zhu, J., Song, Y., et al.: A3T-GCN: Attention temporal graph convolutional network for traffic forecasting. ISPRS Int. J. Geo-Inf. 10(7), 485 (2021).

Bao, Y. X., Liu, J. L., Shen, Q. Q., Cao, Y., Ding, W. P., Shi, Q.: PKET-GCN: Prior knowledge enhanced time-varying graph convolution network for traffic flow prediction. Information Sciences 634, 359-381 (2023).

Benarmas, R. B., Bey, K. B.: A deep learning-based framework for road traffic prediction. Journal of Supercomputing 80, 6891-6916 (2024).

Bogaerts, T., Masegosa, A. D., Angarita-Zapata, J. S., Onieva, E., Hellinckx, P.: A graph CNN-LSTM neural network for short and long-term traffic forecasting based on trajectory data. Transportation Research: Part C 112, 62-77 (2020)

Cai, J. B., Wang, C. H., Hu, K.: LCDFormer: Long-term correlations dual-graph transformer for traffic forecasting. Expert Systems with Applications 249 (2024).

Cai, S., Hu, J., Wei, M., Zhang, X.: Semantic-aware adaptive traffic flow prediction driven by real-time trajectories. Information Sciences 728, 122819 (2026).

Chang, Z., Liu, C., Jia, J.: STA-GCN: Spatial-temporal self-attention graph convolutional networks for traffic-flow prediction. Appl. Sci. 13(11), 6796 (2023).

Chen, J., Ye, H. C., Ying, Z., Sun, Y. T., Xu, W. Q.: Dynamic trend fusion module for traffic flow prediction. Applied Soft Computing 174 (2025).

Dai, R., Xu, S., Gu, Q.: Hybrid Spatio-Temporal Graph Convolutional Network: Improving Traffic Prediction with Navigation Data. Alibaba Group (2020).

Feng, A., Tassiulas, L.: Adaptive Graph Spatial-Temporal Transformer Network for Traffic Forecasting. arXiv (2022).

Fu, H. Y., Ma, H. D., Liu, Y. X., Lu, D. W.: A vehicle classification system based on hierarchical multi-SVMs in crowded traffic scenes. Neurocomputing 211, 182-190 (2016).

Gao, C., Liu, H., Huang, J. J., Wang, Z., Li, X. H., Li, X. L.: Regularized Spatial-Temporal Graph Convolutional Networks for Metro Passenger Flow Prediction. IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS 25, 11241-11255 (2024).

Guo, G., Zhang, T.: A residual spatio-temporal architecture for travel demand forecasting (vol 115, 102639, 2020). TRANSPORTATION RESEARCH PART C-EMERGING TECHNOLOGIES 120, 102728 (2020).

Guo, S., Lin, Y., Wan, H., Li, X., Cong, G.: Learning Dynamics and Heterogeneity of Spatial-Temporal Graph Data for Traffic Forecasting. IEEE Transactions on Knowledge & Data Engineering 34, 5415-5428 (2022).

Hamed, M. M., Al-Masaeid, H. R.: Short-term prediction of traffic volume in urban arterials: Proceedings of the American Society of Civil Engineers. Journal of Transportation Engineering 121, 249 (1995).

Huang, X., Wang, J., Lan, Y., Jiang, C., Yuan, X.: MD-GCN: A Multi-Scale Temporal Dual Graph Convolution Network for Traffic Flow Prediction. Sensors (Basel, Switzerland) 23, 841 (2023).

Jiang, J., Han, C., Zhao, W. X., Wang, J.: PDFormer: Propagation Delay-Aware Dynamic Long-Range Transformer for Traffic Flow Prediction. arXiv (2023).

Li, L., Qin, L., Qu, X., Zhang, J., Ran, B.: Day-ahead traffic flow forecasting based on a deep belief network optimized by the multi-objective particle swarm algorithm. Knowledge-Based Systems 172, 1-14 (2019).

Lin, Z., Feng, J., Lu, Z., Li, Y., Jin, D.: DeepSTN+: Context-Aware Spatial-Temporal Neural Network for Crowd Flow Prediction in Metropolis. (2019).

Lint, J. W. C. van, Hoogendoorn, S. P., van Zuylen, H. J.: Freeway Travel Time Prediction with State-Space Neural Networks: Modeling State-Space Dynamics with Recurrent Neural Networks. Transportation Research Record 1811, 30-39 (2002).

Liu, L., Chen, J., Wu, H., Zhen, J., Li, G., Lin, L.: Physical-Virtual Collaboration Modeling for Intra- and Inter-Station Metro Ridership Prediction. IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS 23, 3377-3391 (2022).

Liu, Y. T., Feng, T., Rasouli, S., Wong, M. L.: ST-DAGCN: A spatiotemporal dual adaptive graph convolutional network model for traffic prediction. Neurocomputing 601 (2024).

Ma, X. L., Yu, H. Y., Wang, Y. P., Wang, Y. H.: Large-Scale Transportation Network Congestion Evolution Prediction Using Deep Learning Theory. Plos One 10 (2015).

