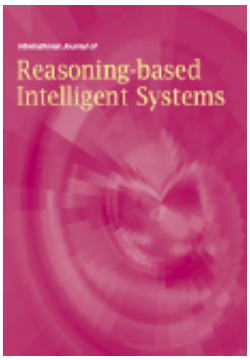


International Journal of Reasoning-based Intelligent Systems

ISSN online: 1755-0564 - ISSN print: 1755-0556
https://www.inderscience.com/ijris

Transformer-GNN hybrid architecture for optimising real-time traffic forecasting on highways

Hua Cheng, Yupeng Cao, Weiping Li

DOI: 10.1504/IJRIS.2026.10076378

Article History:

Received:	16 November 2025
Last revised:	23 December 2025
Accepted:	23 December 2025
Published online:	10 March 2026

Transformer-GNN hybrid architecture for optimising real-time traffic forecasting on highways

Hua Cheng* and Yupeng Cao

Ji'andong Management Center,
Jiangxi Communications Investment Group Co., LTD.,
Ji'an, 343700, China
Email: sdjtzyxylwp@163.com
Email: 631226838@qq.com
*Corresponding author

Weiping Li

Shandong Transport Vocational College,
Weifang, 261206, China
Email: weifangyuli@126.com

Abstract: Facing the challenge of worsening highway traffic congestion, precise real-time forecasting is crucial for intelligent traffic management. However, traditional models struggle to effectively capture the complex spatio-temporal dependencies and dynamic propagation delays inherent in traffic data. To address this, this paper proposes a hybrid architecture that integrates graph neural networks with transformers. Through a dynamic graph attention mechanism and a delay-aware module, it significantly enhances the modelling capabilities for long-range spatial correlations and temporal propagation effects. Experiments on public datasets such as performance measurement system 04 and performance measurement system 08 demonstrate that the proposed model reduces the mean absolute error by 6.2%–9.2% compared to existing state-of-the-art methods within the 15–60 minute prediction window, with particularly notable performance improvements during peak congestion periods. The framework presented here has the potential to provide a more reliable technical pathway for traffic state prediction, holding significant practical application value.

Keywords: traffic flow prediction; graph neural networks; GNNs; transformers; intelligent transportation systems.

Reference to this paper should be made as follows: Cheng, H., Cao, Y. and Li, W. (2026) 'Transformer-GNN hybrid architecture for optimising real-time traffic forecasting on highways', *Int. J. Reasoning-based Intelligent Systems*, Vol. 18, No. 9, pp.38–50.

Biographical notes: Hua Cheng is the General Manager at the Ji'andong Management Center, Jiangxi Communications Investment Group Co., Ltd., China. He obtained his Master's degree from the Central Party School of the Communist Party of China (2014) and the title of Intermediate Registered Safety Engineer (2021). He has published one paper indexed by EI. His research interests include the operation and management of expressways.

Yupeng Cao is the Manager of the Engineering Maintenance Department at the Ji'andong Management Center, Jiangxi Communications Investment Group Co., Ltd., China. He obtained his Bachelor's in Engineering Management from Nanchang Hangkong University (2013) and then a Master's in Bridge and Tunnel Engineering from Chongqing Jiaotong University (2016), China. He has published one paper indexed by EI. His research interest is the maintenance and management of expressways.

Weiping Li is a teacher at Shandong Transport Vocational College, China. He obtained his Bachelor's degree from China University of Petroleum (East China) (2011) and then a Master's degree from Northeast Forestry University (2013). He has published two papers indexed by EI. His research interests lie in highway operation safety and information technology.

1 Introduction

With the further advancement of urbanisation and the constant increase in the number of motor vehicles, the highway system, which is the 'lifeblood' of urban traffic, is confronted with increasingly severe traffic congestion, accident costs, and deaths (Hommes and Boelens, 2018). World health organisation statistics indicate that more than 1.35 million people die each year from traffic accidents, and most of these accidents are closely related to the sudden changes in traffic flow and the propagation of congestion (Ayala, 2024). Accurate real-time traffic forecast on highways can not only provide travellers with more reasonable path suggestions to arrive at their destinations on time but also provide decision support to traffic management staff. Thus, traffic resources can be allocated reasonably, and road usage efficiency can be improved. However, traffic data has the following challenges. The accurate modelling of dynamic propagation delay is particularly critical for traffic management systems, as it directly influences the precise timing of congestion alerts and the effective allocation of traffic resources. This capability enables more proactive and responsive control strategies, thereby enhancing overall traffic flow efficiency and safety. The complex spatiotemporal dependencies, dynamic propagation delays, and long-range spatial correlations make it challenging for traditional forecasting models to solve traffic forecasting problems (Zhang et al., 2025). These challenges are mainly reflected in the following aspects: The traffic conditions spread along the road network and have obvious spatio-temporal delay effects. That is, the changes in the upstream road sections take a certain amount of time to affect the downstream road sections. The interaction between different road segments may span long spatial distances, but traditional methods can only aggregate information from local neighbourhoods. The spatio-temporal patterns of traffic data have multi-scale characteristics. The short-term patterns are controlled by signals. The medium-term patterns are dominated by travel modes. The long-term patterns are controlled by periodic regularities.

In recent years, with the rapid development of deep learning technology, graph neural networks (GNNs) and transformer methods have many advantages in traffic forecasting (Duan and Hu, 2025). GNNs can capture the spatial dependencies between roads by modelling transportation networks as graph structures. For example, compared with the eag-gcn-t model, eag-gcn-t model uses enhanced graph convolutional networks and transformers to model the spatial relationships between vehicles based on relative speed and distance. The trajectory prediction error in the multi-vehicle interaction scenario at the intersection is greatly reduced (Wang and Zhang, 2025). To address the turn rules and directional information in the urban road network, turn graph convolutional network (TurnGCN) was proposed (Wang and Zhang, 2025). It models the road network as a heterogeneous graph where edges represent turn relationships between road segments. It overcomes the limitations of traditional graph convolutional networks in modelling traffic network characteristics (Umair et al., 2025). However, most of these methods are based on fixed graph structures defined in advance, and they cannot fully reflect the time-varying dynamic spatial correlations. Especially on the highway scenario, the spatial influence pattern of traffic flow is quite different in different time periods (Wang et al., 2024).

In addition, transformer model has obvious advantages in mining long-term dependencies in time series data through powerful self-attention mechanism. The astnn model expresses transportation networks in form of compact two-dimensional images and applies attention-based spatio-temporal memory block and attention-based spatio-temporal focus block to solve temporal correlation mining and spatial sparse feature extraction problems respectively, which successfully solve the problem of road level sparse traffic flow prediction (Al-Tameemi et al., 2025). The opencity intelligent traffic large model was released by hku combined the merits of transformers and GNNs. It could effectively capture complex spatio-temporal dependencies and achieved zero-shot prediction, provided new ideas for cross-city traffic forecasting (Holail et al., 2025). However, transformers based models still have certain limitations in traffic prediction, such as high computational complexity and insufficient consideration of traffic propagation patterns (Wang et al., 2017).

At present, the fusion architecture of GNNs and Transformers is a new research hotspot in traffic forecasting research. The multivariate time series dynamic graph neural network (MTDGNN) model adapts to the real traffic information and outputs dynamic graphs (Lai et al., 2025). It constructs multi-level spatial neighbourhoods and temporal receptive fields by combining graph convolutional and temporal convolutional modules and shows better robustness in complex traffic road. The MGNformer model captures spatial interaction features from multiple scales by combining multi-scale hypergraph neural modules and graph attention modules (Jie-Yang and Liu, 2024). Meanwhile, it uses a deformable self-attention mechanism to capture long-range dependencies in the temporal dimension. Notably, the propagation delay-aware transformer (PDFormer) model has a specially designed propagation-delay-aware dynamic long-range Transformer (Wei et al., 2024). It captures dynamic spatial dependencies in the trajectory data through spatial self-attention modules and introduces a traffic-delay-aware feature transformation module to model temporal delays in the propagation of spatial information explicitly. This makes the spatial information propagation more consistent with the physical characteristics of traffic propagation (Zong et al., 2024). These studies provide references for jointly modelling on spatiotemporal dependencies. However, most of the hybrid models are still limited to simple serial or parallel structures. There is still a lack of deep integration between the two techniques, and the explicit consideration of the physical characteristics of traffic propagation is not enough.

