Multi-Period Diffusion Generative Graph Recurrent Transformer Network for Traffic Flow Prediction in Vehicular Networks

Yinxin Bao, Qinqin Shen, Yang Cao, Yingyan Hou, Wanxuan Lu, and Quan Shi, *Member, IEEE*

***Abstract*— Real-time and accurate traffic flow prediction enhances the efficiency of vehicular networks and improves traffic management. As vehicular networks become essential in intelligent transportation systems (ITS), precise predictions optimize performance and reduce congestion. However, graph convolutional network (GCN)-based methods rely on preprocessed data, losing critical features and reducing accuracy in complex environments. To address this challenge, we propose a novel Multi-period Diffusion generative Graph Recurrent Transformer Network (MD-GRTN) for traffic flow prediction. MD-GRTN's innovation lies in its enhanced performance under noisy conditions. It consists of three main components: the Multi-period Diffusion Attention Fusion (MDAF) module, the Multi-Graph Recurrent Convolution (MGRC) module, and the Spatial-Temporal Transformer (STFormer) module. The MDAF module combines the Multi-period Diffusion (MD) module and the Multi-head Attention Fusion (MAF) module to enhance historical trend features of multi-period traffic flows. The MGRC module integrates a multi-graph fusion module, a graph convolutional network, and a gated recurrent unit to strengthen spatial features influenced by various factors. Lastly, the STFormer module, comprising spatial and temporal transformer modules, further enhances the global dynamic spatial-temporal features of traffic flow. The key advantage of MD-GRTN lies in its ability to effectively capture multi-period temporal dependencies, global spatial correlations, and latent features in noisy traffic data, significantly improving prediction robustness under real-world conditions. Extensive experiments on the PEMS (03–08) datasets demonstrate that MD-GRTN outperforms state-of-the-art models, achieving average RMSE reductions of 4.7% when compared with STFGCN and PDFormer.**

Received 21 January 2025; revised 24 March 2025 and 27 April 2025; accepted 26 May 2025. This work was supported in part by the National Natural Science Foundation of China under Grant 62476145, in part by the Humanity and Social Science Foundation of Ministry of Education of China under Grant 24YJAZH126, in part by the 6th "333 Talents" Technology Research and Development Talent Foundation of Jiangsu Province, in part by the Transportation Technology and Achievement Transformation Foundation of Jiangsu Province under Grant 2024G01, and in part by the Key Laboratory of Target Cognition and Application Technology under Grant 2023-CXPT-LC-005. The Associate Editor for this article was T. R. Gadekallu. *(Corresponding author: Quan Shi.)*

Yinxin Bao, Yang Cao, and Quan Shi are with the School of Information Science and Technology and the School of Transportation and Civil Engineering, Nantong University, Nantong 226019, China (e-mail: baoyinxin@stmail.ntu.edu.cn; caoyangnt@ntu.edu.cn; sq@ntu.edu.cn).

Qinqin Shen is with the School of Transportation and Civil Engineering, Nantong University, Nantong 226019, China (e-mail: shenqq@ntu.edu.cn).

Yingyan Hou and Wanxuan Lu are with the Target Key Laboratory of Cognition and Application Technology and the Key Laboratory of Network Information System Technology, Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing 100045, China (e-mail: houyy@aircas.ac.cn; luwx@aircas.ac.cn).

Digital Object Identifier 10.1109/TITS.2025.3575586

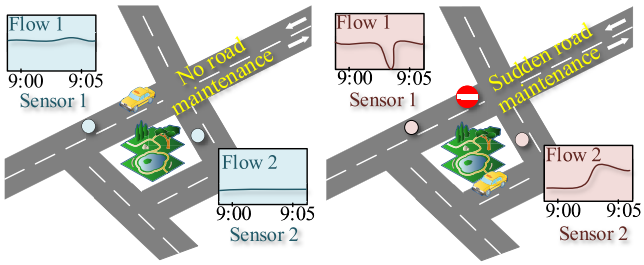
***Index Terms*— Neural network, traffic prediction, denoising diffusion, attention mechanism, graph convolution.**

I. Introduction

RAPID urbanization has exacerbated the imbalance between road capacity and the number of vehicles, leading to increased traffic congestion and pollution. Vehicular networks enable real-time data exchange, significantly improving traffic management. Real-time and accurate traffic flow prediction within these networks provides essential information for downstream applications such as route planning, signal timing optimization, and estimated arrival time, making it a key solution to traffic congestion [1], [2], [3]. However, real-world traffic data are often noisy due to sensor errors, external disturbances, and missing values, which can degrade prediction accuracy. While advancements in data collection, sensor networks, and GPU processing have enhanced prediction models, effectively leveraging noisy data to improve robustness and accuracy in complex traffic environments remains a key research challenge [4], [5]. This paper addresses this challenge by proposing a model designed to extract valuable features from noisy data, ensuring more reliable and adaptable traffic predictions.

Traffic flow prediction methods exploiting learning techniques can be broadly categorized into machine learning and deep learning approaches. Machine learning methods have become the mainstream approach for traffic flow prediction [6], [7]. While simple and straightforward, traditional non-machine learning methods, such as the historical average, grey model [8], and the Autoregressive Integrated Moving Average (ARIMA) model [9], often struggle to adapt to the dynamic changes of complex traffic flows. In contrast, machine learning methods have emerged as a research focus due to their superior data adaptability and nonlinear modeling capabilities. Deep learning techniques, exemplified by Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) [10], [11], can learn deep spatial features and temporal dependencies from vast amounts of data, enhancing prediction accuracy.

The transportation system is dynamically evolving, where traffic flow at a given location is influenced by historical data and neighboring traffic, exhibiting a complex spatial-temporal feature [12]. To more effectively capture this spatial-temporal dependency, a graph-based deep learning model, such as a



(a) No road maintenance (b) Sudden road maintenance

Fig. 1. Different traffic status.

Graph Convolutional Network (GCN), has gained widespread attention in recent years [7], [13]. A GCN performs convolution operations on graph-structured data, effectively extracting the spatial relationships between adjacent nodes, making it well-suited for representing complex road network structures. When combined with a time series model like Long Short-Term Memory (LSTM) network, a GCN can handle spatial dependencies while capturing the dynamic changes in temporal data [14], [16],. Although a GCN offers significant advantages in spatial feature extraction, it still faces limitations when dealing with unstructured data. Recent studies have begun exploring integrating an attention mechanism with GCN [17], [18]. By incorporating an attention mechanism, the model can dynamically focus on important adjacent nodes, improving traffic flow predictions' accuracy and robustness [19]. The inclusion of an attention mechanism not only optimizes the information aggregation process but enhances the model's adaptability to prediction tasks across different traffic environments [20].

Various factors, such as construction, accidents, or weather changes, often affect traffic flow, introducing noise into the data [21], as shown in Fig. 1. Specifically, the figure illustrates that after an emergency maintenance event, the traffic flow at affected nodes exhibits abrupt fluctuations, highlighting the challenge of modeling under noisy conditions. These disturbances are often considered erroneous data and are subsequently removed during data cleaning. However, this process may lead to the loss of critical features, potentially impacting the accuracy of downstream analyses. Existing traffic flow prediction methods based on GCN and attention mechanisms typically require extensive data preprocessing to filter and clean the training dataset, ensuring input data quality and consistency. While this approach improves training efficiency and prediction accuracy, it often leads to an over-reliance on idealized data conditions and overlooks the valuable insights hidden in raw, noisy data [22]. However, real-world traffic data are inherently noisy due to unpredictable disturbances such as sudden road closures or holiday-induced traffic surges. Most existing methods primarily focus on extracting spatial-temporal features while failing to effectively leverage the critical information embedded in noisy data [23], [24]. Despite recent advances, a significant research gap remains in developing models that can directly handle and utilize raw, noisy traffic data without extensive preprocessing. Current methods struggle to maintain predictive accuracy in complex and highly dynamic traffic environments. The key challenge lies in designing a model that not only extracts spatial-temporal features but also adapts to varying noise levels and exploits the underlying patterns within noisy data.

To address the aforementioned challenges, this paper proposes a novel Multi-Period Diffusion Generative Graph Recurrent Transformer Network (MD-GRTN) for traffic flow prediction. Unlike existing methods that often rely on complex preprocessing to denoise input data or fail to fully capture long-range temporal patterns, MD-GRTN is designed to enhance predictive robustness under raw noisy conditions and explicitly model multi-period temporal dependencies. The network is composed of three key modules: the Multi-Period Diffusion Attention Fusion (MDAF) module, the Multi-Graph Recurrent Convolution (MGRC) module, and the Spatial-Temporal Transformer (STFormer) module. The main contributions are as follows:

1) The MDAF module, consisting of the Multi-period Diffusion (MD) module and the Multi-head Attention Fusion (MAF) module, is used to enhance the historical trend features of multi-period traffic flow.

2) The MGRC module, composed of a Multi-Graph Fusion module, GCN, and Gated Recurrent Unit (GRU), is designed to enhance spatial features under the influence of multiple factors.

3) The STFormer module, comprising a Spatial Transformer module and a Temporal Transformer module, is used to further enhance the global dynamic spatial-temporal features of traffic flow.

4) Extensive experiments on five real-world datasets demonstrate that MD-GRTN excels in handling noisy data and outperforms state-of-the-art models in traffic flow prediction.

The remainder of this paper is organized as follows: Section II reviews related works on traffic flow prediction, Section III introduces the preliminaries of traffic flow prediction, Section IV presents the MD-GRTN method, Section V discusses the experimental setup and results, and Section VI concludes the work of MD-GRTN.

