ADG-Former: Adaptive Dynamic Graph and Gated Fusion Transformer for Traffic Flow Prediction

Chunyang Shi
School of Computer Science and Technology
Beihua University
Jilin, China
email:1209739796@qq.com

Dongmin Chao
School of Computer Science and Technology
Beihua University
Jilin, China
email:chao13280057449@163.com

Lingling Zhang*
School of Computer Science and Technology
Beihua University
Jilin, China
*Corresponding author: zhangling_beihua@163.com

Jianhui Lv
Multi-modal Data Fusion and Precision Medicine Laboratory
The First Affiliated Hospital of Jinzhou Medical University
Jinzhou , China
email:lvjh@jzmu.edu.cn

***Abstract*—Traffic flow prediction has been recognized as a pivotal technology in the development of smart cities, facilitating data support for resource scheduling and traffic management. By leveraging an efficient spatiotemporal graph attention mechanism, outstanding performance in balancing prediction accuracy and computational efficiency has been demonstrated by the STGformer model. However, a static adjacency matrix fails to accommodate dynamic changes in road connectivity within actual traffic scenarios, such as road closures or peak-hour congestion, consequently limiting prediction performance under complex conditions. To address these limitations, an enhanced STGformer model, termed ADG-Former, is introduced based on adaptive optimization of dynamic graph structures. Initially, a dynamic adjacency matrix is constructed using spatiotemporal correlations of traffic flow and speed, wherein effective connectivity relationships between nodes are updated in real time through an adaptive threshold strategy. Subsequently, a gated fusion module is employed to dynamically balance the contribution weights of static road topology and dynamic traffic dependencies, enhancing adaptability to complex scenarios. This study presents a feasible optimization scheme for traffic flow prediction in dynamic scenarios and demonstrates engineering applicability.**

Keywords—traffic prediction, dynamic adjacency matrix, spatial-temporal graph transformer, gating fusion

I. INTRODUCTION

Traffic flow prediction is a core component of intelligent transportation systems (ITS) [1]. Research has shifted from traditional statistical models to data-driven deep learning, and spatiotemporal graph neural networks (STGNNs) have become the mainstream paradigm for their strong ability in modeling spatiotemporal dependencies [2] [3].

STGformer achieves high efficiency and accuracy by integrating graph convolution and linear attention, but it is limited by a fixed static adjacency matrix [4]. Such a static structure cannot capture real-time road changes caused by congestion, accidents, or road closures, nor can it model the time-delay effect in traffic propagation, leading to degraded performance in dynamic scenarios [5] [6].

Existing dynamic graph methods either rely on a single traffic indicator or introduce heavy computation overhead, making it difficult to balance adaptability and efficiency [7] [8] [9]. To address these issues, we propose ADG-Former, an enhanced model based on STGformer. It builds a multi-dimensional dynamic adjacency matrix by fusing flow-speed correlation and congestion propagation delay, and uses an adaptive gated fusion module to dynamically adjust the weights of static topology and dynamic dependencies according to real-time traffic density.

The main contributions of this study are outlined as follows:

- We propose a lightweight dynamic graph model ADG-Former that preserves linear complexity and significantly improves adaptability to dynamic traffic conditions.
- We construct a multi-dimensional adaptive dynamic adjacency matrix with real-time update and noise filtering based on traffic density.
- We design a learnable gated fusion mechanism to balance static road structure and dynamic traffic relations.
- Extensive experiments on three real-world datasets validate the effectiveness, superiority, and interpretability of the proposed model..

II. RELATED WORK

Traffic flow forecasting has evolved from linear-constrained statistical methods (e.g., ARIMA) to deep learning [1]. Spatiotemporal Graph Neural Networks (STGNNs) are now the dominant framework. Representative models such as STGCN, GWNET, and DCRNN effectively extract local spatial features but struggle to capture long-range global spatiotemporal dependencies due to limited receptive fields [10] [11] [12].

Transformer-based models (STAEformer, PDFormer) use self-attention for global modeling but suffer from quadratic complexity, which is unsuitable for large-scale road networks [13] [14]. STGformer achieves high efficiency with linear attention while maintaining accuracy, yet it still relies on a predefined static adjacency matrix [4] and ignores dynamic road connectivity changes.