Mao, M., Han, Y., Wang, B.: Hybrid-relation dynamic hypergraph attention network for traffic flow prediction. Knowledge-Based Systems 331, 114766 (2026).

Okutani, I., Stephanedes, Y. J.: DYNAMIC PREDICTION OF TRAFFIC VOLUME THROUGH KALMAN FILTERING THEORY. TRANSPORTATION RESEARCH PART B-METHODOLOGICAL 18, 1-11 (1984).

Peng, C., Cheng, Y., Li, A.: Simplicity meets power: robust traffic flow prediction with ST-ConvLSTMNet model. Applied Intelligence 56, 1-16 (2026).

Shirzadeh, M., Amirkhani, A., Tork, N., Taghavifar, H.: Trajectory tracking of a quadrotor using a robust adaptive type-2 fuzzy neural controller optimized by cuckoo algorithm. Isa Transactions 114, 171-190 (2021).

Singh, V., Sahana, S. K., Bhattacharjee, V.: A novel CNN-GRU-LSTM based deep learning model for accurate traffic prediction. Discover Computing 28 (2025).

Song, L., Ren, Q.: Spatial-Temporal Adaptive Graph Convolution Network with Channel Attention. Heilongjiang University (2024).

Sun, S. L., Zhang, C. S., Yu, G. Q.: A Bayesian network approach to traffic flow forecasting. IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS 7, 124-132 (2006).

Teng, G., Wu, H., He, A., Chen, Y., Zhao, M.: A sparse dynamic graph transformer for traffic flow prediction. Neurocomputing 659, 131765 (2026).

Wang, P., Feng, L. X., Zhu, Y. J., Wu, H. P.: Hybrid spatial-temporal graph neural network for traffic forecasting. Information Fusion 118 (2025).

Wang, R., Yue, J., Guo, Q. W., Zhong, P. F., Jia, C. L.: Traffic dynamics HD: A federated learning traffic flow prediction model inspired by high-definition image generation techniques. Information Sciences 745 (2026).

Wang, X., Chen, J., Chen, F., Jin, B., Zou, F., Liao, L., Zeng, R.: FedCTPM: A federated cross-regional collaboration learning approach for traffic flow prediction. Expert Systems with Applications 305, 130779 (2026).

Williams, B. M., Hoel, L. A.: Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: Theoretical basis and empirical results: Proceedings of the American Society of Civil Engineers. Journal of Transportation Engineering 129, 664-672 (2003).

Wu, C.-H., Ho, J.-M., Lee, D. T.: Travel-time prediction with support vector regression. IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS 5, 276-281 (2004).

Wu, Y., Tan, H.: Short-term traffic flow forecasting with spatial-temporal correlation in a hybrid deep learning framework. (2016).

Xing, X., Wang, B., Ning, X., Wang, G., Tiwari, P.: Short-term OD flow prediction for urban rail transit control: A multi-graph spatiotemporal fusion approach. Information Fusion 118 (2025).

Yan, X., Gan, X. H., Tang, J. J., Zhang, D. P., Wang, R.: ProSTformer: Progressive Space-Time Self-Attention Model for Short-Term Traffic Flow Forecasting. IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS 25, 10802-10816 (2024).

Yang, H., Jiang, C., Song, Y., Fan, W., Deng, Z., Bai, X.: TARGCN: temporal attention recurrent graph convolutional neural network for traffic prediction. Complex & Intelligent Systems 10, 8179-8196 (2024).

Yang, X., Xue, Q. C., Yang, X. X., Yin, H. D., Qu, Y. C., Li, X., Wu, J. J.: A novel prediction model for the inbound passenger flow of urban rail transit. Information Sciences 566, 347-363 (2021).

Yao, C., Zhou, W.: Attention Graph Neural Controlled Differential Equations Approach for Traffic Flow Forecasting. Anhui Normal University (2025).

Ye, J. H., Xue, S. J., Jiang, A. W.: Attention-based spatio-temporal graph convolutional network considering external factors for multi-step traffic flow prediction. Digital Communications and Networks 8, 343-350 (2022)

Yu, B., Yin, H., Zhu, Z.: Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. arXiv:1709.04875 (2017).

Yu, J. J., Fang, X., Zhang, S., Ma, Y.: CLEAR: Spatial-Temporal Traffic Data Representation Learning for Traffic Prediction. IEEE Transactions on Knowledge and Data Engineering 37, 1-16 (2025).

Zhang, L., Liu, Q., Yang, W., Wei, N., Dong, D.: An Improved K-nearest Neighbor Model for Short-term Traffic Flow Prediction. Procedia - Social and Behavioral Sciences 96, 653-662 (2013).

Zhang, Q., Shang, Q., Zhou, Q., Yang, M., Xu, B., Yang, Z.: DGSSformer : Dynamically global-aware spatiotemporal synchronous transformer for traffic prediction. Expert Systems with Applications 307, 131063 (2026).

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 TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS 21, 3848-3858 (2020).