II. Related Works

A. Deep Learning Methods

With the rapid development of ITS and vehicular networks, research on traffic flow prediction has gradually shifted from traditional machine learning methods to deep learning-based approaches in recent years [18], [26], [27], [28],. Traditional machine learning methods, such as grey prediction models, gained popularity in the early stages due to their simplicity and intuitive nature. However, they have shown significant limitations when dealing with the complex, nonlinear nature of traffic data, particularly in dynamic environments enabled by vehicular networks, where real-time data from connected vehicles and infrastructure add further complexity to prediction tasks [29]. In contrast, deep learning methods, particularly CNNs and RNNs, have become key research areas due to their powerful spatial and temporal feature extraction capabilities, which are especially beneficial in the context of vehicular networks where both spatial and temporal dynamics play crucial

roles in traffic flow prediction [30], [31]. To further enhance the performance of these models, researchers have explored hybrid models that combine the spatial analysis capabilities of CNNs with the temporal sequence processing strengths of RNNs, effectively addressing the complexity of traffic flow prediction in vehicular environments [32]. However, deep models often face technical challenges such as vanishing or exploding gradients, especially when processing large-scale data from vehicular networks, prompting researchers to introduce techniques like residual connections to stabilize the training process and improve model robustness [33],.

B. GCN-Based Methods

Traditional Euclidean spatial models efficiently handle regular data but have high limitations when facing complex road network structures [35], [36]. GCN and its derivatives are widely used for traffic flow prediction due to their advantages in handling non-Euclidean spatial data. Diffusion Convolutional Recurrent Neural Network (DCRNN) effectively integrates spatial-temporal information by utilizing the bi-directional stochastic wandering mechanism of the graph to extract spatial features and capture temporal dependencies through an encoder-decoder architecture [37]. In addition, the spatial-temporal synchronous graph convolutional network (STSGCN) [38] and spatial-temporal fusion graph neural network (STFGNN) [39] improve the model's adaptability to the dynamics of the traffic data and prediction accuracy through innovative network architectures and modeling strategies. The Automated Dilated Spatial-Temporal Synchronous Graph Network (Auto-DSTSGN) further enhances the generalization and flexibility of the model for different traffic scenarios by automating the construction of spatial-temporal graphs [40]. Li et al. proposed a novel Spatial-Temporal Fusion Graph Convolutional Network (STFGCN) for accurate traffic prediction by extracting multiscale temporal dependencies from multiple semantic environments and constructing a temporal feature based on dynamic adaptive graphs to model spatial dependencies [41].

C. Attention-Based Methods

Although GCN-based methods have made significant progress in improving traffic prediction accuracy, it is still a challenge to accurately capture and interpret the complex spatial-temporal correlations in traffic networks. To address this problem, researchers have introduced attention-based GCNs such as Attention based Spatial-Temporal Graph Convolutional Network (ASTGCN) [42] and Attention based Spatial-Temporal Graph Neural Network (ASTGNN) [43]. These models enhance the perception of key traffic features by integrating novel self-attention mechanisms and optimizing the combined processing of spatial-temporal features. The Propagation Delay-aware dynamic long-range transformer (PDFormer) proposed by Jiang et al. further extracts the spatial self-attention module and specific graph mask matrix by combining the spatial self-attention module with the introduction of a feature transformation module for the traffic delays and further extracts the complex temporal correlations in the traffic flow [44]. Building upon attention mechanisms, recent advancements in diffusion-based models have further enhanced the robustness of traffic prediction by effectively handling noise and capturing complex traffic dynamics. Diffusion models, when combined with spatial-temporal graph structures, leverage denoising mechanisms to refine feature representations and improve predictive accuracy. Zhu et al. introduced the Spatial-Temporal Diffusion Probabilistic Model for Trajectory Generation (DiffTraj) [45], which applies a denoising diffusion process to reconstruct geographic trajectories from white noise, thereby strengthening model resilience against disturbances. Wen et al. proposed Probabilistic Spatio-Temporal Graph Forecasting with Denoising Diffusion Models (DiffSTG) [46], which extends diffusion-based probabilistic modeling to traffic data, effectively capturing evolving dependencies while incorporating uncertainty estimation. Additionally, Shao et al. developed the Decoupled Dynamic Spatial-Temporal Graph Neural Network (D2STGNN) [47], which separates traffic data into diffuse and intrinsic components, applying a residual decomposition mechanism alongside dynamic graph learning to better model changing traffic patterns and mitigate the impact of noise. These advancements highlight the increasing role of diffusion models in spatial-temporal learning, offering a more robust framework for accurate traffic flow prediction in real-world scenarios.

By analyzing the existing works, the following challenges remain in traffic flow prediction: (1) While current methods effectively capture spatial-temporal dependencies, they struggle to simultaneously model multi-periodic temporal variations and global spatial correlations in traffic data. (2) Most models rely on pre-processed data, overlooking valuable latent features within noisy data [48], which may limit their ability to extract traffic flow trends in complex environments. To address these issues, this paper proposes MD-GRTN, which enhances prediction accuracy by effectively leveraging noisy data, capturing multi-period time-varying patterns, and integrating global spatial dependencies.

III. Preliminaries

A. Problem Definition

As shown in Fig. 2, the actual road network is defined as $\boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{E}, \boldsymbol{A})$, where $\boldsymbol{V}$ represents the set of N sensors in the road network, $\boldsymbol{E}$ represents the set of edges between sensors, and $\boldsymbol{A}$ represents the connectivity matrix between sensors. The traffic flow prediction problem is defined as predicting the future output $\boldsymbol{Y} \in \mathbb{R}^{N \times T_{\mathrm{f}}}$ based on the historical input sequence $\boldsymbol{X} \in \mathbb{R}^{N \times T_{\mathrm{h}}}$. The data collection interval is defined as $t_{\min}$ minutes, then the noisy inputs of neighborhood sequence, time sequence, and day sequence are $\boldsymbol{X}_{\mathrm{RecN}}$, $\boldsymbol{X}_{\mathrm{HourN}}$, $\boldsymbol{X}_{\mathrm{DayN}}$, respectively, and the sequence lengths of all three inputs are T_{H}. The noisy traffic flow is obtained after outlier repair and data interpolation to get the noise-free traffic flow, and the three inputs are $\hat{\boldsymbol{X}}_{\mathrm{Rec}}$, $\hat{\boldsymbol{X}}_{\mathrm{Hour}}$, and $\hat{\boldsymbol{X}}_{\mathrm{Day}}$, respectively. The traffic flow prediction problem is then defined as:

$$\boldsymbol{Y}_{\mathrm{Feat}}^{(t+1):(t+T_{\mathrm{f}})} = f_{\mathrm{model}}\left[\boldsymbol{A}; \left(\boldsymbol{X}_{\mathrm{RecN}}, \boldsymbol{X}_{\mathrm{HourN}}, \boldsymbol{X}_{\mathrm{DayN}}\right)\right] \quad (1)$$

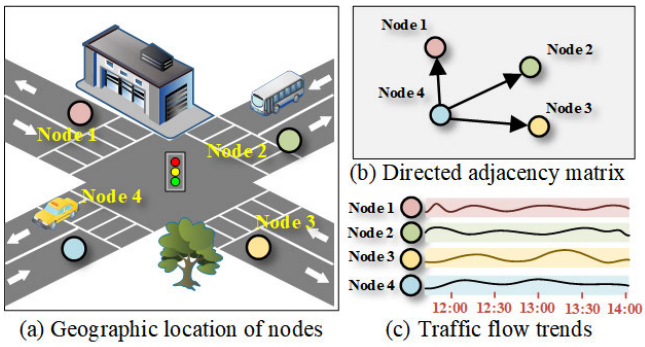


Fig. 2. Definition of the graph structure.

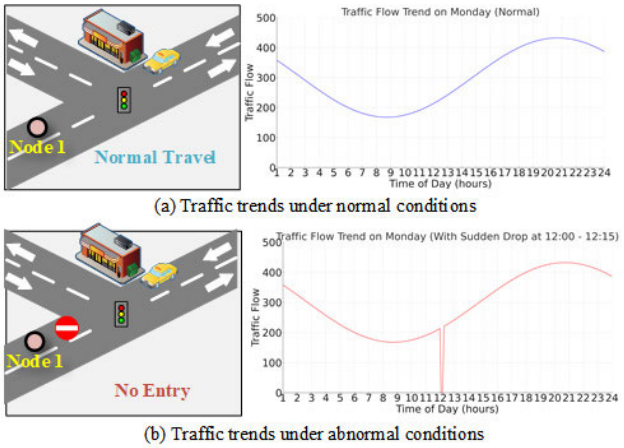


Fig. 3. Effective features in noisy data.

where $\boldsymbol{Y}_{\text{Feat}}^{(t+1):(t+T_{\text{f}})}$ is the output of the prediction model and f_{model} is the prediction model.

B. Effective Features in Noisy Data

In traffic flow prediction research, handling missing data is the key to improving prediction accuracy. However, traditional preprocessing methods such as interpolation, smoothing, and statistical modeling often lose important features in the denoising process [49]. Some missing traffic situations are not noise but are caused by specific events such as road maintenance or traffic control. Fig. 3(a) shows the traffic variation under normal conditions, exhibiting relatively stable flow patterns. In contrast, Fig. 3(b) reflects a sudden drop in traffic volume caused by road maintenance. The figure highlights that when unexpected events occur, such as emergency repairs, the traffic flow can experience abrupt disruptions at affected locations. Traditional methods that fail to recognize these specific events may mistakenly treat valid features as noisy data for processing. For example, suppose smoothing or interpolation is used to fill in the missing values of the traffic plunge triggered by road maintenance. In that case, the surface anomalies can be removed, but the essential feature data brought by the event cannot be retained, resulting in the loss of key information. This paper focuses on mining the intrinsic value of noisy data to improve the performance of traffic flow prediction models by identifying and retaining these feature data caused by external events, which not only avoids the loss of information caused by overgeneralized denoising but also more accurately responds to the various changes in the complex traffic environment, thus improving the model prediction accuracy.