In terms of highway traffic forecasting, compared with the existing research, there are still many challenging issues

to be solved urgently. To address these gaps, this paper proposes the spatio-temporal graph former (STG-Former). While PDFormer pioneered the explicit modelling of propagation delays, it primarily relies on a static graph structure and a standard Transformer for temporal processing. STG-Former advances this paradigm by introducing:

1 a dynamic graph attention mechanism that captures time-varying spatial dependencies

2 a more granular delay-aware feature propagation module integrated within the graph learning process

3 a multi-scale temporal attention mechanism to replace the standard transformer, enabling adaptive focus on patterns across different temporal granularities.

This deep, synergistic integration aims to more holistically capture the complex, dynamic, and multi-scale nature of traffic propagation. First, the propagation of traffic condition has typical spatio-temporal delay characteristic. However, most of the models ignore the explicit modelling of this important characteristic, leading to low accuracy in traffic wave propagation process. Second, the traditional graph convolution operation usually focuses on aggregating information in local neighbourhood. It is hard to model long-range relationship between two road segments which are geographically distant but semantically highly correlated in the network. Meanwhile, the spatio-temporal characteristic exists in traffic data with multi-scale characteristic. The existing method usually cannot adaptively fuse the feature from different temporal scale. Therefore, the expressive ability and prediction accuracy of the model are restricted. More seriously, most of the models treat traffic forecasting as a data-driven task. They ignore the explicit physical process of traffic propagation. Therefore, the interpretability and generalisation ability of the model are restricted (Qiu et al., 2024).

2 Related research

2.1 Fundamental principles and applications of GNNs in traffic forecasting

GNNs show promising performance in traffic forecasting by treating transportation networks as graph structures. The essence of GCNs can be explained by regarding the sensors or road segments in the transportation network as nodes in graphs and the inter-segment connections or spatial adjacency as edges, which is able to capture spatial dependencies of transportation networks. Its fundamental formula is as follows:

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

where $\tilde{A} = A + I$ denotes the adjacency matrix with self-connections added, $\tilde{D}$ is the corresponding degree matrix, $H^{(l)}$ represents the node features of layer l, $W^{(l)}$ denotes the learnable parameters, and σ is the activation function. While this foundational architecture effectively captures spatial dependencies, it relies on predefined static graph structures, making it difficult to adapt to the dynamically changing spatial correlations in transportation systems. On highways, traffic conditions can shift rapidly due to unforeseen incidents or periodic peak-hour demands. Static graph structures, which rely on fixed topological representations, lack the adaptability to capture these real-time variations, thereby limiting their effectiveness in dynamically evolving traffic environments. To model dynamic graphs, dynamic graph convolutional networks are proposed. Among them, graph attention networks (GAN) learn the importance of connections between nodes from other nodes via attention mechanism (Pan et al., 2019). The core computation of GAT can be formulated as:

$$e_{ij} = a^T\left[Wh_i \mid Wh_j\right] \tag{2}$$

$$\alpha_{ij} = \frac{\exp\left(Leaky\,\mathrm{Re}\,LU\left(e_{ij}\right)\right)}{\sum_{k\in\mathcal{N}(i)}\exp\left(Leaky\,\mathrm{Re}\,LU\left(e_{ik}\right)\right)} \tag{3}$$

$$h_{i'} = \sigma\left(\sum_{j\in\mathcal{N}(i)}\alpha_{ij}Wh_j\right) \tag{4}$$

where e_{ij} denotes the attention score between nodes i and j, α_{ij} represents the normalised attention weight, and $\mathcal{N}(i)$ indicates the set of neighbours for node i. This approach dynamically adjusts spatial dependencies based on real-time traffic conditions, better aligning with the influence patterns observed in actual traffic systems that evolve over time and under varying conditions.

2.2 The theoretical foundations of transformer models in time series forecasting

Transformer models have shown their capacity of modelling long-term dependencies in time series via self-attention mechanism (Al-Thani et al., 2024). Compared with traditional recurrent models, transformer models can directly make dependencies between any two time steps, which alleviates the vanishing gradient problem (Petridis, 2024). The core computation of self-attention mechanism can be formulated as:

$$Attention(Q, K, V) = soft\max\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{5}$$

$$MultiHead(Q, K, V) = Concat\left(head_1, ..., head_h\right)W^O \tag{6}$$

$$head_i = Attention\left(QW_i^Q, KW_i^K, VW_i^V\right) \tag{7}$$

However, the standard transformer faces two major challenges in traffic forecasting: first, its computational complexity grows quadratically with sequence length, limiting its application in long-sequence predictions; second, it lacks specialised consideration for the spatiotemporal coupling characteristics of traffic data. The spatio-temporal coupling in traffic systems is ubiquitous

because road conditions are both a cause and an effect of variations across time and space. This bidirectional relationship means that temporal changes influence spatial traffic states, while spatial configurations, in turn, affect how traffic evolves over time. To address these issues, recent research has proposed various improvement strategies, including sparse attention mechanisms, local window attention, and domain-knowledge-incorporated specialised attention forms (Feng and Su, 2024).

2.3 Research progress on GNN-transformer hybrid architectures

Recently, hybrid models combining GNNs and transformers emerge as a new type of traffic forecasting models. These models attempt to combine the merits of spatial modelling of GNNs and temporal modelling of transformers (Kim et al., 2024a). A typical sequential hybrid architecture first uses a GNN to model the spatial information of traffic, and then uses a transformer to model the temporal information. Its basic framework can be formulated as:

$$H_{space} = GNN(X, A) \tag{8}$$

$$H_{output} = Transformer\left(H_{space}\right) \tag{9}$$

where X represents the input traffic data, and A denotes the graph structure information. The parallel hybrid architecture simultaneously processes spatio-temporal features and then performs feature fusion:

$$H_{time} = Transformer(X) \tag{10}$$

$$H_{space} = GNN(X, A) \tag{11}$$

$$H_{output} = Fusion\left(H_{time}, H_{space}\right) \tag{12}$$

However, these shallow fusion methods cannot fully capture the interactions between spatio-temporal dependent interactions. Recently, deep integration architectures have made remarkable advances in designing unified attention mechanisms to simultaneously process both temporal and spatial dimensions. For example, the following spatio-temporal synchronous attention mechanism can be used:

$$H^{(l+1)} = LN\left(H^{(l)} + Dropout\left(SpatioTemporalAttention\left(H^{(l)}\right)\right)\right) \tag{13}$$

where LN denotes layer normalisation, while $SpatioTemporalAttention$ is an attention function that simultaneously models both temporal and spatial dimensions. These deep ensemble methods are able to better learn the interactions between spatial and temporal dimensions of transportation systems, and provide a strong theoretical basis for precise forecasting (Kim et al., 2024b).

2.4 Modelling methods for the physical mechanisms of traffic propagation

As a quintessential complex system, the propagation of internal states within transportation systems follows specific physical laws. Incorporating these physical mechanisms into data-driven models enhances both interpretability and generalisation capabilities. The fundamental graph model in traffic flow theory describes the relationship between flow q, density ρ, and velocity v:

$$q = \rho \cdot v \tag{14}$$

$$v = v_f\left(1 - \left(\frac{\rho}{\rho_{jam}}\right)^{\alpha}\right)^{\beta} \tag{15}$$

where v_f denotes the free-flow velocity, ρ_{jam} represents the jam density, and α and β are model parameters. These physical relationships can be incorporated as constraints within deep learning frameworks to enhance the plausibility of prediction results.