Dynamic graph methods (DGCRN, D2STGNN, DSTAGNN) aim to handle temporal variations, but most either increase complexity significantly, rely on single traffic metrics, or use fixed thresholds [7] [8] [9]. They fail to efficiently integrate multi-feature correlations and adapt to diverse congestion scenarios.

III. METHODOLOGY

A. Problem Definition

The number of traffic sensor nodes in the road network be N. The entire road network structure is defined as an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, A)$ where $\mathcal{V}$ denotes the set of nodes ($|\mathcal{V}| = N$), $\mathcal{E}$ denotes the set of edges and $\mathbf{A} \in \mathbb{R}^{N\times N}$ is the adjacency matrix of the graph. If $v_i, v_j \in \mathcal{V}$ and $(v_i, v_j) \in \mathcal{E}$ then $A_{i,j}=1$; otherwise, $A_{i,j}=0$. Given the road network graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, A)$, the traffic state information observed at each node over the previous T time steps is denoted as $\mathbf{X} = [\mathbf{X}^{t-T+1}, \mathbf{X}^{t-T+2}, \dots, \mathbf{X}^{t}]$, where $\mathbf{X} \in \mathbb{R}^{T\times N\times C}$. Here, T is the input time step length, and C is the feature dimension (including core indicators such as traffic flow and speed). The goal of traffic flow prediction is to map the historical traffic state $\mathbf{X}$ to the traffic flow $\mathbf{Y} = [\mathbf{X}^{t+1}, \mathbf{X}^{t+2}, \dots, \mathbf{X}^{t+S}]$ for the future S time steps, where $\mathbf{Y} \in \mathbb{R}^{S\times N\times 1}$ and $S = 12$ (predicting traffic flow for the next 60 minutes with a 5-minute interval).

B. Model Architecture

ADG-Former is an improved STGformer for dynamic traffic scenarios. As shown in Fig. 1, it consists of five modules: data embedding, multi-dimensional dynamic adjacency matrix construction, adaptive gated fusion, graph propagation with spatiotemporal attention, and prediction layer. These cascaded modules enable end-to-end prediction while retaining the lightweight property.

The key improvement is the addition of a dynamic adjacency matrix and a gated fusion module. Traffic data is embedded into high-dimensional features, and a dynamic matrix is built in parallel. The gated module fuses the dynamic matrix and static topology to generate adaptive adjacency relations. Finally, the spatiotemporal attention module captures global dependencies and outputs predictions.

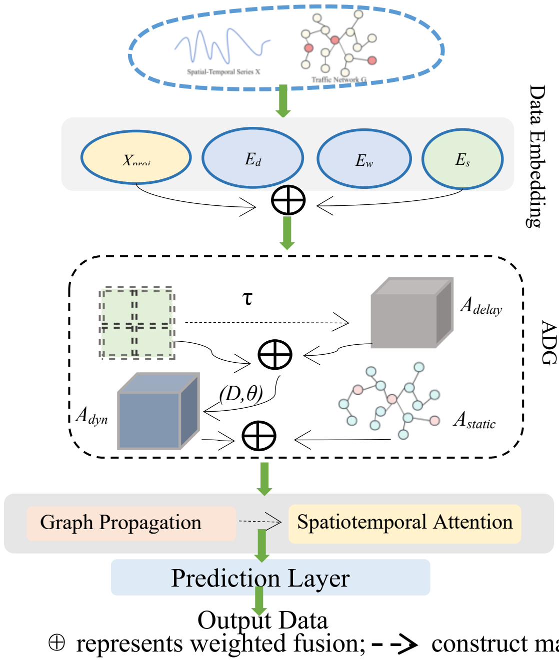

Fig. 1. The overall architecture of ADG-Former.