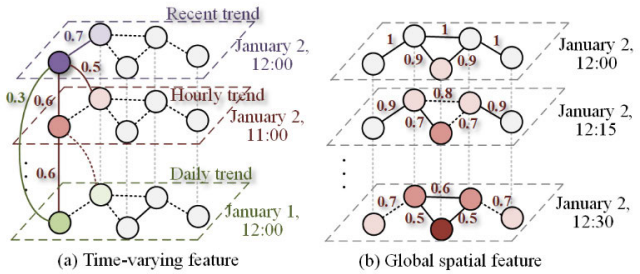


Fig. 4. Time-varying and glbal spatial features.

C. Time-Varying and Global Spatial Features of Traffic Flow

Traffic flow has significant time-varying and global spatial features, as shown in Fig. 4. Fig. 4(a) demonstrates the time-varying features of traffic flow, which is jointly influenced by the recent, hourly, and daily trends. There is a correlation between the traffic flow on different dates at the same moment and the neighboring hours on the same date. Fig. 4(b) shows the global spatial features of traffic flow, when traffic congestion occurs at a node, this congestion state will be transmitted to the global node, reducing the road capacity. Therefore, accurately extracting the time-varying and global spatial features of traffic flow is the key to improving the accuracy of traffic flow prediction.

IV. The Proposed Method: MD-GRTN

This paper introduces the MD-GRTN model to improve traffic flow prediction under noisy conditions. The model integrates three core modules: MDAF, MGRC, and STFormer. The MDAF module, composed of the MD and MAF components, captures multi-period traffic flow trends by enhancing historical feature extraction. The MGRC module combines Multi-Graph Fusion, Graph Convolutional Networks, and Gated Recurrent Units to effectively capture spatial dependencies and adapt to the influence of multiple factors. The STFormer module, consisting of Spatial Transformer and Temporal Transformer components, refines global dynamic spatial-temporal features, enabling robust predictions. Together, these modules synergistically enhance the model's ability to learn and predict traffic flow patterns in complex environments, as illustrated in Fig. 5.

A. Multi-Period Diffusion Attention Fusion (MDAF) Module

The MDAF module consists of Multi-period Diffusion (MD) module and Multi-head Attention Fusion (MAF) module for enhancing the historical trend features of multi-period traffic flow as shown in Fig. 5. The MD module is used to reduce the multi-period noisy inputs $\boldsymbol{X}_{\text{RecN}}$, $\boldsymbol{X}_{\text{HourN}}$, $\boldsymbol{X}_{\text{DayN}}$ to be reduced to multi-cycle noise-free inputs $\hat{\boldsymbol{X}}_{\text{Rec}}$, $\hat{\boldsymbol{X}}_{\text{Hour}}$, $\hat{\boldsymbol{X}}_{\text{Day}}$. for each cycle, the MD module consists of forward process and backward process.

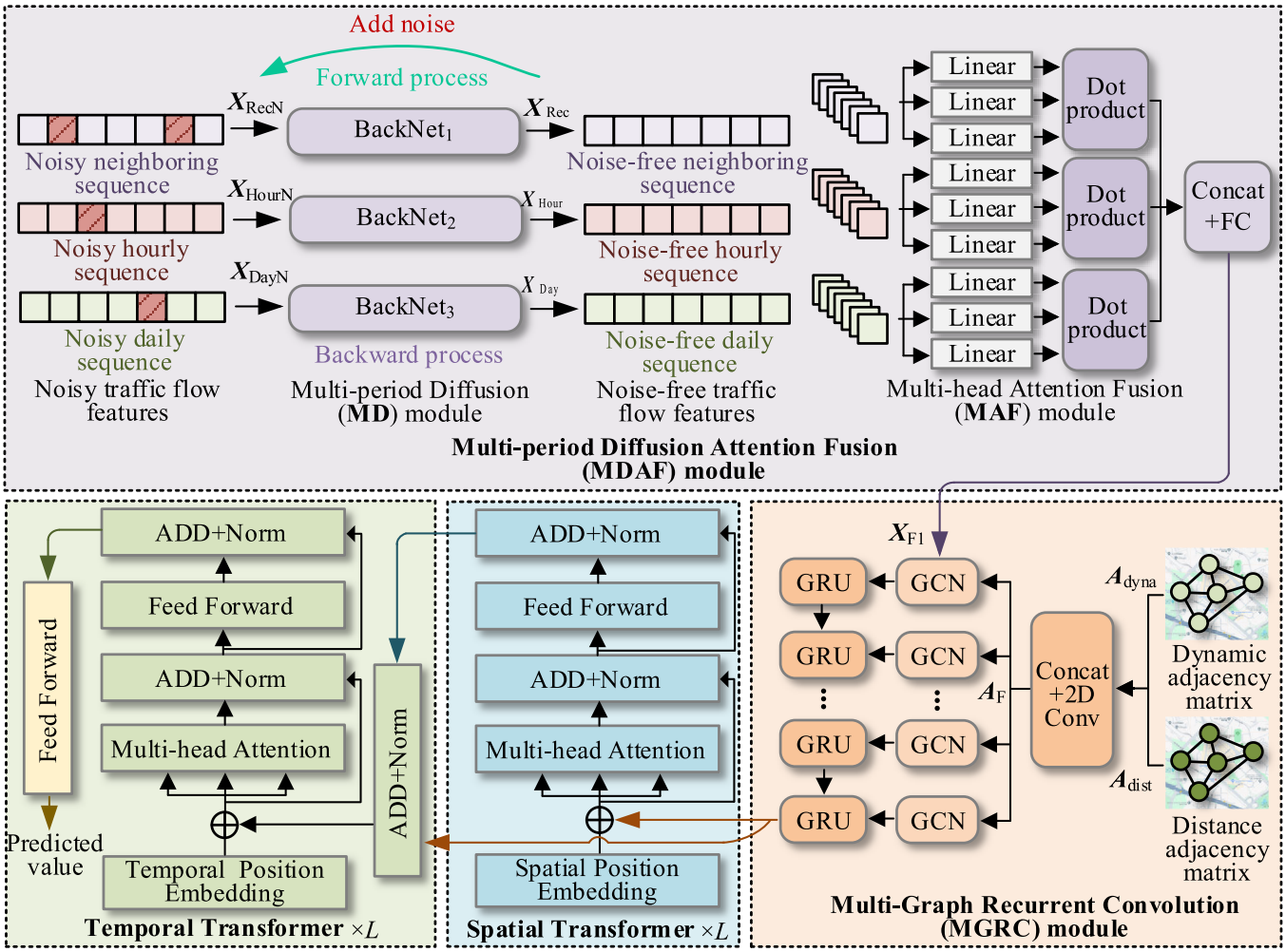


Fig. 5. Structure of MD-GRTN.

The forward process uses a Markov chain for modeling the gradual noisiness of the data, where the raw data $\boldsymbol{X}_0$ is progressively injected with noise until a predefined noise level is reached. The forward process is defined as:

$$\boldsymbol{X}_{t+1} = \sqrt{1-\beta_t}\boldsymbol{X}_t + \sqrt{\beta}_t\epsilon_t \tag{2}$$

where $\boldsymbol{X}_t$ is the state of the data at step t; β_t is the variance scheduling parameter that controls the amount of noise added at each step, usually set to a small positive number; and ϵ_t is the noise drawn from a standard normal distribution.

The backward process is the process of recovering the original data from the noisy data and can be considered as the inverse process of the forward process. The backward process is defined as:

$$\boldsymbol{X}_t = \frac{1}{\sqrt{1-\beta_t}}\left(\boldsymbol{X}_{t+1} - \frac{\beta_t}{\sqrt{1-\beta_t}}\epsilon_\theta\left(\boldsymbol{X}_{t+1}, t\right)\right) \tag{3}$$

where $\epsilon_\theta\left(\boldsymbol{X}_{t+1}, t\right)$ is the noise estimated by the network, this paper is based on the U-Net network for parameter learning [50], aiming to infer the noise added at step t from $\boldsymbol{X}_{t+1}$. The input is the noisy traffic flow, and the output is the noise-free traffic flow.

For each cycle with noisy input $\boldsymbol{X}_k$ (where k denotes the cycle type, i.e., RecN, HourN, and DayN), the backward procedure in the MD module is used to reduce to get the noiseless output, defined as:

$$\hat{\boldsymbol{X}}_k = \text{BackNet}_k\left(\boldsymbol{X}_k\right) \tag{4}$$

where BackNet_k is the MD module backward process for the k-th cycle input and $\hat{\boldsymbol{X}}_k$ is the cycle input after denoising. For each sequence input, the $BackNet_k$ is shown in Equation (3).

The denoised three periodic features $\hat{\boldsymbol{X}}_{\text{Rec}}$, $\hat{\boldsymbol{X}}_{\text{Hour}}$, and $\hat{\boldsymbol{X}}_{\text{Day}}$ are further fused by the MAF module to obtain the periodic fusion output. A linear transformation is performed on each denoised periodic feature input $\hat{\boldsymbol{X}}_k$ to match the dimension:

$$\boldsymbol{Q}_k = \boldsymbol{W}_k^Q\hat{\boldsymbol{X}}_k \tag{5}$$

$$\boldsymbol{K}_k = \boldsymbol{W}_k^K\hat{\boldsymbol{X}}_k \tag{6}$$

$$\boldsymbol{V}_k = \boldsymbol{W}_k^V\hat{\boldsymbol{X}}_k \tag{7}$$

where $\boldsymbol{W}_k^Q$, $\boldsymbol{W}_k^K$ and $\boldsymbol{W}_k^V$ are the learnable matrices of the k-th denoising cycle features which are used to generate the query, key and value matrices respectively. The attention score

is further computed using scaled dot product as follows:

$$\text{Attention}_k\left(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}\right) = f_{\text{SOFT}}\left(\frac{\boldsymbol{Q}_k \boldsymbol{K}_k^T}{\sqrt{d}_j}\right)\boldsymbol{V}_k \tag{8}$$

where Attention_k is the attention score of the k-th denoising cycle feature, d_j is the scaling factor, and f_{SOFT} is the SoftMax activation function.