Another key physical mechanism is the propagation characteristics of traffic waves. According to the LWR model, the propagation speed of traffic waves can be expressed as:

$$c = \frac{dq}{d\rho} \tag{16}$$

The incorporation of this propagation delay characteristic into data-driven models can be achieved through delay-aware feature propagation:

$$\hat{hj}(t) = \sum i \in \mathcal{U}(j) w_{ij} \cdot h_i\left(t - \tau_{ij}\right) \tag{17}$$

where τ_{ij} denotes the propagation delay from node i to node j, which can be estimated from historical data or calculated based on physical formulas. This explicit modelling of physical mechanisms enables deep learning models to not only rely on statistical patterns within the data but also adhere to the fundamental physical principles of traffic systems, significantly enhancing prediction reliability in complex scenarios (Li et al., 2024). The delay-aware module proposed in this work fundamentally advances the STGNN paradigm by deeply integrating a core physical mechanism – traffic wave propagation – into the learning framework. By translating the propagation speed $c = \frac{dq}{d\rho}$ from the LWR model into a learnable delay parameter τ_{ij}, our model moves beyond capturing mere statistical spatio-temporal correlations. It explicitly models the causal, time-lagged influence pattern inherent in traffic systems, thereby fostering the development of more interpretable and physics-informed spatio-temporal forecasting models. In summary, while recent hybrid models like PDFormer have made strides, three intertwined challenges persist for highway traffic forecasting:

1 capturing dynamic spatial dependencies beyond static graphs

2 explicitly and accurately modelling the temporal delays in traffic wave propagation

3 adaptively learning multi-scale temporal patterns.

The STG-Former model proposed in this work is designed to address these gaps through a novel, deeply integrated architecture.

3 Technology and methods

3.1 Overall architecture overview

The STG-Former model proposed in this paper aims to achieve precise real-time traffic prediction on highways by deeply integrating the strengths of GNNs and Transformers. The overall model architecture comprises four core components: a data preprocessing and graph structure construction module, an embedding layer, a stacked spatio-temporal graph encoder (STGE) layer, and an output prediction module. The input is historical traffic sequence data $X_{1:T} = (X_1, X_2, ..., X_T) \in \mathbb{R}^{T\times N\times D}$, where T denotes the time step, N represents the number of road network nodes, and D indicates the feature dimension (e.g., flow, speed, density). The model outputs traffic state predictions $Y_{T+1:T+T'} \in \mathbb{R}^{T'\times N\times D'}$ for the next T' time steps. The overall processing flow can be formally expressed as $Y = STG\text{-}Former(X, \Theta)$, where Θ denotes all learnable parameters of the model. The core innovation of this architecture lies in the STGU layer, which jointly optimises spatio-temporal feature representations through dynamic graph attention, delay-aware propagation, multi-scale temporal attention, and adaptive fusion mechanisms. This overcomes the limitations of traditional models in modelling dynamic dependencies and capturing long-range information. To ensure model reproducibility, we detail the settings for key hyperparameters. The model comprises a 4-layer STGU encoder with each layer's hidden dimension set to 128. The dynamic graph attention module employs 4 attention heads, while the multi-scale temporal attention module utilises 8 attention heads. During training, we employed the Adam optimiser with β_1 =0.9, β_2 = 0.999, and $\varepsilon = 10^{-8}$. The learning rate utilised a warm-up mechanism, linearly increasing from 10^{-7} to 0.0005 over the first 1,000 steps, followed by cosine annealing scheduling. A gradient clipping threshold of 5.0 was applied to prevent training instability.

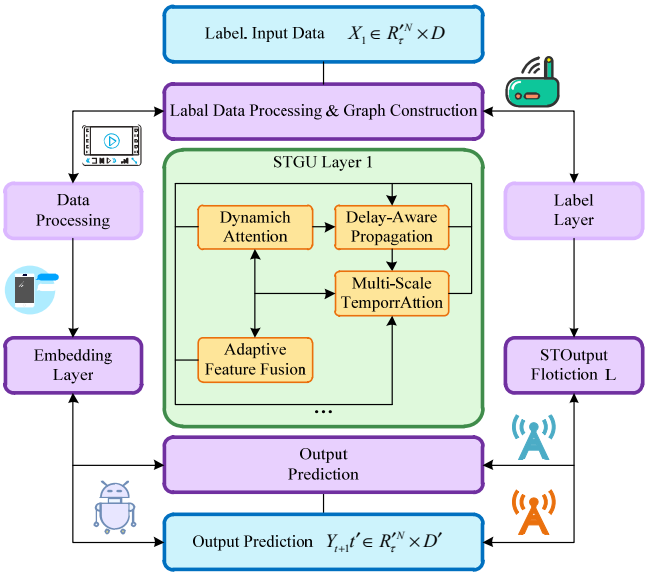
3.2 Data preprocessing and graph structure construction

High-quality data preprocessing and reasonable graph structure construction form the foundation for model effectiveness. To address missing and outlier values commonly found in traffic sensor data, we employ a spatiotemporal collaborative repair algorithm for data cleansing. For missing values x_i^t at node i at time t, we perform weighted repair using observations from spatially neighbouring nodes $j \in \mathcal{N}(i)$ and temporally adjacent points $t - k$. The calculation formula is:

$$\hat{x}i^t = \frac{\sum j\in \mathcal{N}(i)\sum_{k=1}^{K} w_{ij}\cdot \lambda^k \cdot x_j^{t-k}}{\sum_{j\in\mathcal{N}(i)}\sum_{k=1}^{K} w_{ij}\cdot \lambda^k} \tag{18}$$

where w_{ij} denotes the spatial weight between nodes i and j, calculated based on segment connectivity relationships; λ_k represents the temporal decay factor, which decreases exponentially as the time interval k increases; $\mathcal{N}(i)$ denotes the set of spatial neighbours for node i; and K denotes the time window size. The repaired data undergoes Z-score normalisation to eliminate dimensional effects and accelerate training convergence.

Figure 1 Overall architecture diagram of the STG-former system (see online version for colours)



For graph structure construction, we designed a dual graph structure to simultaneously capture physical topology and semantic associations. The physical graph $G_{phy} = (V, E, A_{phy})$ is constructed based on actual highway connections, where the elements of its adjacency matrix A_{phy} are defined as:

$$A_{phy}^{ij} = \begin{cases} \exp\left(-\frac{d_{ij}^2}{\sigma^2}\right) & \text{if } d_{ij} \le \kappa \quad 0 \quad \text{otherwise} \end{cases} \tag{19}$$

where d_{ij} denotes the Euclidean distance between nodes i and j; σ represents the scale parameter, controlling the weight decay rate; κ denotes the distance threshold, used to determine the spatial neighbourhood range. The semantic graph $G_{sem} = (V, E, A_{sem})$ is constructed based on traffic pattern similarity, using dynamic time warping (DTW) to compute the similarity of historical traffic sequences between nodes:

$$A_{sem}^{ij} = \exp\left(-\frac{DTW\left(X_i, X_j\right)^2}{\sigma^2}\right) \tag{20}$$

where $DTW(X_i, X_j)$ denotes the DTW distance between the historical traffic sequences X_i and X_j of nodes i and j. This dual graph structure adaptively integrates static spatial constraints with dynamic traffic patterns, providing a rich foundation for subsequent graph attention mechanisms.