(1) *Data Embedding Layer*

The data embedding layer converts low-dimensional raw traffic data into high-quality high-dimensional features, providing reliable inputs for dynamic adjacency matrix construction and spatiotemporal correlation modeling. It includes three steps:

Raw data projection: The input data is mapped to the embedding dimension of d = 32 via a single-layer fully connected layer to yield $X_{proj} \in \mathbb{R}^{T\times N\times d}$, with the calculation formula expressed as:

$$X_{proj} = \sigma\,(\,X \cdot W_p + b_p\,) \tag{1}$$

where $W_p \in \mathbb{R}^{C\times d}$ denotes the projection weight matrix, $b_p \in \mathbb{R}^{d}$ is the bias term, and the ReLU activation function is adopted for σ to enhance the nonlinear expression capability of features.

Periodic Feature Embedding: To effectively capture temporal periodicity, learnable embedding vectors are introduced, including $E_d \in \mathbb{R}^{288\times d}$ (covering an entire day with 5-minute intervals) and $E_w \in \mathbb{R}^{2016\times d}$ (representing a full week). The corresponding embedding vectors are matched according to the timestamps of the data and then element-wise added to X_{proj} to achieve feature fusion, This process enables efficient fusion of periodic features without introducing significant computational complexity.

Spatiotemporal positional encoding: Sinusoidal function-based positional encoding $E_{st} \in \mathbb{R}^{N\times T\times d}$ is introduced to separately mark the spatial positions of nodes and the chronological order of time steps [15], preventing the model from confusing feature correlations across different spatiotemporal scales.

The final embedding output is expressed as:

$$X_{emb} = X_{proj} + E_d + E_w + E_{st} \tag{2}$$

(2) *Multi-dimensional Dynamic Adjacency Matrix*

This module is the core innovation of ADG-Former. It uses embedded high-dimensional features to build a dynamic adjacency matrix that reflects real-time traffic states, overcoming the limitations of static graphs. As shown in Fig. 2, it includes three steps: multi-index fusion, delay effect modeling, and adaptive filtering.

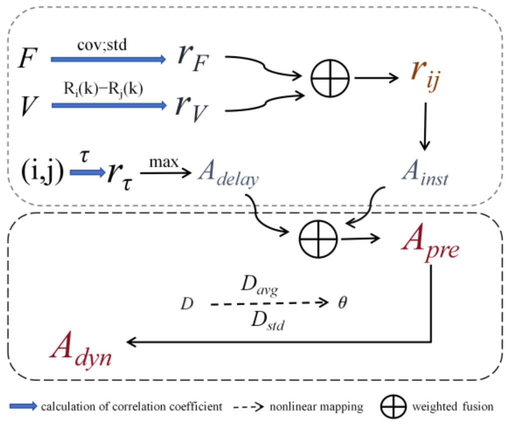

Fig. 2. Construction of Multi-Dimensional Dynamic Adjacency Matrix.

Multi-index Instant Dependency Modeling: Traffic node correlations depend on both flow and speed. Using only one metric causes incomplete or biased results. We use flow sequence F and speed sequence V from the embedding layer. For any node pair i,j:

Calculate the Pearson correlation coefficient $r_F(i,j)$ of the traffic flow sequences to reflect the correlation of congestion levels;Calculate the Spearman rank correlation coefficient $r_V(i,j)$ of the speed sequences to reduce the interference of outliers [16] and make the correlation judgment more stable;

Obtain the instant dependency coefficient via weighted fusion, expressed as:

$$r_{ij} = 0.5 \cdot r_F(i,j) + 0.5 \cdot r_V(i,j) \tag{3}$$

Congestion Propagation Delay Effect Modeling: In real-world traffic scenarios, traffic congestion does not spread instantaneously, and congestion on downstream road segments lags behind that on upstream ones—a characteristic that cannot be captured by the instant dependency matrix alone [14]. We set a delay step set $\tau \in 1,2,3$ (corresponding to 5 - 15 minutes, which is consistent with the actual speed of congestion propagation), calculate the cross-correlation coefficient $r_\tau(i,j)$ for node pair (i,j) under each delay step τ, and take the maximum value as the delay dependency coefficient, expressed as:

$$r_{\tau,max}(i,j) = \max_{\tau \in 1,2,3} r_\tau(i,j) \tag{4}$$

We construct a delay dependency matrix $A_{delay} \in \mathbb{R}^{N\times N}$, which is fused with the instant dependency matrix to yield an initial dynamic matrix, expressed as:

$$A_{pre} = 0.6 \cdot A_{inst} + 0.4 \cdot A_{delay} \tag{5}$$

The weights are determined via tuning on the validation set, which prioritizes the dominant role of instant correlations while incorporating delay correlations, thus making the matrix more consistent with the propagation laws of real-world traffic flow.