The three denoising cycle features get output as $\boldsymbol{head}_{\text{Rec}}$, $\boldsymbol{head}_{\text{Hour}}$ and $\boldsymbol{head}_{\text{Day}}$ after self-attention mechanism and after Concat splicing operation as follows:

$$\boldsymbol{X}_{\text{F1}} = \text{Concat}\left(\boldsymbol{head}_{\text{Rec}}, \boldsymbol{head}_{\text{Hour}}, \boldsymbol{head}_{\text{Day}}\right)\boldsymbol{W}_{\text{MH}} \tag{9}$$

where $\boldsymbol{X}_{\text{F1}} \in \mathbb{R}^{N\times T_{\text{h}}}$ is the output of the MDAF module and $\boldsymbol{W}_{\text{MH}}$ is the output transformation matrix.

B. Multi-Graph Recurrent Convolution (MGRC) Module

The MGRC module consists of a multi-graph fusion module, graph convolution networks and gated recurrent units for enhancing spatial features under the influence of multiple factors, as shown in Fig. 5. The traditional method uses the adjacency matrix to describe the $\boldsymbol{A} \in \mathbb{R}^{N\times N}$ node relationship, which ignores the dynamic change of the flow between nodes. In this paper, two feature vectors $\boldsymbol{E}_1 \in \mathbb{R}^{N\times 1}$ and $\boldsymbol{E}_2 \in \mathbb{R}^{N\times 1}$ for describing the nodes are constructed, and the dynamic neighbor matrix of nodes is obtained as:

$$\boldsymbol{A}_{\text{dyna}} = f_{\text{SOFT}}\left(f_{\text{RELU}}\left(\boldsymbol{E}_1 \boldsymbol{E}_2^{Tr}\right)\right) \tag{10}$$

where $\boldsymbol{A}_{\text{dyna}} \in \mathbb{R}^{N\times N}$ is the dynamic adjacency matrix, f_{RELU} is the ReLU activation function, and Tr is the transpose operation.

The distance between nodes has an important impact on the traffic flow, this paper constructs the distance metric equation as:

$$\boldsymbol{A}_{\text{dist}}\left(i, j\right) = \exp\left(-\frac{dist\left(v_i, v_j\right)^2}{\sigma^2}\right) \tag{11}$$

where $dist(\cdot)$ is the Euclidean distance and σ is the standard deviation of the distance, and finally the correlation matrix $\boldsymbol{A}_{\text{dist}} \in \mathbb{R}^{N\times N}$ of the node distances is obtained.

The multi-graph fusion module consists of a 2D convolution for fusing $\boldsymbol{A}_{\text{dyna}}$ and $\boldsymbol{A}_{\text{dist}}$, defined as:

$$\boldsymbol{A}_{\text{F}} = f_{\text{RELU}}\left(f_{\text{Conv2D}}\left(\text{Concat}\left(\boldsymbol{A}_{\text{dyna}}, \boldsymbol{A}_{\text{dist}}\right), \boldsymbol{W}_{\text{Conv2D}}\right)\right) \tag{12}$$

where $\boldsymbol{W}_{\text{Conv2D}}$ is the weight of 2D convolution, f_{Conv2D} is the 2D convolution operation, and $\boldsymbol{A}_{\text{F}}$ is the multi-graph adjacency matrix. Multi-graph spatial features are extracted using graph convolution network defined as:

$$\boldsymbol{X}' = f_{\text{RELU}}\left(\boldsymbol{A}_{\text{F}} \boldsymbol{X}_{\text{F1}} \boldsymbol{W}_{\text{GCN}}\right) \tag{13}$$

where $\boldsymbol{W}_{\text{GCN}}$ is the weight of the graph convolution for extracting the spatial features of each node given its multi-graph neighborhood. The gated recursive unit is further used to obtain the multi-graph temporal features, update gate $\boldsymbol{z}_t$, reset gate $\boldsymbol{r}_t$, candidate hidden state $\tilde{\boldsymbol{h}}_t$ and hidden state $\boldsymbol{h}_t$ are defined as follows:

$$\begin{aligned}
\boldsymbol{z}_t &= f_{\text{SIG}}\left(\boldsymbol{W}_z \boldsymbol{X}' + \boldsymbol{U}_z \boldsymbol{h}_{t-1} + \boldsymbol{b}_z\right) \\
\boldsymbol{r}_t &= f_{\text{SIG}}\left(\boldsymbol{W}_r \boldsymbol{X}' + \boldsymbol{U}_r \boldsymbol{h}_{t-1} + \boldsymbol{b}_r\right) \\
\tilde{\boldsymbol{h}}_t &= f_{\text{TANH}}\left(\boldsymbol{W}_h \boldsymbol{X}' + \boldsymbol{U}_h\left(\boldsymbol{r}_t \odot \boldsymbol{h}_{t-1}\right) + \boldsymbol{b}_h\right) \\
\boldsymbol{h}_t &= \left(1 - \boldsymbol{z}_t\right) \odot \boldsymbol{h}_{t-1} + \boldsymbol{z}_t \odot \tilde{\boldsymbol{h}}_t
\end{aligned} \tag{14}$$

where f_{SIG} is the Sigmoid activation function, f_{TANH} is the Tanh activation function, $\boldsymbol{W}_z$, $\boldsymbol{W}_r$, $\boldsymbol{W}_h$, $\boldsymbol{U}_z$, $\boldsymbol{U}_r$, $\boldsymbol{U}_h$ are the weight matrices, $\boldsymbol{b}_z$, $\boldsymbol{b}_r$, $\boldsymbol{b}_h$ are the bias vectors, and $\odot$ is the Hadamard product. The final output of the MGRC module is $\boldsymbol{X}_{\text{F2}} \in \mathbb{R}^{N\times T_{\text{h}}}$.

C. Spatial-Temporal Transformer (STFormer) Module

The STFormer module consists of spatial Transformer modules and temporal Transformer modules for further enhancing the global dynamic spatial-temporal features of the traffic flow. The STFormer module consists of L spatial Transformer modules and L temporal Transformer modules, and the inputs are the final outputs $\boldsymbol{X}_{\text{F2}}$ of the MGRC.

For the spatial Transformer module at layer l, given an input of $\boldsymbol{X}_{\text{ST}}^{l-1}$. spatial position information is first captured by adding spatial position coding to the input features through the spatial position embedding module. The adjacency matrix A is used to weight the spatial position information and the operation is defined as:

$$\boldsymbol{X}_{\text{S1}}^{l} = \boldsymbol{X}_{\text{ST}}^{l-1} + \boldsymbol{A}\boldsymbol{W}_{\text{SPE}}^{l} \tag{15}$$

where $\boldsymbol{W}_{\text{SPE}}^{l}$ is the weight of the spatial position embedding module and $\boldsymbol{X}_{\text{S1}}^{l}$ is the output after spatial position embedding.

The spatial Transformer module multi-head attention operation of layer l is defined as f_{SMHA}^{l}, and $\boldsymbol{X}_{\text{S1}}^{l}$ gets the output after going through the multi-head attention operation:

$$\boldsymbol{X}_{\text{S2}}^{l} = f_{\text{SMHA}}^{l}\left(\boldsymbol{X}_{\text{S1}}^{l}\right) \tag{16}$$

Add and normalize the output of the multi-attention to get the output:

$$\boldsymbol{X}_{\text{S3}}^{l} = f_{\text{LN}}^{l}\left(\boldsymbol{X}_{\text{S2}}^{l} + \boldsymbol{X}_{\text{S1}}^{l}\right) \tag{17}$$

where f_{LN}^{l} is the layer normalization operation. The features are further processed by feed forward neural network to get the output:

$$\boldsymbol{X}_{\text{S4}}^{l} = f_{\text{FC}}^{l}\left(\boldsymbol{X}_{\text{S3}}^{l}\right) \tag{18}$$

where f_{FC}^{l} is the feed forward neural network. Add and normalize again to get the final output of the spatial Transformer module for the l-th layer:

$$\boldsymbol{Y}_{\text{S}}^{l} = f_{\text{LN}}^{l}\left(\boldsymbol{X}_{\text{S4}}^{l} + f_{\text{FC}}^{l}\left(\boldsymbol{X}_{\text{S4}}^{l}\right)\right) \tag{19}$$

For the temporal Transformer module of layer l, the input is $\boldsymbol{X}_{\text{ST}}^{l-1}$ and the output of the spatial Transformer module of layer l is $\boldsymbol{Y}_{\text{S}}^{l}$. Add and normalize to obtain:

$$\boldsymbol{X}_{\text{T1}}^{l} = f_{\text{LN}}^{l}\left(\boldsymbol{X}_{\text{ST}}^{l-1} + \boldsymbol{Y}_{\text{S}}^{l}\right) \tag{20}$$

Temporal position encoding is added to the input feature $\boldsymbol{X}_{\mathrm{T1}}^{l}$ to capture temporal position information. The temporal position encoding is categorized into hourly encoding $\boldsymbol{E}_{\mathrm{hour}} \in \mathbb{R}^{1\times T_{\mathrm{h}}}$, daily encoding $\boldsymbol{E}_{\mathrm{day}} \in \mathbb{R}^{1\times T_{\mathrm{h}}}$, and weekly encoding $\boldsymbol{E}_{\mathrm{week}} \in \mathbb{R}^{1\times T_{\mathrm{h}}}$. $\boldsymbol{E}_{\mathrm{hour}}(i) \in [1,60]$, $\boldsymbol{E}_{\mathrm{day}}(i) \in [1,24]$, $\boldsymbol{E}_{\mathrm{week}}(i) \in [1,7]$. After embedding the input feature $\boldsymbol{X}_{\mathrm{T1}}^{l}$ with temporal position, we get:

$$\boldsymbol{X}_{\mathrm{T2}}^{l} = \boldsymbol{X}_{\mathrm{T1}}^{l} + \boldsymbol{W}_{\mathrm{hour}}^{l}\boldsymbol{E}_{\mathrm{hour}} + \boldsymbol{W}_{\mathrm{day}}^{l}\boldsymbol{E}_{\mathrm{day}} + \boldsymbol{W}_{\mathrm{week}}^{l}\boldsymbol{E}_{\mathrm{week}} \quad (21)$$

where $\boldsymbol{W}_{\mathrm{hour}}^{l} \in \mathbb{R}^{N\times 1}$, $\boldsymbol{W}_{\mathrm{Hour}}^{l} \in \mathbb{R}^{N\times 1}$ and $\boldsymbol{W}_{\mathrm{Day}}^{l} \in \mathbb{R}^{N\times 1}$ are the weights of the temporal location embedding module.