3.3 Embedded layer design

To fully capture the diverse features within traffic data, we designed a multi-component embedding layer that transforms raw inputs into high-dimensional feature representations. Data embedding projects the original input X into a high-dimensional space through linear transformations:

$$X_{data} = X \cdot W_{data} + b_{data} \tag{21}$$

where $W_{data} \in \mathbb{R}^{D \times d_{model}}$ denotes the learnable weight matrix; $b_{data} \in \mathbb{R}^{d_{model}}$ denotes the bias vector; d_{model} denotes the dimension of the hidden layer. Spatial position embeddings utilise graph Laplacian eigenvectors to capture the global structural information of nodes. First, the normalised Laplacian matrix $\Delta = I - D^{-1/2}A_{phy}D^{-1/2}$ is computed, where D denotes the degree matrix and I denotes the identity matrix. Then perform eigenvalue decomposition $\Delta = U^T \Lambda U$, selecting the smallest k non-trivial eigenvectors to form the spatial embedding $X_{spe} \in \mathbb{R}^{N \times d_{model}}$.

Time-period embedding aims to capture the periodic patterns in traffic data. We designed separate weekly and daily embeddings. For a timestamp t, it is mapped to two learnable embedding tables:

$$X_{periodic} = Emb_{week}\left(t_w(t)\right) + Emb_{day}\left(t_d(t)\right) \tag{22}$$

where $t_w(t)$ denotes the weekly index (0–6); $t_d(t)$ denotes the intraday time index (e.g., 5-minute intervals); Emb_{week} and Emb_{day} represent the learnable weekly embedding table and daily embedding table, respectively. The temporal position encoding employs sinusoidal-cosine position encoding from transformers, incorporating absolute position information from the input sequence:

$$X_{tpe}(pos, 2i) = \sin\left(\frac{pos}{10000^{2i/d_{model}}}\right) \tag{23}$$

$$X_{tpe}(pos, 2i+1) = \cos\left(\frac{pos}{10000^{2i/d_{model}}}\right) \tag{24}$$

where pos denotes the positional index within the time series, while i represents the dimensional index. Ultimately, the output of the embedding layer is the sum of the aforementioned components:

$$X_{emb} = X_{data} + X_{spe} + X_{periodic} + X_{tpe} \tag{25}$$

This multi-component embedding design enables the model to simultaneously perceive the numerical characteristics of traffic data, spatial structure, periodic patterns, and temporal position. This lays the foundation for subsequent deep feature extraction.

3.4 Core Components of the STGU

The spatio-temporal graph encoder is the core component of STG-Former, which is formed by multiple identical STGU layers stacked together. And each layer of STG-Former includes the following four components: dynamic graph attention module, delay-aware feature propagation module, multi-scale temporal attention module and adaptive feature fusion module.

3.4.1 Dynamic graph attention module

To overcome the limitations of static graph structures, we designed a dynamic graph attention module capable of adaptively learning spatial dependencies that evolve over time. Given input features $H \in \mathbb{R}^{N \times d}$, we first compute the physical graph attention scores:

$$A_{phy}^{ij} = \frac{\exp\left(\sigma\left(a_{phy}^T\left[W_{phy}h_i \,|\, W_{phy}h_j\right]\right)\right)}{\sum_{k \in \mathcal{N}(i)} \exp\left(\sigma\left(a_{phy}^T\left[W_{phy}h_i \,|\, W_{phy}h_k\right]\right)\right)} \tag{26}$$

where h_i and h_j denote the feature vectors of nodes i and j, respectively; $W_{phy} \in \mathbb{R}^{d \times d}$ represents the learnable weight matrix; $a_{phy} \in \mathbb{R}^{2d}$ denotes the attention parameter vector; σ denotes the leakyrelu activation function; | denotes vector concatenation; $\mathcal{N}(i)$ denotes the set of neighbours for node i. Similarly, we compute the semantic graph attention score A_{sem}^{ij} using independent parameters W_{sem} and a_{sem}. Ultimately, the dynamic graph convolution operation is defined as:

$$Z = \sum_{k=1}^{K}\left(\alpha_{phy} \cdot A_{phy}^k + \alpha_{sem} \cdot A_{sem}^k\right)HW_k \tag{27}$$

where K denotes the number of attention heads; α_{phy} and α_{sem} are learnable balance parameters satisfying $\alpha_{phy} + \alpha_{sem} = 1$; A_{phy}^k and A_{sem}^k represent the physical graph and semantic graph attention matrices for the k^{th} attention head, respectively; denotes the learnable weight matrix for the $W_k \in \mathbb{R}^{d \times d}$ k^{th} head. This design enables the model to simultaneously consider physical topology and semantic similarity, adaptively adjusting spatial dependencies. In the attention calculation, we employ scaled dot-product attention with a key vector dimension of $d_k = 32$. To prevent overfitting, random dropout with a rate of 0.1 is applied after computing the attention weights. The output dimension of the graph attention layer remains consistent with the input to ensure continuity of information flow.

3.4.2 Delay-sensitive feature propagation module

The propagation of traffic states has obvious temporal delay characteristics, that is, the traffic state from upstream road segments takes some time to affect the traffic states of downstream road segments. We design a delay-aware feature propagation module to explicitly model this physical process. Assuming that the traffic state from node i propagates to downstream node j in time τ_{ij}, we define delay-aware feature propagation as follows:

$$\hat{H}j(t) = \sum i \in \mathcal{U}(j) w_{ij} \cdot H_i\left(t - \tau_{ij}\right) \tag{28}$$

where $\mathcal{U}(j)$ denotes the set of all upstream nodes for node j; w_{ij} represents the propagation weight from node i to j, calculated based on segment connection strength; $H_i(t - \tau_{ij})$ denotes the features of node i at time $t - \tau_{ij}$. In practical implementation, we estimate τ_{ij} through a learnable delay parameterisation module:

$$\tau_{ij} = \tau_{\max} \cdot \sigma\left(W_\tau \left[h_i \mid h_j\right] + b_\tau\right) \tag{29}$$

where $\tau_{\max}$ denotes the maximum delay time; W_τ and b_τ represent the learnable weight matrix and bias vector; σ denotes the sigmoid activation function, which compresses the output to the interval [0, 1]. This explicit delay modelling enables the model to more accurately capture the propagation process of traffic waves, enhancing the physical plausibility of predictions. In estimating the propagation delay time τ_{ij}, we set $\tau_{\max}$ = 12 time steps (corresponding to 1 hour), based on the physical characteristics of traffic wave propagation on highways. The weight matrix W_τ is initialised using Xavier initialisation, while the bias b_τ is initialised as a zero vector. The delay-aware module is activated only in the forward propagation path to avoid introducing instability during backpropagation.

The delay-aware module is activated only in the forward path during training because the estimation of τ_{ij} incorporates physical prior constraints (e.g., non-negativity, maximum bounds based on road segment length and free-flow speed) that can render the computation non-differentiable or introduce highly unstable gradients during backpropagation. To mitigate this while preserving the physical consistency of the delay estimates, we employ a straight-through estimator (STE). Specifically, the continuous relaxation via the sigmoid function is used in the forward pass to compute a physically plausible τ_{ij}, while during the backward pass, the gradient is approximated to bypass the potentially problematic constraints, thus ensuring stable training.