Adaptive Threshold Filtering Based on Traffic Density: The initial matrix contains a large number of weak correlations (e.g., faint correlations among long-distance nodes), which will increase the computational overhead of subsequent modules without practical effects and thus requires dynamic filtering according to traffic states. We define the traffic density of the current time window as $D = \frac{1}{N}\sum_{i=1}^{N} F(t,i)$, and generate an adaptive threshold θ through nonlinear mapping, expressed as:

$$\theta = 0.2 + 0.3 \cdot exp\left(-\frac{D - D_{avg}}{D_{std}}\right) \tag{6}$$

where D_{avg} and D_{std} denote the mean and standard deviation of traffic density across the dataset, with the value range of θ set to 0.2 - 0.5. A higher traffic density (e.g., during peak hours) results in a smaller θ, allowing more dynamically correlated nodes to be included; a lower traffic density (e.g., in the late night) leads to a larger θ, retaining only the core dependencies. The final dynamic adjacency matrix is expressed as:

$$A_{dyn}[i][j] = \begin{cases} A_{pre}[i][j], & A_{pre}[i][j] \geq \theta \\ 0, & A_{pre}[i][j] < \theta \end{cases} \tag{7}$$

This matrix is directly fed into the gated fusion module, providing an accurate spatial correlation foundation for graph propagation.

(3) *Adaptive Gating Fusion Module*

This module serves as the core for balancing static and dynamic correlations. It receives the output from the dynamic adjacency matrix construction module and integrates it with the original static adjacency matrix, resulting in an optimally fused adjacency matrix for the subsequent graph propagation module. By combining both static and dynamic information, the model avoids biases associated with relying on a single matrix, while maintaining computational efficiency without adding extra overhead.

We define a static adjacency matrix $A_{static} \in \mathbb{R}^{N\times N}$, which represents the inherent structural connections within the road network and serves as a stable reference under normal traffic conditions. In contrast, the dynamic adjacency matrix A_{dyn} captures real-time traffic relationships and allows the model to adapt to changing scenarios. To effectively balance the contributions of these two matrices, we introduce a learnable gating weight α, whose values are constrained to the range [0, 1] using a sigmoid function, and is formulated as follows:

$$\alpha = \sigma\left(MLP\left(X_{emb}^{avg}\right)\right) \tag{8}$$

where $X_{emb}^{avg} \in \mathbb{R}^d$ denotes the global mean of embedded features, the global information of embedded features can reflect the overall current traffic state, making the adjustment of gating weights more targeted; the MLP is a two-layer fully connected network that adaptively outputs weights according to the current traffic features.

The final fused adjacency matrix is expressed as:

$$A_{fuse} = \alpha \cdot A_{static} + (1-\alpha) \cdot A_{dyn} \tag{9}$$

During training, α is adaptively adjusted based on the current traffic state: it emphasizes physical topology $\alpha \approx 0.6$ during off-peak hours, while prioritizing dynamic dependencies $\alpha \approx 0.3$ during peak hours or construction periods. This flexible balancing mechanism provides the graph propagation module with a context-sensitive basis for spatial correlations, allowing it to adjust effectively to varying traffic scenarios.