The multi-head attention operation in the temporal Transformer module of layer l is defined as f_{TMHA}^{l}, and $\boldsymbol{X}_{\mathrm{T2}}^{l}$ gets the output after going through the multi-head attention operation:

$$\boldsymbol{X}_{\mathrm{T3}}^{l} = f_{\mathrm{TMHA}}^{l}\left(\boldsymbol{X}_{\mathrm{T2}}^{l}\right) \quad (22)$$

Add and normalize the output of the multi-attention to give the output:

$$\boldsymbol{X}_{\mathrm{T3}}^{l} = f_{\mathrm{LN}}^{l}\left(\boldsymbol{X}_{\mathrm{T3}}^{l} + \boldsymbol{X}_{\mathrm{T2}}^{l}\right) \quad (23)$$

The features are further processed by feed forward neural network to get the output:

$$\boldsymbol{X}_{\mathrm{T4}}^{l} = f_{\mathrm{FC}}^{l}\left(\boldsymbol{X}_{\mathrm{T3}}^{l}\right) \quad (24)$$

Add and normalize again to get the output:

$$\boldsymbol{X}_{\mathrm{ST}}^{l} = f_{\mathrm{LN}}^{l}\left(\boldsymbol{X}_{\mathrm{T4}}^{l} + f_{\mathrm{FC}}^{l}\left(\boldsymbol{X}_{\mathrm{T4}}^{l}\right)\right) \quad (25)$$

Ultimately, the output of the STFormer module is passed through a feed forward neural network to obtain the final predicted value, defined as:

$$\boldsymbol{Y}_{\mathrm{Feat}} = f_{\mathrm{FC}}\left(\boldsymbol{Y}_{\mathrm{ST}}\right) \quad (26)$$

where $\boldsymbol{Y}_{\mathrm{ST}}$ is the final output of the STFormer module.

D. MD-GRTN Training Process

The training phase of MD-GRTN is divided into two parts: pre-training and main training phase. In the pre-training phase, the MD module in the MDAF module is trained and the optimal weights are saved to be loaded in the backward network. In the main training phase, the weights in the MD-GRTN other than the MD module are iteratively updated. The Huber loss function is widely used in regression tasks due to its robustness to outliers. It seamlessly combines MSE for small residuals, ensuring smooth optimization, and MAE for large residuals, reducing sensitivity to extreme values. This balance enhances model stability, making it particularly effective for noisy data scenarios such as traffic flow prediction. In the main training phase, the Huber loss function is used [51], defined as:

$$\mathrm{Huber}\left(\boldsymbol{Y}_{\mathrm{R}}, \boldsymbol{Y}_{\mathrm{P}}\right) = \begin{cases} \frac{1}{2}\left(\boldsymbol{Y}_{\mathrm{R}} - \boldsymbol{Y}_{\mathrm{P}}\right)^{2}, & \left|\boldsymbol{Y}_{\mathrm{R}} - \boldsymbol{Y}_{\mathrm{P}}\right| \le \delta_{h} \\ \delta_{h}\left|\boldsymbol{Y}_{\mathrm{R}} - \boldsymbol{Y}_{\mathrm{P}}\right| - \frac{1}{2}\delta_{h}^{2}, & \left|\boldsymbol{Y}_{\mathrm{R}} - \boldsymbol{Y}_{\mathrm{P}}\right| > \delta_{h} \end{cases} \quad (27)$$

where $\boldsymbol{Y}_{\mathrm{R}}$ and $\boldsymbol{Y}_{\mathrm{P}}$ are the real and predicted values of traffic flow, respectively, and δ_h is the loss threshold. The specific process is shown in Algorithm 1.

Algorithm 1 MD-GRTN Training Algorithm

Input: Noisy traffic flow features: $[\boldsymbol{X}_{\mathrm{RecN}}, \boldsymbol{X}_{\mathrm{HourN}}, \boldsymbol{X}_{\mathrm{DayN}}]$, Noise-free traffic flow features: $[\hat{\boldsymbol{X}}_{\mathrm{Rec}}, \hat{\boldsymbol{X}}_{\mathrm{Hour}}, \hat{\boldsymbol{X}}_{\mathrm{Day}}]$.
Output: Learned MD-GRTN model.
$f_{\mathrm{MD}}(\boldsymbol{\theta}_{\mathrm{MD}})$ // Initialize MD module
for epoch in max epoch **do**// Pre training phase
  $\hat{\boldsymbol{H}}_k = BackNet_k\left(\boldsymbol{X}_k\right)$
  Loss (ϵ_θ) = MSE $(\hat{\boldsymbol{H}}_k, \hat{\boldsymbol{X}}_k)$ // Minimize Loss
  Update $\{\boldsymbol{\theta}_{\mathrm{MD}}\}$
end for
$f_{\mathrm{MDAF}}(\boldsymbol{\theta}_{\mathrm{MD}}, \boldsymbol{\theta}_{\mathrm{MAF}})$ // Initialize MDAF module
$f_{\mathrm{MGRC}}(\boldsymbol{\theta}_{\mathrm{MGRC}})$ // Initialize MGRC module
$f_{\mathrm{ST}}(\boldsymbol{\theta}_{\mathrm{ST}})$ // Initialize Spatial Transformer module
$f_{\mathrm{TT}}(\boldsymbol{\theta}_{\mathrm{TT}})$ // Initialize Temporal Transformer module
for epoch in max epoch **do**// Main training phase
  $\boldsymbol{H}_{\mathrm{F1}} = f_{\mathrm{MDAF}}(\boldsymbol{X}_{\mathrm{RecN}}, \boldsymbol{X}_{\mathrm{HourN}}, \boldsymbol{X}_{\mathrm{DayN}})$
  $\boldsymbol{H}_{\mathrm{F2}} = f_{\mathrm{MGRC}}(\boldsymbol{A}_{\mathrm{dyna}}, \boldsymbol{A}_{\mathrm{dist}}, \boldsymbol{H}_{\mathrm{F1}})$
  $\boldsymbol{H}_{\mathrm{ST}} = f_{\mathrm{TT}}(\boldsymbol{E}_{\mathrm{hour}}, \boldsymbol{E}_{\mathrm{Hour}}, \boldsymbol{E}_{\mathrm{Day}}, f_{\mathrm{ST}}(\boldsymbol{A}, \boldsymbol{H}_{\mathrm{F2}}))$
  Loss $= Huber(\boldsymbol{Y}_{\mathrm{R}}, \boldsymbol{H}_{\mathrm{ST}})$ // Minimize Loss
  Update $\{\boldsymbol{\theta}_{\mathrm{MAF}}, \boldsymbol{\theta}_{\mathrm{MGRC}}, \boldsymbol{\theta}_{\mathrm{ST}}, \boldsymbol{\theta}_{\mathrm{TT}}\}$
end for

TABLE I
DATASET INFORMATION

Datasets	Country	Date	Nodes
PEMS03	USA	9/1-11/30/2018	358
PEMS04	USA	1/1-2/28/2018	307
PEMS07	USA	5/1-8/31/2017	883
PEMS08	USA	7/1-8/31/2016	170
SZTaxi	CHINA	1/1–1/31/2015	156

V. EXPERIMENTS

A. Datasets

This paper selects five real-world datasets to validate the model's predictive performance. The PEMS03, PEMS04, PEMS07, and PEMS08 datasets are widely used public datasets commonly employed to compare the performance of different models [38]. The PEMS datasets are sourced from the Performance Measurement System of the California Department of Transportation, with a collection interval of 5 minutes. The SZTaxi dataset, collected from taxi GPS data in Shenzhen, China, includes speed information with a collection interval of 15 minutes [52]. Detailed descriptions of the five datasets are provided in TABLE I. To obtain noisy traffic flow data, Gaussian noise is added to the original datasets. The Gaussian noise added to the PEMS datasets has a mean of 0 and a standard deviation of 10, while for the SZTaxi dataset, the noise has a mean of 0 and a standard deviation of 2. These settings are designed to simulate real-world sensor noise, where loop detectors (used in PEMS) typically exhibit higher variance due to environmental interference and hardware limitations, whereas GPS-based measurements in the SZTaxi dataset tend to be more stable, justifying a lower noise level. All datasets are standardized using Z-Score normalization before being input into the model.

B. Experimental Settings

1) Hyperparameter Settings: The model was trained and tested on a Windows 11 platform with an Nvidia 4070 Ti GPU, 64GB of RAM, and an i9-13900K CPU. The MD-GRTN was implemented using PyTorch, and the optimal hyperparameters were determined through grid search. The input and output time steps of the MD-GRTN were set to 12. The optimizer was AdamW, with an initial learning rate of 0.001, a batch size of 64, and a maximum of 800 iterations. The number of attention heads was set to 3, the number of layers in the STFormer module was set to 3, and the number of layers in the MGRC module was set to 6. The weight decay was set to 0.01. The datasets were split into training, validation, and test sets in a 7:1:2 ratio. The input to the MD-GRTN was noisy traffic flow data, while the output labels were noise-free traffic flow data.

2) Baseline Models: This paper selects three typical types of models in the traffic flow prediction field to compare the proposed method's performance:

(a) Type 1: Classical models

ARIMA: A statistical time series model designed for capturing linear temporal dependencies, making it effective for forecasting stationary data with trend and seasonality components.