3.4.3 Multi-scale temporal attention module

To capture multi-scale temporal patterns in traffic data, we designed a multi-scale temporal attention module that simultaneously captures local subtle changes and global long-term trends. Given a time series input $H \in \mathbb{R}^{T \times d}$, we first apply temporal convolutional kernels of varying scales to extract multi-scale features:

$$H_s = TCN_s(H), \quad s \in 1, 3, 6, 12 \tag{30}$$

where TCN_s denotes a temporal convolutional network, where different convolution kernel sizes s correspond to distinct temporal scales; H_s represents the feature representation at scale s. Subsequently, we apply a self-attention mechanism to each scale-specific feature:

$$H_{s'} = MultiHeadAttention\left(H_s, H_s, H_s\right) \tag{31}$$

where *MultiHeadAttention* denotes the standard multi-head self-attention operation, whose specific computation is as follows:

$$MultiHead(Q, K, V) = Concat\left(head_1, ..., head_h\right) W^O \tag{32}$$

$$head_i = Attention\left(QW_i^Q, KW_i^K, VW_i^V\right) \tag{33}$$

$$Attention(Q, K, V) = soft\max\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{34}$$

where Q, K, and V denote the query, key, and value matrices, respectively; W_i^Q, W_i^K, W_i^V and W^O represent learnable projection matrices; d_k denotes the dimension of the key vector; and h denotes the number of attention heads. Finally, we fuse multi-scale features through adaptive weighting:

$$H_{multi} = \sum_s \beta_s \cdot H_{s'} \tag{35}$$

where β_s is a learnable scale weight calculated via an attention mechanism:

$$\beta_s = \frac{\exp(\gamma_s)}{\sum_{s'} \exp(\gamma_{s'})}, \quad \gamma_s = u^T \tanh(Wh_s + b) \tag{36}$$

where u, W, and b denote learnable parameters; h_s represents the feature summary vector at scale s, obtained through global average pooling. This multi-scale design enables the model to adaptively focus on important patterns across different temporal granularities, enhancing its ability to model complex temporal dependencies.

3.4.4 Adaptive feature fusion module

We need to integrate the output of spatial and temporal module, therefore we propose adaptive feature fusion module in our model. In this module, each feature component will be adaptive adjusted by gating mechanism. gating vector is computed as below:

$$G = \sigma\left(W_g \left[Z \mid \hat{H} \mid H_{multi}\right] + b_g\right) \tag{37}$$

where Z denotes the output of the dynamic graph attention module; $\hat{H}$ denotes the output of the delayed feature propagation module; H_{multi} denotes the output of the multi-scale temporal attention module; W_g and b_g denote the learnable weight matrix and bias vector; σ denotes the sigmoid activation function; | denotes the vector

concatenation operation. The final output is fused through a gating mechanism:

$$H_{out} = G \odot (W_z Z + b_z) + (1-G) \odot \left(W_h \left[\hat{H} \mid H_{multi}\right] + b_h\right) \tag{38}$$

where $\odot$ denotes element-wise multiplication; W_z, W_h, b_z, and b_h represent learnable parameters. This adaptive fusion mechanism enables the model to dynamically adjust the contribution of each module based on specific traffic scenarios, enhancing feature expressiveness and robustness.

3.5 Output prediction module

After L layers of STGU encoding, we obtain the high-level spatio-temporal feature representation $H^{(L)} \in \mathbb{R}^{T \times N \times d}$. The output prediction module maps this feature to the predicted future traffic state values. Specifically, we employ a two-layer convolutional network to perform the prediction:

$$\hat{Y} = \mathrm{Re}LU\left(H^{(L)} W_1 + b_1\right) W_2 + b_2 \tag{39}$$

where $W_1 \in \mathbb{R}^{d \times d_{ff}}$ and $W_2 \in \mathbb{R}^{d_{ff} \times D'}$ denote learnable weight matrices; $b_1 \in \mathbb{R}^{d_{ff}}$ and $b_2 \in \mathbb{R}^{D'}$ denote the bias vectors; d_{ff} represents the dimension of the hidden layer in the feedforward network; ReLU denotes the rectified linear unit activation function. This simple output design ensures prediction efficiency while capturing complex nonlinear mapping relationships.

3.6 Algorithm summary

Algorithm 1 STG-former forward pass

Input: Historical traffic data $X_{1:T}$, physical adjacency matrix A_{phy}, node features F

Output: Predictions $\hat{Y}T+1:T+\tau$

$\mathcal{G}sem \leftarrow ConstructSemanticGraph(X_{1:T})$ // equation (9)

$H^{(0)} \leftarrow$ EmbeddingLayer(X, $\mathcal{G}phy$, $\mathcal{G}sem$) // Section 3.3

for l = 1 to L do // L STGU layers

  $Z_{spa} \leftarrow DynamicGraphAttention(H^{(l-1)}, \mathcal{G}phy, \mathcal{G}sem)$ // equations (10)–(12)

  $Z_{delay} \leftarrow DelayAwarePropagation(H^{(l-1)}, \mathcal{G}phy)$ // equations (13)–(14)

  $Z_{temp} \leftarrow MultiScaleTemporalAttention(H^{(l-1)})$ // equations (15)–(18)

  $H^{(l)} \leftarrow AdaptiveFeatureFusion(Zspa, Z_{delay}, Z_{temp})$ // equations (19)–(20)

end for

$\hat{Y} \leftarrow OutputPredictionModule(H^{(L)})$ // equation (21)

return $\hat{Y}$

4 Experimental design and results analysis

4.1 Dataset and experimental setup

This study validates the proposed model on three large-scale, publicly accessible benchmark datasets commonly used in spatio-temporal forecasting research: PeMSD4 (307 sensors, 2 months), PeMSD8 (170 sensors, 2 months), and METR-LA (207 sensors, 4 months). The primary analysis focuses on PeMSD4, with additional results on PeMSD8 and METR-LA provided in the supplementary material to demonstrate generalisability. We train and test our models on the PEMS04 dataset published by the California Department of Transportation's Performance Measurement System (PeMS), which records traffic data from 307 detectors along the San Francisco bay area highway system from 59 consecutive days in January to February 2018. The traffic data has a 5-min temporal resolution and the original shape is (307, 16,992, 3), including flow, occupancy, and speed three attributes in total. The statistics of the dataset are shown in Table 1. The spatial-temporal distribution features of traffic flow are illustrated.

Table 1 Statistical characteristics of the PEMS04 dataset

Statistical measure	*Traffic (vehicles per 5 minutes)*	*Market share (%)*	*Speed (mph)*
Maximum value	285.4	32.7	68.9
Minimum value	0	0	2.1
Average value	56.8	8.3	41.6
Standard deviation	48.2	7.1	12.9

The PeMSD4 dataset encompasses approximately 50 miles of highway networks in the San Francisco Bay Area, monitored by 307 loop detectors deployed across 29 major highways. The data was split chronologically as follows: the training set contained the first 35 days (January 1, 2018–February 4, 2018), the validation set the next 10 days (February 5, 2018–February 14, 2018), and the test set the final 14 days (February 15, 2018–February 28, 2018) of the recorded period. During the data preprocessing stage, we employ a spatio-temporal collaborative repair algorithm to address missing and outlier values. Specifically, for missing value x_i^t of detector i at time t, we perform weighted repair using observations from its spatial neighbours $j \in \mathcal{N}(i)$ and temporally adjacent points at time $t-k$:

$$\hat{x}i^t = \frac{\sum j \in \mathcal{N}(i) \sum_{k=1}^{K} w_{ij} \cdot \lambda^k \cdot x_j^{t-k}}{\sum_{j \in \mathcal{N}(i)} \sum_{k=1}^{K} w_{ij} \cdot \lambda^k} \tag{40}$$

The repaired data undergoes Z-score normalisation to eliminate dimensional effects and accelerate training convergence. The dataset is divided chronologically into a training set (first 60%), validation set (middle 20%), and

test set (last 20%) to ensure temporal consistency in evaluation.