(4) *Graph Propagation and Spatiotemporal Graph*

These two modules process the output of the gated fusion module and serve as the core components for capturing spatiotemporal correlations within the model. The graph propagation module is responsible for aggregating high-order spatial dependencies, while the spatiotemporal attention block captures global spatiotemporal correlations. Working together, these components enable comprehensive spatiotemporal modeling within a lightweight framework:

Graph Propagation Module: Based on the fused adjacency matrix A_{fuse}, it aggregates spatial features of nodes and provides feature inputs containing spatial correlations for the attention module [2] [3]. A simplified graph convolution operation is adopted to avoid complex computations, with the formula expressed as:

$$X_{spatial} = \sum_{k=0}^{2} A_{fuse}^{k} \cdot X_{emb} \tag{10}$$

where k = 0,1,2 corresponds to 0 - 2 hop neighbors, which is sufficient to cover the essential spatial range while avoiding feature confusion caused by excessive propagation. A_{fuse}^{k}

represents the k-th power of the fused adjacency matrix, allowing the model to capture higher-order spatial dependencies among nodes. $X_{spatial} \in \mathbb{R}^{T\times N\times d}$ is the aggregated spatial feature, which deeply integrates spatial correlations with the embedded features and provides a strong foundation for subsequent spatiotemporal modeling within the attention block.

Spatiotemporal Attention Module: While maintaining the lightweight architecture of the original model, this module employs a single-layer attention mechanism and modifies its computational approach to enhance adaptability to dynamic features [4] [15]. The input consists of spatial features $X_{spatial}$ produced by graph propagation, and the module refines feature representations by capturing global spatiotemporal correlations. The first step in the process involves generating queries Q, keys K, and values V:

$$Q = X_{spatial} \cdot W_Q, K = X_{spatial} \cdot W_K, V = X_{spatial} \cdot W_V \quad (11)$$

where $W_Q, W_K, W_V \in \mathbb{R}^{d\times d}$ denote attention weight matrices.

Spatial and temporal attention are calculated separately:

$$Attn_{spatial} = \frac{Q \cdot K^T}{\sqrt{d}} \odot A_{fuse}, Attn_{temporal} = \frac{Q^T \cdot K}{\sqrt{d}} \quad (12)$$

where $Attn_{spatial} \in \mathbb{R}^{T\times N\times N}$ incorporates the fused adjacency matrix as a mask to strengthen effective spatial correlations and filter redundant information; $Attn_{temporal} \in \mathbb{R}^{N\times T\times T}$ captures the dependency relationships in the temporal dimension. The final attention output is given by:

$$X_{attn} = Attn_{spatial} \cdot V + Attn_{temporal}^T \cdot V \quad (13)$$

(5) *Prediction Layer*

The prediction layer takes outputs from the spatiotemporal attention module, converting high-dimensional spatiotemporal features into final traffic flow predictions while reducing overfitting.A fully connected layer first compresses feature dimensions to cut parameters and retain key information:

$$X_{compress} = ReLU(X_{attn} \cdot W_{compress} + b_{compress}) \quad (14)$$

Subsequently, the future traffic flow for the subsequent $S = 12$ time steps is predicted through the output layer:

$$X_{pred} = X_{compress} \cdot W_{out} + b_{out} \quad (15)$$

where $X_{pred} \in R^{S\times N\times 1}$ denotes the prediction result, with $W_{out} \in R^{d\times 1}$ and $b_{out} \in R^1$ being the output layer parameters. The entire prediction layer consists of only two fully connected layers, which ensures the model maintains its lightweight characteristics while directly outputting the final prediction results, thus forming a complete closed loop from data input to prediction output.

IV. EXPERIMENTS

A. DataSets

We conduct traffic flow prediction experiments on three real-world public datasets: METR-LA, PEMS04, and PEMS08 [11] [12] [17]. Their detailed information, including the number of nodes, number of edges, time steps, and time span, is presented in TABLE I. .

TABLE I. DATASET DESCRIPTION

Datasets	Nodes	Edges	Time steps	Time span
METR-LA	207	1596	34272	03/01/2012-06/30/2012
PEMS04	307	340	16992	01/01/2018-02/28/2018
PEMS08	170	295	17856	07/01/2016-18/31/2016

B. Experiment Settings

Datasets are split into train/validation/test sets with a ratio of 6:2:2. The input time step *T = 12* (60 minutes) and prediction step *S = 12* (next 60 minutes). Embedding dimension *d = 32*; the gated fusion module uses a 2-layer MLP with hidden dimension 64. Models are optimized by Adam with learning rate 0.001, batch size 32, and max epoch 200. Early stopping with patience 10 is used to prevent overfitting. Performance is evaluated by MAE, RMSE, and MAPE.