LSTM: A recurrent neural network model specialized for capturing long-term temporal dependencies, making it well-suited for sequence-based tasks.

CNN: CNNs are effective in recognizing patterns in spatially organized data, but they lack the ability to capture temporal dependencies, which are essential in traffic flow prediction.

(b) Type 2: Improved GCN models

DCRNN [37]: DCRNN combines diffusion convolution with recurrent layers to model spatial and temporal dependencies in traffic data, particularly effective for structured road networks.

STSGCN [38]: STSGCN captures spatial temporal dependencies in traffic data by synchronizing spatial and temporal feature extraction, improving prediction accuracy on non-Euclidean road networks.

STFGNN [39]: STFGNN uses a data-driven approach to create "temporal graphs" that complement spatial correlations, enabling effective learning of hidden spatial-temporal dependencies.

STFGCN [41]: STFGCN: A graph neural network model designed for traffic prediction, integrating multi-scale temporal dependencies and dynamic spatial dependencies to enhance forecasting accuracy.

(c) Type 3: Attention-enhanced models

ASTGNN [43]: ASTGNN is an attention-based spatial-temporal graph neural network for traffic forecasting, featuring a self-attention mechanism for capturing temporal dynamics and a dynamic graph convolution module for spatial correlations.

PDFormer [44]: PDFormer is a propagation delay-aware dynamic transformer for traffic flow prediction, featuring a spatial self-attention module to capture dynamic spatial dependencies and graph masking matrices for short- and long-range spatial views.

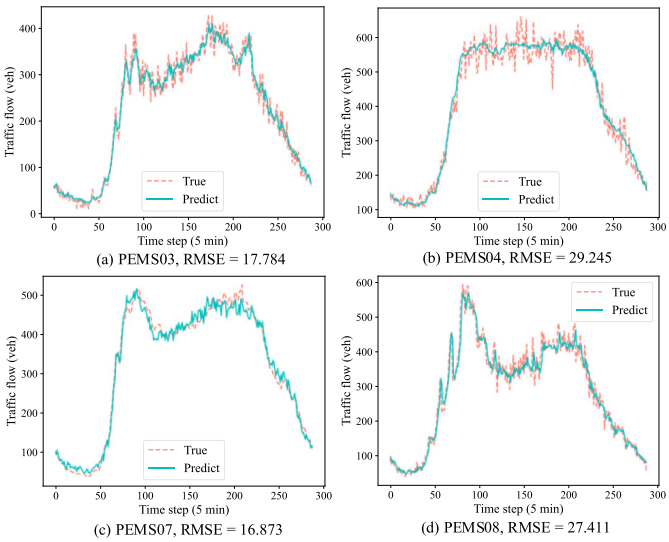


Fig. 6. Local prediction performance of MD-GRTN.

The aforementioned nine baseline models and MD-GRTN were tested and compared in the same environment.

3) Evaluation Indicators: Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE) were used to evaluate the prediction accuracy of the models. The number of parameters (Params) and floating point operations (FLOPs) were used to assess the spatial-temporal complexity of the models. The evaluation equations for the three prediction errors are as follows:

$$RMSE = \sqrt{\frac{1}{p}\sum_{i=1}^{p}(\boldsymbol{Y_R}(i) - \boldsymbol{Y_P}(i))^2} \tag{28}$$

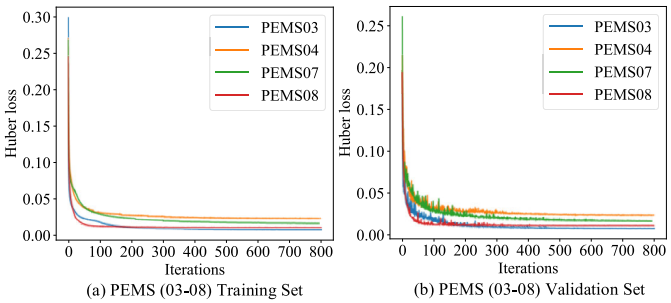
$$MAE = \frac{1}{p}\sum_{i=1}^{p}|\boldsymbol{Y_R}(i) - \boldsymbol{Y_P}(i)| \tag{29}$$

$$MAPE = \frac{100\%}{p}\sum_{i=1}^{p}\left|\frac{\boldsymbol{Y_R}(i) - \boldsymbol{Y_P}(i)}{\boldsymbol{Y}_{\mathrm{R}}(i)}\right| \tag{30}$$

where p is the number of samples to be tested.

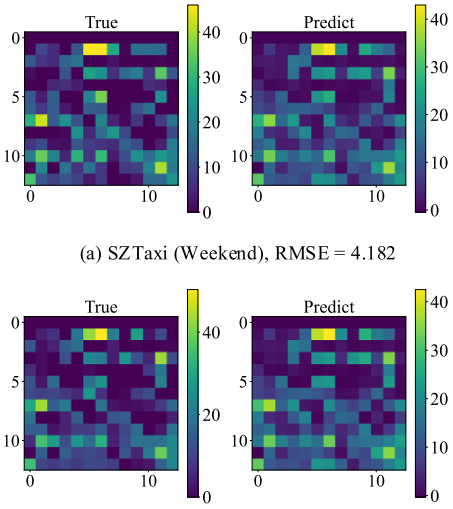
C. Experimental Results

1) Predictive Performance of MD-GRTN: This paper uses the PEMS (03-08) datasets to validate the prediction performance of the proposed model at local nodes, as shown in Fig. 6. All four datasets exhibit significant peak and valley effects in traffic flow, and the proposed model effectively captures local traffic flow trends under different conditions. In all four datasets, traffic flow begins to peak around 6:30 AM. Among them, PEMS03 and PEMS04 show greater fluctuations in traffic flow during peak periods, while PEMS07 and PEMS08 exhibit less volatility in traffic flow. The addition of Gaussian noise is applied only to the input data to simulate real-world fluctuations, ensuring the model's robustness in



(a) PEMS (03-08) Training Set (b) PEMS (03-08) Validation Set

Fig. 7. Convergence evaluation.



(a) SZTaxi (Weekend), RMSE = 4.182

(b) SZTaxi (Weekday), RMSE = 4.939

Fig. 8. Global prediction performance of MD-GRTN.

handling perturbed traffic conditions while maintaining accurate predictions of actual traffic flow. This paper also tests the convergence of the proposed method on the PEMS (03-08) datasets, as shown in Fig. 7, where it can be found that the proposed method reaches convergence after 600 rounds.

The SZTaxi dataset is used to validate the performance of MD-GRTN in predicting traffic flow on holidays across different dates, as shown in Fig. 8. Fig. 8(a) and Fig. 8(b) respectively show the global fitting status on a randomly selected weekday and a randomly selected weekend day. It can be observed that the speed peak points are similar on both the weekday and the weekend, indicating that traffic on this road segment is not easily affected by holidays. The results demonstrate that MD-GRTN effectively captures the global traffic flow trends across different dates.

To validate the noise robustness of MD-GRTN, Gaussian noise with a mean of 0 and standard deviations of 10, 12, 14, and 16 is added to the PEMS (03-08) datasets as input, as shown in Fig. 9. As the noise level increases, the RMSE prediction error of MD-GRTN consistently remains within 6.296%, demonstrating strong anti-interference and noise-reduction capabilities.

2) Comparison Results: The PEMS (03-08) datasets are used to demonstrate the advantages of MD-GRTN compared to

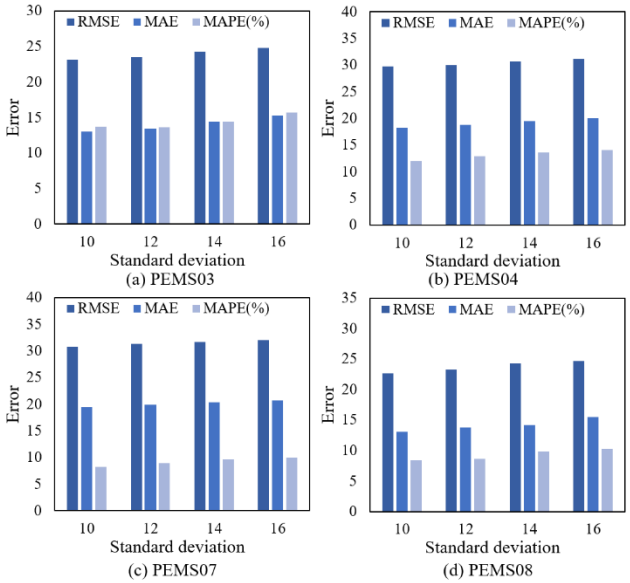


Fig. 9. Disturbance resistance of MD-GRTN.

TABLE II
COMPARISON WITH BASELINE MODELS IN PEMS03

Baselines	Dataset	PEMS03		
	Metrics	RMSE	MAE	MAPE(%)
ARIMA (1970)		41.587	25.896	27.314
LSTM (1997)		35.233	21.132	21.561
CNN (1998)		32.987	20.725	20.773
DCRNN (2018)		30.427	18.239	18.924
STSGCN (2020)		28.873	17.379	16.882
STFGNN (2021)		28.342	16.773	16.309
STFGCN (2024)		25.136	14.549	14.034
ASTGNN (2022)		25.178	14.773	14.572
PDFormer (2023)		25.047	14.176	13.851
MD-GRTN		**23.145**	**13.019**	**13.670**

baseline models, as shown in TABLEs II to V. The comparison results indicate that among the baseline models, the best performance is achieved by attention-enhanced models, while the worst performance is seen in classical models. Within the classical models, CNN performs the best; among the GCN-enhanced models, STFGCN performs the best; and among the attention-enhanced models, PDFormer performs the best. The proposed MD-GRTN achieves an average RMSE reduction of 25.456%, 4.672%, and 4.665% compared to CNN, STFGCN, and PDFormer.

The SZTaxi dataset on weekdays and the PEMS04 dataset on weekends are used to validate the predictive performance of the CNN, STFGCN, PDFormer, and MD-GRTN models, as shown in Fig. 10. The proposed MD-GRTN model accurately captures traffic flow on both weekdays and weekends, demonstrating superior predictive performance compared to the baseline models.