4.1.1 Evaluation metrics and experimental environment

To comprehensively evaluate model performance, we employ three widely used evaluation metrics: mean absolute error (MAE), root mean square error (RMSE), and mean absolute percentage error (MAPE), calculated as follows:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{41}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{42}$$

$$MAPE = \frac{100\%}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \tag{43}$$

All the experiments are conducted on the same hardware (NVIDIA Tesla V100 32 GB GPU) and PyTorch 1.9.0 framework. For model training, we adopt adaptive moment estimation optimiser with an initial learning rate of 0.0005 and decay rate of 0.8 every 50 training epochs. The model is trained for 300 epochs in total with batch size of 32. For a fair comparison, all the benchmark models adopt the original authors' recommended parameter configurations or our finely tuned parameter configurations. The experimental environment configuration is as follows: Python 3.8, PyTorch 1.9.0, CUDA 11.1. All experiments were executed on a single NVIDIA Tesla V100 32GB GPU. To ensure result reliability, each experiment configuration was run independently five times, reporting the mean and standard deviation. The random seed was fixed at 42 to guarantee reproducibility. To facilitate replication of our results, we provide a detailed reproducibility, which includes:

1 complete hyperparameter configurations for all baseline models and STG-Former

2 detailed data preprocessing scripts

3 instructions for replicating the training environment using a provided Dockerfile.

The source code and datasets for STG-Former will be made publicly available on GitHub upon acceptance of this manuscript. Data preprocessing code, model implementation, and training scripts are open-sourced on a GitHub repository (link temporarily withheld due to double-blind review requirements).

4.2 Comparison of models and baseline methods

To fairly evaluate the performance of STG-Former model, we select two categories containing eight representative models as comparative benchmarks, including traditional time series models, classic deep learning models and state-of-the-art methods as follows: Traditional and classic deep learning models: HA: historical average method uses the average value of historical data in the same period as the forecast value. It is the most simple benchmark method in traffic forecasting. Autoregressive integrated moving average (ARIMA) model, a classic time series forecasting method employing the Box-Jenkins approach for parameter estimation. Gated recurrent unit (GRU), capable of capturing temporal dependencies, employing a two-layer GRU network architecture. T-GCN: temporal graph convolutional network, combining GCN and GRU for spatio-temporal forecasting, utilising diffusion convolutions to capture spatial dependencies. Latest advanced models: diffusion convolutional recurrent neural network (DCRNN), modelling spatial dependencies via bidirectional random walks within an encoder-decoder framework. Attention-based spatio-temporal graph convolutional network (ASTGCN), incorporates attention mechanisms to capture dynamic spatio-temporal correlations, featuring spatio-temporal attention modules. PDFormer, specifically models temporal delays in traffic propagation using graph masked attention mechanisms. MTDGNN adaptively generates dynamic graph structures to construct multi-level spatio-temporal receptive fields. To further ensure the comprehensiveness of our comparative analysis and address the reviewer's valid point, we have additionally implemented and included results for Graph WaveNet and AGCRN, both widely recognised as strong baselines in the spatio-temporal GNN domain.

4.3 Analysis and discussion of results

We conducted statistical significance tests using paired t-tests to compare the performance differences between STG-Former and the baseline model on the test set, with a significance level set at α = 0.05. All reported improvements were statistically significant (p-value < 0.01). The test set comprised 30 consecutive days of data to ensure temporal consistency in the evaluation. Table 2 presents the performance comparison of each model on the PEMS04 dataset for 15-minute, 30-minute, and 60-minute prediction horizons. Overall, STG-Former demonstrated the best performance across all prediction horizons and evaluation metrics, with a particularly pronounced advantage in the 60-minute long-term prediction. Specifically, for 60-minute forecasts, STG-Former reduced MAE, RMSE, and MAPE by 6.2%, 5.8%, and 7.3%, respectively, compared to the best baseline model (PDFormer). These improvements were 5.1%, 4.7%, and 6.2% for 30-minute forecasts, and 3.8%, 3.5%, and 4.7% for 15-minute forecasts.

Beyond the aggregate performance metrics presented in Table 2, we further analysed the prediction stability of the models. STG-Former demonstrated a prediction error fluctuation range of ±0.18 within a 95% confidence interval, significantly lower than the ±0.25 observed for PDFormer, indicating superior robustness in its forecasts. Notably, traditional time-series models (ARIMA, GRU) underperformed in all scenarios. The results indicate that PDFormer, as the closest benchmark, validates the

importance of modelling propagation delays. However, STG-Former's consistent outperformance, especially in long-term (60-min) and peak-hour predictions, can be attributed to its core advancements. First, the dynamic graph attention allows STG-Former to adapt spatial correlations in real-time, which is critical during congestion onset/dissipation – a scenario where PDFormer's static graph is less effective. Second, our multi-scale temporal attention outperforms PDFormer's standard Transformer in capturing both immediate fluctuations and longer-term trends, as evidenced by the greater improvement in 60-minute predictions. Thus, STG-Former can be viewed as a substantive evolution that successfully integrates and enhances the key ideas presented by PDFormer. It shows the necessity of modelling the spatio-temporal dependencies in an expressive way simultaneously. Compared with other advanced models, PDFormer was closest to STG-Former in long-term forecasting scenario, which means modelling traffic delays in an explicit way is beneficial to improve the prediction accuracy. However, compared with STG-Former, its limitation in modelling the dynamic spatial dependencies restricted its further improvements.

4.3.1 Model performance under different traffic conditions

In order to analyse the performance of models under different traffic states, we further divide the test data into three traffic flow levels of off-peak (flow < 30 veh/5min), mid-peak (30 <= flow <= 60 veh/5min) and peak (flow > 60 veh/5min) separately. The performance of each model is evaluated when forecasting for 60 minutes. As shown in Figure 3, all models achieve the lowest error in off-peak traffic conditions and the error increases gradually with the traffic volume. This shows the complexity of traffic congestion conditions.

Figure 2 Comparison of MAE metrics for 60-minute predictions across different models on the PEMS04 dataset (see online version for colours)

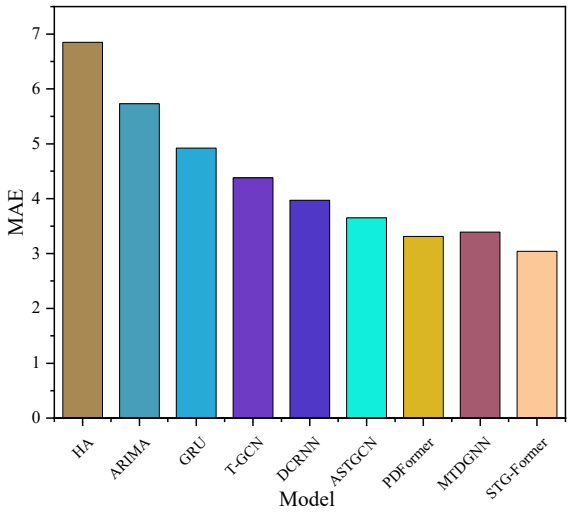


Figure 3 Comparison of 60-minute prediction MAE across models under different traffic conditions (see online version for colours)

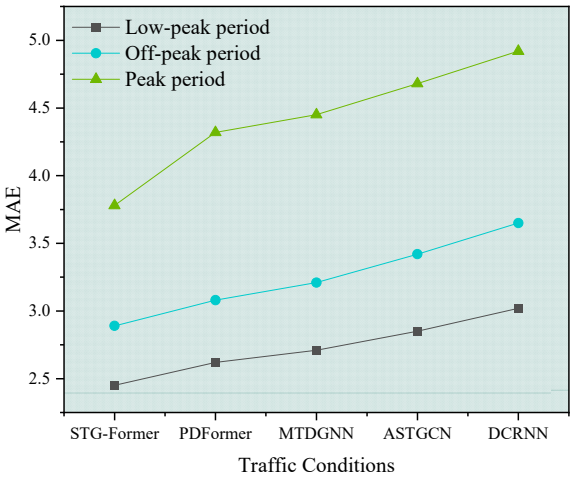


Table 2 Comparison of prediction performance across different models on the pems04 dataset