C. Baseline Methods

Nine traffic flow prediction models are selected as baselines for a comparative analysis of the performance of ADG-Former:

(1) HA: Predicts future values based on the historical average.

(2) ARIMA [1]: A stationary time-series forecasting model based on differencing and moving averages.

(3) VAR [1]: A model for multivariate time-series modeling.

(4) LSTM [18]: An improved variant of RNN designed to capture long-term dependencies in time series.

(5) GRU [18]: A simplified variant of RNN with lower structural complexity than LSTM.

(6) STGCN [10]: Employs graph convolution along both temporal and spatial dimensions to handle spatiotemporal traffic data.

(7) ASTGCN [19]: Integrates a self-attention mechanism into a graph convolutional network to capture spatioemporal features and dynamic dependencies.

(8) PDFormer [14]: Incorporates a propagation-delay-ware mechanism to model time-lag characteristics of traffic state propagation.

(9) STGformer [4]: The original model of this study, featuring an efficient spatiotemporal graph Transformer architecture.

TABLE II. COMPARISON OF PREDICTION ACCURACY WITH BASELINE MODEL

Data	Metric	Methods									
		HA	ARIMA	VAR	LSTM	GRU	STGCN	ASTGCNN	PDFormer	STGformer	ADG-Former
METR-LA	MAE	35.28	31.64	32.89	28.76	27.92	24.58	21.03	19.56	18.42	**17.05**
	RMSE	52.67	66.89	50.34	44.28	43.51	37.64	31.75	30.18	**26.95**	27.02
	MAPE(%)	20.78	18.92	19.36	16.84	16.21	14.53	12.46	11.59	10.87	**9.93**
PEMS04	MAE	36.76	32.11	33.76	29.45	28.65	25.15	21.80	20.23	19.15	**17.78**
	RMSE	54.14	68.13	51.73	45.82	45.11	38.29	32.82	31.67	30.02	**28.57**
	MAPE(%)	22.38	19.45	19.92	17.36	16.72	15.01	13.02	12.15	11.42	**10.56**
PEMS08	MAE	29.52	24.04	21.41	23.18	22.20	18.88	16.63	15.37	14.26	**13.08**
	RMSE	44.03	43.30	31.21	36.96	35.95	27.87	25.27	**21.88**	22.84	21.97
	MAPE(%)	18.62	16.34	15.19	14.87	14.15	12.68	11.05	10.23	9.58	**8.84**

D. Experiments results and analysis

(1) *Prediction Performance Comparison*

To evaluate the performance advantages of the proposed ADG-Former model in traffic flow prediction tasks, we selected several state-of-the-art models as baselines and conducted comparative experiments on three public datasets . The main evaluation metrics include mean absolute error (MAE), root mean square error (RMSE), and mean absolute percentage error (MAPE) in 0.

Traditional statistical models focus only on temporal relations and perform poorly in complex traffic scenarios. Deep learning models outperform them but lack effective spatial modeling. GNNs capture spatial dependencies via graph convolution yet are limited to local patterns. Transformer-based models use self-attention for global dependencies and perform better; PDFormer considers propagation delays but lacks multi-dimensional dynamic correlations. STGformer is efficient but relies on static spatial topology, reducing adaptability.

The overall accuracy comparison above highlights the superior performance of ADG-Former with fixed prediction horizons. However, in real-world traffic forecasting, different applications often require varying prediction intervals. To further evaluate each model's robustness in multi-step prediction tasks, Fig. 3 presents results for different prediction horizons on the PEMS04 dataset.