The prediction accuracy for missing data is a key indicator of a model's noise robustness. The PEMS (03-08) datasets are used to validate the predictive accuracy of the CNN, STFGCN,

TABLE III
COMPARISON WITH BASELINE MODELS IN PEMS04

Baselines	Dataset	PEMS04		
	Metrics	RMSE	MAE	MAPE(%)
ARIMA (1970)		57.139	37.311	26.050
LSTM (1997)		41.732	27.162	18.598
CNN (1998)		39.076	27.078	17.623
DCRNN (2018)		38.383	24.926	17.482
STSGCN (2020)		34.136	21.014	13.905
STFGNN (2021)		31.891	19.836	13.021
STFGCN (2024)		30.163	18.387	12.232
ASTGNN (2022)		30.791	18.606	12.476
PDFormer (2023)		29.965	18.321	12.103
MD-GRTN		**29.737**	**18.174**	**12.048**

TABLE IV
COMPARISON WITH BASELINE MODELS IN PEMS07

Baselines	Dataset	PEMS07		
	Metrics	RMSE	MAE	MAPE(%)
ARIMA (1970)		55.348	37.849	17.366
LSTM (1997)		45.927	30.021	13.532
CNN (1998)		40.794	32.141	12.361
DCRNN (2018)		38.856	25.764	11.693
STSGCN (2020)		38.894	24.768	10.663
STFGNN (2021)		35.878	22.087	9.253
STFGCN (2024)		33.102	19.648	8.154
ASTGNN (2022)		33.782	20.503	8.766
PDFormer (2023)		32.870	19.832	8.529
MD-GRTN		**30.716**	**19.374**	**8.248**

TABLE V
COMPARISON WITH BASELINE MODELS IN PEMS08

Baselines	Dataset	PEMS08		
	Metrics	RMSE	MAE	MAPE(%)
ARIMA (1970)		41.671	28.123	18.259
LSTM (1997)		34.074	22.373	14.682
CNN (1998)		29.528	21.858	13.790
DCRNN (2018)		27.837	17.861	11.458
STSGCN (2020)		27.555	17.883	11.709
STFGNN (2021)		26.231	16.642	10.608
STFGCN (2024)		23.119	13.618	8.973
ASTGNN (2022)		24.173	15.214	9.482
PDFormer (2023)		23.505	13.583	9.046
MD-GRTN		**22.623**	**13.114**	**8.471**

PDFormer, and MD-GRTN models under conditions where 5%, 10%, 15%, and 20% of the data is randomly missing, as shown in Fig. 11. As the percentage of missing data increases, the error values for all models also increase. However, MD-GRTN consistently maintains the highest prediction accuracy. The SZTaxi dataset is used to validate the predictive performance of the STFGCN, PDFormer, and MD-GRTN models during peak traffic periods, as shown in Fig. 12. Among the three models, MD-GRTN achieves the highest prediction accuracy, effectively capturing the global speed trends during peak periods.

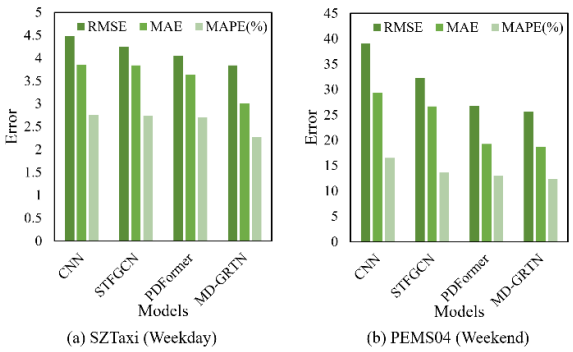


Fig. 10. Comparison of prediction performance on weekday and weekend.

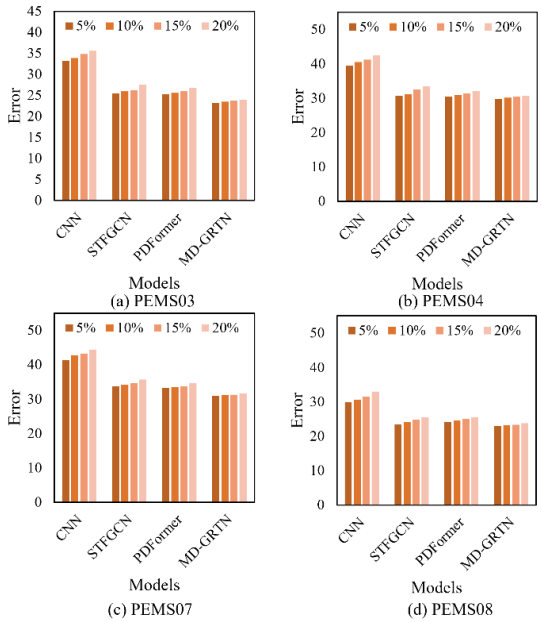


Fig. 11. Performance comparison of noise robustness.

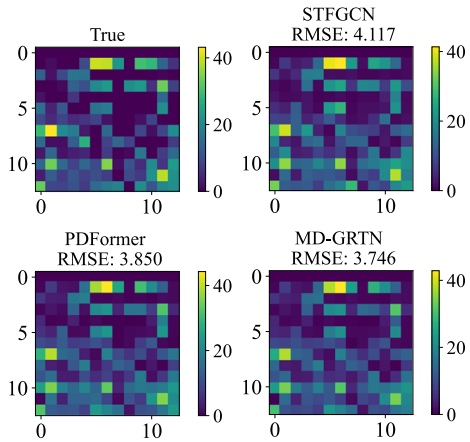


Fig. 12. Comparison of global prediction performance at peak periods.

The long-term prediction performance of the models is crucial. This paper chooses the PEMS03 dataset to validate the long-term prediction performance of nine models such as LSTM, CNN, etc., and the prediction intervals are categorized

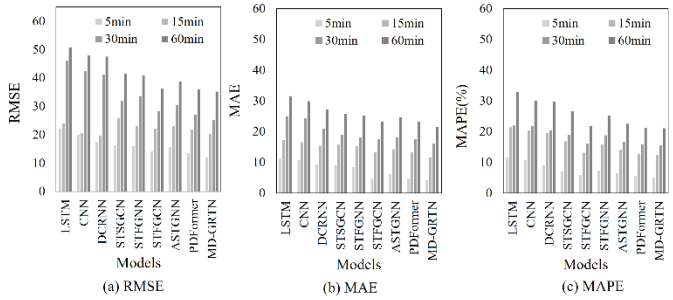


Fig. 13. Comparison of long-term predictive performance.

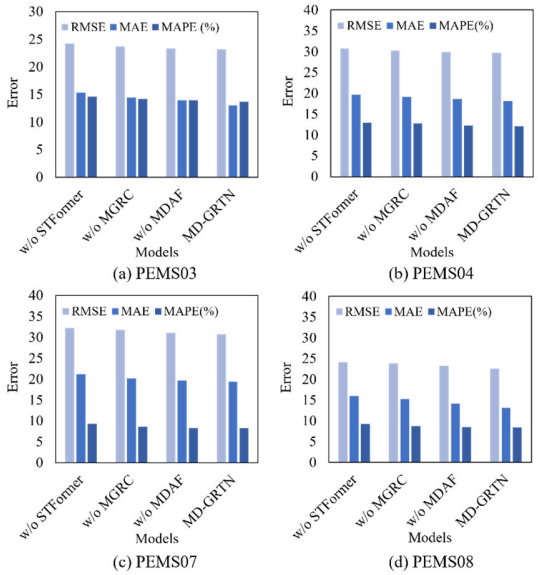


Fig. 14. Component ablation experiments.

into 5, 15, 30 and 60 minutes, as shown in Fig. 13. As the prediction interval increases, the prediction errors of all models increase, and MD-GRTN always maintains the optimal prediction effect, which reflects that the proposed method has better long-term prediction performance.

3) Ablation Experiments: To validate the superiority of the model structure, three variants are designed: Variant 1 (w/o MDAF), which removes the MDAF module; Variant 2 (w/o MGRC), which removes the MGRC module; and Variant 3 (w/o STFormer), which removes the STFormer module. The PEMS (03-08) datasets are used to test the predictive performance of MD-GRTN and its three variants, as shown in Fig. 14. Compared to MD-GRTN, the RMSE error increases by 4.887%, 3.279%, and 1.405% for Variant 1, Variant 2, and Variant 3, respectively. The STFormer module is the most critical to the performance of MD-GRTN, while the MDAF module is the least important.

This paper also tested the spatial-temporal complexity of MD-GRTN and its three variants. FLOPs were used to measure time complexity, and Params were used to measure space complexity, as shown in Fig. 15. Within the MD-GRTN structure, the STFormer module has the highest spatial-temporal complexity, while the MDAF module has the lowest. Although the spatial-temporal complexity of MD-GRTN is higher than

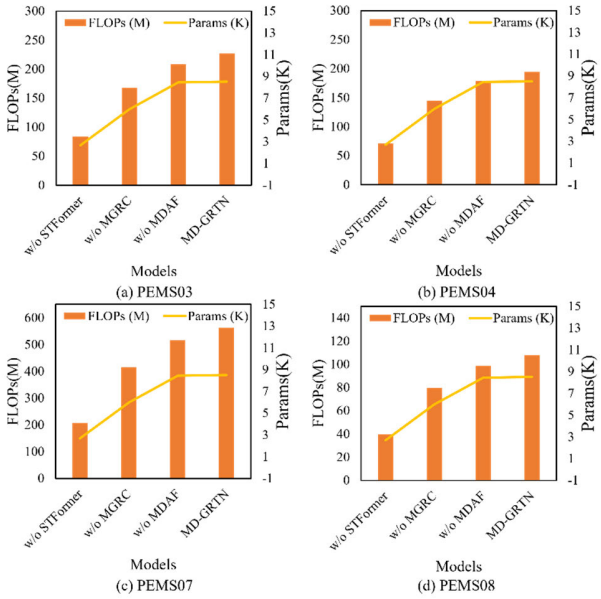


Fig. 15. Comparative spatial and temporal complexity.

that of the three variants, it achieves the best prediction accuracy.