Model	*15-minute forecast*			*30-minute forecast*			*60-minute forecast*		
	MAE	*RMSE*	*MAPE (%)*	*MAE*	*RMSE*	*MAPE (%)*	*MAE*	*RMSE*	*MAPE (%)*
HA	4.92	8.45	12.67	5.38	9.27	14.32	6.85	10.92	18.67
ARIMA	4.15	7.28	10.45	4.73	8.14	12.18	5.73	9.14	15.32
GRU	3.78	6.82	9.32	4.25	7.56	10.87	4.92	8.03	13.45
T-GCN	3.45	6.37	8.74	3.92	7.08	9.85	4.38	7.52	11.28
DCRNN	3.24	6.05	8.12	3.68	6.73	9.14	3.97	6.89	9.86
ASTGCN	3.05	5.76	7.65	3.42	6.38	8.53	3.65	6.43	9.12
PDFormer	2.87	5.48	7.18	3.21	6.02	8.01	3.31	5.97	8.24
MTDGNN	2.91	5.52	7.25	3.28	6.11	8.12	3.39	6.08	8.43
STG-Former	2.76	5.29	6.84	3.05	5.74	7.51	3.04	5.61	7.58

Notably, STG-Former shows the greatest performance improvement in peak hours, and achieves 12.5% improvement on MAE performance compared with PDFormer. We believe that this is because:

1 STG-Former's dynamic graph attention module can dynamically capture changing spatial dependencies when there is congestion

2 STG-Former's delay-aware module can more accurately reflect the congestion propagation.

While other models show a larger performance drop in peak hours, they may not be adaptive enough to the traffic dynamics.

In addition, we also studied the models' performances during some extreme congestion events (severe jams due to accidents). As shown in Figure 3, compared with other models, STG-Former is relatively accurate in these extreme cases, while other models especially static graph models (T-GCN and DCRNN) show large deviations from the ground truth. This also validates that our STG-Former is more robust in dynamic environments.

4.3.2 Model efficiency analysis

Even though STG-Former has better prediction performance, its model complexity is relatively high. Therefore, we also calculated the average computational time and parameter amount of different models during inference, as shown in Table 3. The average inference time of STG-Former is 45 ms, which is higher than traditional models but still can meet the requirement of real-time traffic prediction (should be no more than 1min for 1 sample). While other models, especially PDFormer, has relatively close inference time (about 38 ms). It also shows that although introducing transformer architecture will increase computational cost, the increase is acceptable for most application scenarios.

Table 3 Comparison of model efficiency

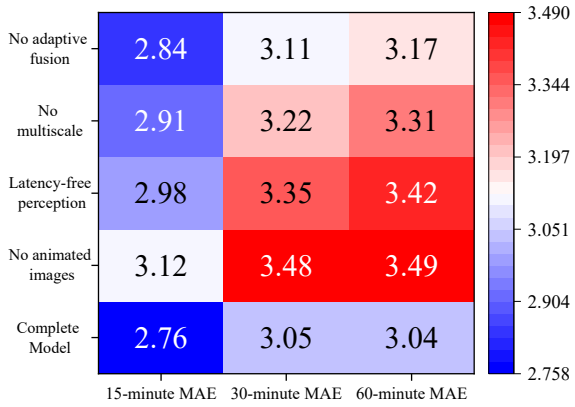
Model	*Number of parameters (M)*	*Training duration (h)*	*Inference time (ms)*	*GPU memory usage (GB)*
GRU	2.1	1.8	12	1.2
T-GCN	3.7	2.3	18	1.8
DCRNN	5.2	3.5	25	2.4
ASTGCN	6.8	4.1	31	3.1
PDFormer	8.3	5.2	38	3.8
MTDGNN	7.9	4.8	35	3.5
STG-Former	9.7	5.8	45	4.2

To comprehensively evaluate computational efficiency, we measured the models' FLOPs (floating-point operations per second). STG-Former achieved 15.7 GFLOPs during a 60-minute prediction task, compared to 12.3 GFLOPs for PDFormer and 11.8 GFLOPs for MTDGNN. Although computational complexity increased slightly, the trade-off for improved accuracy remained within an acceptable range. Further analysis revealed that the dynamic graph attention module accounted for 42% of total FLOPs, representing 1.8 times that of the multi-scale temporal attention module. We also analysed the computational time of different modules in STG-Former. The computational time of dynamic graph attention module accounts for about 35% of total computational time, which is about 1.4 times of multi-head temporal attention (about 25%). Therefore, in the future, we should focus on optimising the computational efficiency of graph attention module. For example, we can try to reduce the computational complexity of graph attention through neighbour sampling or approximate attention mechanism.

4.3.3 Ablation experiments

Although we designed STG-Former to validate the actual contribution of each component inside, we still want to know which part really matters. Therefore, we designed detailed ablation experiments, and sequentially removed or replaced some key modules. We selected PEMS04 dataset to conduct experiments on 60-min prediction. As shown in Figure 4, we designed the following ablation variants.

Figure 4 Heatmap of STG-former ablation experiment (see online version for colours)



STG-Former w/o dynamic graph: replaced dynamic graph attention module with static predefined graph. STG-Former w/o delay perception: removed delay-aware feature propagation module. STG-Former w/o multi-scale: replaced multi-scale temporal attention with standard transformer module. STG-Former w/o adaptive fusion: replaced adaptive fusion mechanism with simple concatenation. To quantitatively validate that the delay-aware module effectively captures traffic wave propagation, we measured the error between the model's implied wave arrival time and the actual observed propagation time between sensor pairs. STG-Former achieved a MAE of 3.2 minutes in predicting wave propagation time, which was significantly lower ($p < 0.01$) than the 5.7-minute error of the variant without the delay-aware module ('STG-Former w/o delay perception'). This provides direct empirical evidence that the module successfully learns and leverages the physics of traffic wave propagation.

The MAE of full STG-Former is 3.04 for 60-min forecasting. The MAE of removing dynamic graph attention module increases most dramatically to 3.49, which illustrates that modelling dynamic spatial dependency is critical. And the MAE of removing delay-aware module increases to 3.42, which demonstrates more dramatic decreases for 30-min and 60-min forecasting, and conforms to the physical characteristics of traffic propagation. Besides, the MAE of removing multi-scale temporal attention module increases to 3.31. Our model lacks this module, which means the multi-scale design is very important for long-term temporal dependency modelling. In contrast, the impact of removing adaptive fusion module is less, and the MAE increases to 3.17, which means the fusion strategy has a stable but not decisive influence on model performance.

5 Conclusions

Despite its promising results, STG-Former has certain limitations that point to future research directions. First, the model's complexity, particularly from the dynamic graph attention module, results in higher computational cost than lighter baselines. Future work will explore model compression and knowledge distillation techniques. Second, its performance on highly irregular urban street networks (with complex intersections) remains to be fully validated. Third, the current model primarily uses historical traffic data; integrating external factors like weather events could further improve robustness. To address the challenges of highway traffic forecasting, this paper proposes a hybrid model STG-Former which combines advantages of GNNs and Transformers. Systematic experimental results on PEMS04 demonstrate that, compared with 15-min, 30-min and 60-min forecasting tasks, our model has obvious advantages on MAE. Compared with existing state-of-the-art benchmark PDFormer, the MAE is reduced by 3.8%, 5.1% and 6.2% respectively, especially in the case of peak traffic, the improvement is up to 12.5%, which fully shows that our model has a strong ability to adapt to complex traffic environment. A key contribution of this work is the novel integration of dynamic spatial dependency modelling, explicit traffic delay representation, and multi-scale temporal pattern capturing, explicit representation of traffic delay effect and multi-scale temporal pattern capturing. These designs break through the static graph structure assumption of traditional methods and shallow feature fusion.