(2) *Ablation Experiments*

To evaluate the impact of each core component in ADG-Former, ablation studies are performed on the PEMS04 dataset. The tested model variants include: *w/o Dyn*, which

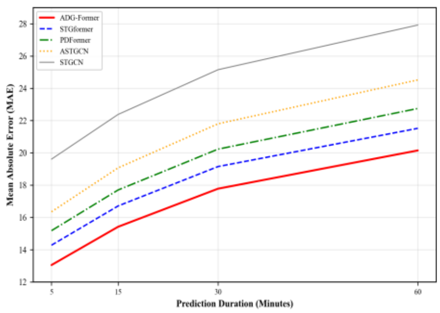

Fig. 3. MAE Comparison of Models on PEMS04.

removes the multidimensional dynamic adjacency matrix; *w/o Gate*, which omits the adaptive gating fusion module; *w/o Delay*, which excludes congestion propagation delay modeling in the adjacency matrix construction; and *w/o Adaptiveθ*, Fig. 4 disables the adaptive threshold mechanism.

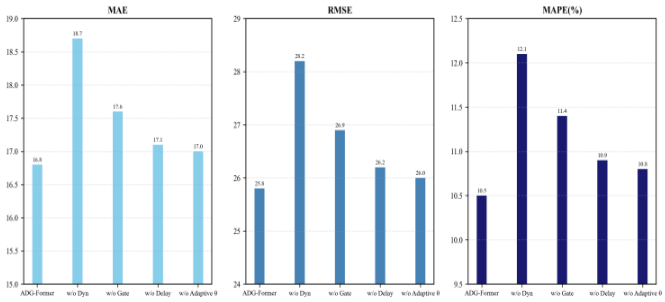

Fig. 4. ADG-Former Ablation: MAE & RMSE & MAPE (30-min).

To better understand how the proposed modules function, we visualize the adaptive threshold and gated fusion components. As illustrated in Fig. 5 (based on PEMS08), the adaptive threshold θ generally decreases as traffic density rises, particularly during peak hours, and increases when traffic is lighter. This pattern aligns with the design goal of retaining more dynamic correlations during periods of heavy traffic, while filtering out redundant information when the traffic flow is low.

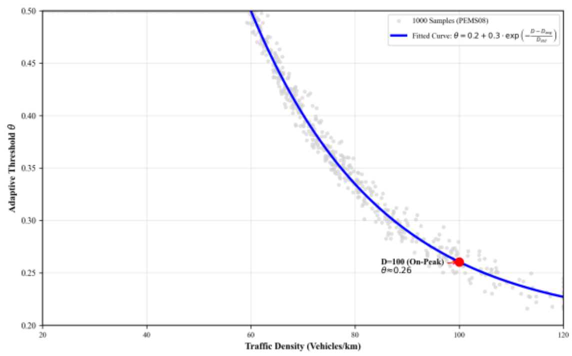

Fig. 5. Relationship between Adaptive Threshold.

Fig. 6 presents results on the METR-LA, where the gating weight α exhibits a strong negative correlation with traffic density ($\alpha \approx 0.3-0.4$ during peak hours and $\alpha \approx 0.5-$

0.6 during off-peak hours), enabling the adaptive balance of static and dynamic information.

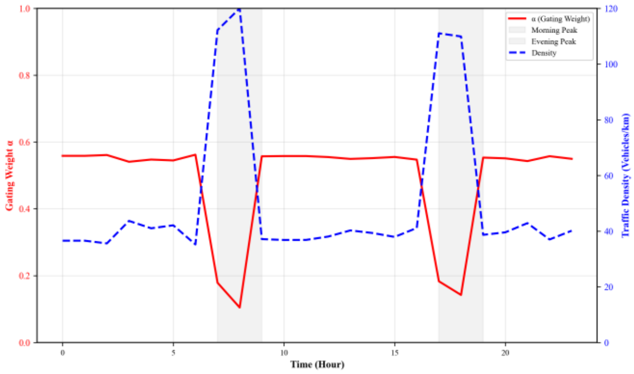

Fig. 6. Diurnal Variation of Gating Weight α and Traffic Density.

V. CONCLUSION

This work addresses the static graph limitation of spatiotemporal graph Transformers in dynamic traffic scenarios. We propose ADG-Former, a lightweight and efficient model for high-precision traffic flow prediction.

Its core innovations include a multi-dimensional adaptive dynamic adjacency matrix (fusing flow–speed correlation and congestion delay) and an adaptive gated fusion module that balances static topology and dynamic dependencies. The model retains STGformer's linear complexity with minimal parameter and speed overhead, ensuring high efficiency.

Experiments on three real-world datasets show ADG-Former outperforms nine baselines on MAE, RMSE, and MAPE. Ablation studies verify the effectiveness of the proposed modules. The model provides a practical solution for traffic management and route planning.

Limitations include insufficient use of external factors. Future work will integrate weather, traffic incidents, and improve scalability for large-scale networks [20].

ACKNOWLEDGMENT

This work was supported by the Beihua University Graduate Innovation Program under Grant [2024]061 (Corresponding author: Chunyang Shi.), the Scientific Research Project of the Department of Education of Jilin Province (No. JJKH20261345KJ).

REFERENCE

[1] B. M. Williams, "Traffic flow forecasting: Comparison of modeling approaches," *Journal of Transportation Engineering*, vol. 129, no. 6, pp. 664–672, 2003.

[2] S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan, "Attention based spatial-temporal graph convolutional networks for traffic flow forecasting," in *Proc. AAAI Conf. Artif. Intell*., 2019, vol. 33, no. 1, pp. 922–929.

[3] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, "Graph wavenet for deep spatial-temporal graph modeling," in *Proc. Int. Joint Conf. Artif. Intell*., 2019, pp. 1907–1913..

[4] H. Wang, et al., "STGformer: Efficient spatiotemporal graph transformer for traffic forecasting," arXiv:2410.00385, 2024..

[5] J. Jiang, C. Han, W. X. Zhao, and X. Zhou, "PDFormer: Propagation delay-aware dynamic long-range transformer for traffic flow prediction," arXiv:2301.07945, 2023.

[6] H. Shi, et al., "STAEformer: Spatial-temporal adaptive encoder transformer for traffic flow prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 1, pp. 111–123, 2023.

[7] Z. Cui, R. Ke, Z. Wang, Y. Chen, and H. Ma, "Dynamic graph convolutional recurrent network for traffic prediction," *ACM Trans. Knowl. Discov. Data*, vol. 17, no. 1, pp. 1–21, 2023,.

[8] Z. Shao, et al., "Decoupled dynamic spatial-temporal graph neural network for traffic forecasting," *Proc. VLDB Endow.*, vol. 15, no. 11, pp. 2733-2746, 2022,.

[9] X. Wang, et al., "DSTAGNN: Dynamic spatial-temporal attention graph neural network for traffic flow prediction," *IEEE Internet Things J.*, vol. 9, no. 23, pp. 23862–23873, 2022.

[10] Y. Yu, B. Yin, and Z. Li, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," in *Proc. Int. Joint Conf. Artif. Intell.*, 2018, pp. 3634–3640.

[11] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," arXiv:1707.01926, 2017.

[12] Caltrans Performance Measurement System (PeMS), "PEMS04, PEMS08 traffic datasets," 2016. [Online].

[13] A. Vaswani, et al., "Attention is all you need," in *Adv. Neural Inf. Process. Syst*., 2017, pp. 5998–6008.

[14] J. Jiang, et al., "PDFormer: Propagation delay-aware dynamic long-range transformer for traffic flow prediction," in *Proc. AAAI Conf. Artif. Intell*., 2023, pp. 922–930.

[15] A. Vaswani, et al., "Attention is all you need," in *Adv. Neural Inf. Process. Syst*., 2017, pp. 5998–6008.

[16] R. A. Fisher, *Statistical Methods for Research Workers*. Hafner Press, 1925.

[17] C. Shahabi, Y. Li, and R. Yu, "METR-LA traffic flow dataset," 2012. [Online].

[18] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[19] S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan, "Attention based spatial-temporal graph convolutional networks for traffic flow forecasting," *IEEE Access,* vol. 7, pp. 100872–100881, 2019.

[20] L. Zhao, et al., "T-GCN: A temporal graph convolutional network for traffic prediction," *IEEE Trans. Intell. Transp. Syst*., vol. 21, no. 9, pp. 3848–3858, 2020,.