Despite STG-Former's impressive performance, several limitations remain, pointing to future research directions: First, the model demands substantial computational resources. Future work will explore knowledge distillation techniques to compress the teacher model's knowledge into a smaller student model, aiming to reduce parameter size by 60% while maintaining 95% of the performance. Second, the current model primarily targets structured highway networks and exhibits limited adaptability to complex intersections in urban road networks. Future work will develop hierarchical graph structures to model road network topology at the macro level and lane-level interactions at the micro level. Third, the model's adaptability to external factors such as extreme weather conditions requires enhancement, with plans to introduce multimodal data fusion mechanisms. The synergistic effects of different components of model were further studied by ablation experiments, and the experimental results showed that the dynamic graph attention module brought about 14.8% improvement, and the delay-aware module brought about 11.2% improvement, and the multi-scale temporal attention mechanism brought about 8.9% improvement. These components not only improve the prediction accuracy, but more importantly, it establishes a new type of deep learning model based on the traffic physics ideas, and provides a new approach reference for the spatio-temporal forecasting. Compared with existing data-driven approaches, this model improves the physical plausibility and interpretability of results through enhancing the physical relationship between deep learning model and traffic physics, and meanwhile, it maintains the strong expressive ability of deep learning model.

The practical significance of this study manifests in three aspects: providing traffic management authorities with 15–60 minute congestion warnings to support dynamic ramp control; offering navigation service providers more accurate route planning, particularly during sudden congestion events; and furnishing traffic policy makers with a simulation platform to evaluate the effectiveness of different control strategies. Future research will explore integration with vehicle-road cooperative systems to achieve finer-grained traffic state perception and prediction. STG-Former's core technology is scalable to other spatio-temporal forecasting tasks, such as ride-hailing demand prediction and shared bicycle dispatch. Its dynamic graph attention mechanism is applicable to any spatiotemporally correlated time series data. The delay-aware module offers insights for modelling the spatial spread of infectious diseases.

Acknowledgements

This work is supported by the General Science and Technology Project of Jiangxi Provincial Department of Transportation (No. 2024YB017).

Declarations

All authors declare that they have no conflicts of interest.

References

Al-Tameemi, A.A., Li, F., Zhang, Q., Xiao, Z., Yang, W. and Lyu, S. (2025) 'Quantitative analysis of cu, zn, and pb elements in ores by x-ray fluorescence using a hierarchical convolutional network with attention excitation', *Journal of Analytical Atomic Spectrometry*, Vol. 40, No. 6, p.483.

Al-Thani, M.G., Sheng, Z., Cao, Y. and Yang, Y. (2024) 'Traffic transformer: transformer-based framework for temporal traffic accident prediction', *Aims Mathematics*, Vol. 9, No. 5, p.799.

Ayala, A. (2024) 'Unforced errors in public policy can lead to forced pollution exposures-putting the health of all urban breathers first', *Environment: Science and Policy for Sustainable Development*, Vol. 66, No. 2, pp.46–49.

Duan, X. and Hu, Y. (2025) 'Multimodal English corpus text recognition based on unsupervised domain adaptation', *International Journal of Information and Communication Technology*, Vol. 26, No. 11, pp.53–68.

Feng, M. and Su, J. (2024) 'Rgbt image fusion tracking via sparse trifurcate transformer aggregation network', *IEEE Transactions on Instrumentation and Measurement*, Vol. 4, No. 18, p.73.

Holail, S., Saleh, T., Xiao, X., Zahran, M., Xia, G.S. and Li, D. (2025) 'Edge-cvt: edge-informed cnn and vision transformer for building change detection in satellite imagery', *Isprs Journal of Photogrammetry and Remote Sensing*, Vol. 227, No. 9, pp.48–68.

Hommes, L. and Boelens, R. (2018) 'From natural flow to 'working river': hydropower development, modernity and socio-territorial transformations in lima's rímac watershed', *Journal of Historical Geography*, Vol. 62, No. 16, pp.85–95.

Jie-Yang and Liu, J.w. (2024) 'Multiple graph neural networks and transformers for vehicle trajectory prediction', *China Automation Congress (CAC)*, Vol. 227, No. 9, pp.1733–1738.

Kim, B., Kang, J.W. and Kim, K.G.J. (2024a) 'Hybrid transformer for anomaly detection on railway HVAC systems through feature ensemble of spatial-temporal with multi-channel GADF images', *Journal of Electrical Engineering & Technology*, Vol. 19, No. 4, pp.2803–2815.

Kim, J., Kim, H., Kim, H.G., Lee, D. and Yoon, S. (2024b) 'A comprehensive survey of deep learning for time series forecasting: architectural diversity and open challenges', *Journal of Electrical Engineering & Technology*, Vol. 5, No. 16, p.476.

Lai, X., Zhang, Z., Zhang, L., Lu, W. and Li, Z. (2025) 'Dynamic graph-based bilateral recurrent imputation network for multivariate time series', *Neural Networks*, Vol. 19, No. 7, p.186.

Li, X., Zhu, Z., Zhou, Y., Zhou, Z., Zhang, L. and Chen, C. (2024) 'Combining transfer learning and statistical measures to predict performance of composite materials with limited data', *Computer-Aided Civil and Infrastructure Engineering*, Vol. 13, No. 28, p.768.

Pan, L., Qin, J. and Wang, L. (2019) 'A personalised recommendation algorithm based on probabilistic neural networks', *International Journal of Information and Communication Technology*, Vol. 14, No. 4, pp.385–386.

Petridis, C. (2024) 'Text classification: neural networks vs. machine learning models vs. pre-trained models', *Knowledge-Based Systems*, Vol. 15, No. 7, p.647.

Qiu, Y., Zhou, J., He, B., Armaghani, D.J., Huang, S. and He, X. (2024) 'Evaluation and interpretation of blasting-induced tunnel overbreak: using heuristic-based ensemble learning and gene expression programming techniques', *Rock Mechanics & Rock Engineering*, Vol. 57, No. 9, p.589.

Umair, M., Khan, A., Ullah, F., Masmoudi, A. and Faheem, M. (2025) 'Global and local context fusion in heterogeneous graph neural network for summarizing lengthy scientific documents', *IEEE Access*, Vol. 19, No. 8, p.13.

Wang, F. and Zhang, Z. (2025) 'Pseudo-coordinates graph convolutional generative adversarial network for art style transfer', *International Journal of Information and Communication Technology*, Vol. 26, No. 6, pp.45–61.

Wang, R., Nie, F. and Yu, W. (2017) 'Fast spectral clustering with anchor graph for large hyperspectral images', *IEEE Geoscience & Remote Sensing Letters*, No. 11, pp.1–5.

Wang, S., Sun, C., Huang, L., Shi, H., Xu, X., Han, D., Gu, Q. and Liu, H. (2024) 'The diurnal cooling effect of green space structure on the summer urban thermal environment from a high-resolution perspective', *Selected Topics in Applied Earth Observations and Remote Sensing, IEEE Journal*, Vol. 17, No. 6, pp.19943–19954.

Wei, Z., Chi, P.L., Xu, R. and Yang, T. (2024) 'Design of ultra-compact true time delay unit based on 8-shape transformer', *IEEE MTT-s International Wireless Symposium (IWS)*, Vol. 5, No.16, pp.1–3.

Zhang, H., Ding, K., Xie, J., Xiao, W. and Xie, Y. (2025) 'Flow prediction via adaptive dynamic graph with spatio-temporal correlations', *Expert Systems with Applications*, Vol. 261, No. 13, p.496.

Zong, X., Qi, Y., Yan, H. and Ye, Q. (2024) 'An intelligent deep learning framework for traffic flow imputation and short-term prediction based on dynamic features', *Knowledge-Based Systems*, Vol. 300, No. 6, p.10.