4) Discussion: This paper tested the performance of the proposed MD-GRTN model using the PEMS (03-08) datasets. Across all four datasets, MD-GRTN effectively responded to traffic flow peaks and valleys, accurately capturing the traffic flow trends at local nodes. As shown in Fig. 6, all datasets reached peak traffic flow around 6:30 AM, with PEMS03 and PEMS04 exhibiting greater volatility, while PEMS07 and PEMS08 showed less fluctuation. Validation using the SZTaxi dataset, as depicted in Fig. 7, demonstrated MD-GRTN's global fitting performance on both weekdays and weekends. The similarity in peak traffic points on weekdays and weekends indicated that MD-GRTN could effectively adapt to traffic flow trends across different dates. To test the model's noise robustness, Gaussian noise with varying standard deviations were added to the PEMS (03-08) datasets, as shown in Fig. 9. Despite the increased noise, MD-GRTN's RMSE prediction error consistently remained within 6.296%, demonstrating strong anti-interference and noise-correction capabilities.

In comparison experiments with baseline models, the PEMS (03-08) datasets were used to demonstrate the advantages of MD-GRTN, with specific results shown in TABLEs II to V. The comparison results indicated that attention-enhanced models performed the best among the baseline models, while classical models performed the worst. Within the classical models, CNN performed best, and STFGCN showed the best performance among the GCN-enhanced models. Among the attention-enhanced models, PDFormer performed the best. In comparison, MD-GRTN achieved an average RMSE reduction of 25.456%, 4.672%, and 4.665% compared to CNN, STFGCN, and PDFormer, respectively, demonstrating superior predictive performance among the baseline models. The SZTaxi dataset was used to validate MD-GRTN's predictive

performance on weekdays and holidays, with results shown in Fig. 10. Whether on weekdays or weekends, MD-GRTN outperformed other baseline models in prediction accuracy, indicating its strong generalization capability under different conditions. The prediction accuracy test for missing data showed that MD-GRTN consistently maintained the highest accuracy with 5%, 10%, 15%, and 20% of data randomly missing, as seen in Fig. 11. The validation results of peak-period prediction performance using the SZTaxi dataset are shown in Fig. 12, where MD-GRTN outperformed the other three models, effectively reflecting global speed trends during peak periods. For long-term prediction performance, Fig. 13 shows that MD-GRTN consistently achieved the best prediction results with prediction intervals of 5, 15, 30, and 60 minutes, demonstrating exceptional long-term predictive capability. The poorer performance of classical models can be attributed to their focus on extracting only a single feature, neglecting the contributions of other features. While GCN-enhanced models performed well in mining local spatial-temporal features, they lacked the ability to capture global spatial-temporal correlations. Attention-enhanced models dynamically learned spatial-temporal dependencies between nodes through attention mechanisms, enabling them to adaptively weigh relevant temporal and spatial information. This dynamic learning capability contributed to their superior performance over GCN-based models, which are constrained by fixed graph structures. However, attention-based models still failed to effectively explore the intrinsic generative mechanisms of raw traffic data and long-term temporal dependencies. In contrast, the MD-GRTN model integrated multi-dimensional feature extraction mechanisms, allowing it to more comprehensively capture complex traffic flow patterns, thereby enhancing prediction performance.

To validate the superiority of the MD-GRTN model structure, three variants were created by removing the MDAF module (Variant 1), the MGRC module (Variant 2), and the STFormer module (Variant 3). The PEMS (03-08) datasets were used to test the predictive performance of MD-GRTN and its three variants, as shown in Fig. 14. The RMSE errors for Variant 1, Variant 2, and Variant 3 increased by 4.887%, 3.279%, and 1.405%, respectively, indicating that the STFormer module is the most critical to MD-GRTN, while the MDAF module is the least important. The analysis of MD-GRTN's spatial-temporal complexity is shown in Fig. 15, where FLOPs and Params were used to measure time and space complexity. In the MD-GRTN structure, the STFormer module had the highest spatial-temporal complexity, while the MDAF module had the lowest. Although the spatial-temporal complexity of MD-GRTN is higher than that of the three variants, it achieves the best prediction accuracy, confirming the superiority of MD-GRTN in spatial-temporal feature extraction and processing.

Although MD-GRTN improves traffic flow prediction, it still has limitations. (1) Its spatial-temporal complexity is relatively high, which may impact real-time deployment in resource-constrained environments. (2) While it effectively leverages noisy data, further adaptation to dynamic noise distributions and extreme anomalies is needed. (3) The model focuses on traffic flow prediction, and its generalizability to other transportation tasks requires further validation. Future work will aim to reduce model complexity for efficient edge deployment, enhance robustness to diverse noise patterns, and extend its application to broader intelligent transportation scenarios.

VI. Conclusion

In this paper, we propose a novel MD-GRTN for traffic flow prediction within vehicular networks. MD-GRTN consists of three main components: the MDAF module, the MGRC module, and the STFormer module. The MDAF module combines the MD module and the MAF module to enhance historical trend features of multi-period traffic flows. The MGRC module integrates a multi-graph fusion module, a graph convolutional network, and a gated recurrent unit to strengthen spatial features influenced by various factors in vehicular environments. Lastly, the STFormer module, comprising spatial and temporal Transformer modules, further enhances the global dynamic spatial-temporal features of traffic flow within complex vehicular networks. Extensive experiments on five real-world datasets have demonstrated that MD-GRTN outperforms state-of-the-art models in vehicular network scenarios. In future work, we aim to further reduce the spatial-temporal complexity of the proposed model to improve its suitability for real-time deployment in intelligent vehicular networks. Specifically, we will investigate model compression techniques such as pruning, quantization, and knowledge distillation to achieve efficient inference without compromising accuracy.

Conflict of Interest

The authors declare that they have no conflicts of interest in this work.

Contribution Statement

Yinxin Bao: Conceptualization, Methodology, Writing-Original draft preparation, Writing-review & editing. **Qinqin Shen:** Conceptualization, Validation. **Yang Cao:** Conceptualization, Validation. **Yingyan Hou:** Funding acquisition, Validation. **Wanxuan Lu:** Funding acquisition, Validation. **Quan Shi:** Investigation, Project administration, Funding acquisition.

Data Availability

The data are available at *https://github.com/Bounger2*

References

[1] T. R. Gadekallu et al., "XAI for industry 5.0—Concepts, opportunities, challenges, and future directions," *IEEE Open J. Commun. Soc.*, vol. 6, pp. 2706–2729, 2025, doi: 10.1109/OJCOMS.2024.3473891.

[2] S. Bhattacharya, S. R. K. Somayaji, T. R. Gadekallu, M. Alazab, and P. K. R. Maddikunta, "A review on deep learning for future smart cities," *Internet Technol. Lett.*, vol. 5, no. 1, p. e187, Jan. 2022, doi: 10.1002/itl2.187.

[3] S. K. Singh, J. H. Park, P. K. Sharma, and Y. Pan, "BIIoVT: Blockchain-based secure storage architecture for intelligent Internet of Vehicular Things," *IEEE Consum. Electron. Mag.*, vol. 11, no. 6, pp. 75–82, Nov. 2022, doi: 10.1109/MCE.2021.3089992.

[4] L. Liu, M. Liu, G. Li, Z. Wu, J. Lin, and L. Lin, "Road network-guided fine-grained urban traffic flow inference," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 36, no. 1, pp. 1119–1132, Jan. 2025, doi: 10.1109/TNNLS.2023.3327386.

[5] N. S. Chauhan, N. Kumar, and A. Eskandarian, "A novel confined attention mechanism driven bi-GRU model for traffic flow prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 25, no. 8, pp. 9181–9191, Aug. 2024, doi: 10.1109/TITS.2024.3375890.

[6] L. Ren, Z. Jia, Y. Laili, and D. Huang, "Deep learning for time-series prediction in IIoT: Progress, challenges, and prospects," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 11, pp. 15072–15091, 2023, doi: 10.1109/TNNLS.2023.3291371.

[7] W. Li et al., "Location and time embedded feature representation for spatiotemporal traffic prediction," *Expert Syst. Appl.*, vol. 239, Apr. 2024, Art. no. 122449, doi: 10.1016/j.eswa.2023.122449.

[8] G. Comert, N. Begashaw, and N. Huynh, "Improved grey system models for predicting traffic parameters," *Expert Syst. Appl.*, vol. 177, Sep. 2021, Art. no. 114972, doi: 10.1016/j.eswa.2021.114972.

[9] Y. Miao et al., "A novel short-term traffic prediction model based on SVD and ARIMA with blockchain in Industrial Internet of Things," *IEEE Internet Things J.*, vol. 10, no. 24, pp. 21217–21226, Dec. 2023, doi: 10.1109/JIOT.2023.3283611.

[10] C. Ma, Y. Zhao, G. Dai, X. Xu, and S.-C. Wong, "A novel STFSA-CNN-GRU hybrid model for short-term traffic speed prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 4, pp. 3728–3737, Apr. 2023, doi: 10.1109/TITS.2021.3117835.

[11] Q. Tan, Y. Liu, and J. Liu, "Demystifying deep learning in predictive spatiotemporal analytics: An information-theoretic framework," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 8, pp. 3538–3552, Aug. 2021, doi: 10.1109/TNNLS.2020.3015215.

[12] J. Ji et al., "Spatio-temporal self-supervised learning for traffic flow prediction," in *Proc. AAAI Conf. Artif. Intell.*, vol. 37, 2023, pp. 4356–4364, doi: 10.1609/aaai.v37i4.25555.

[13] X. Huang, J. Wang, Y. Lan, C. Jiang, and X. Yuan, "MD-GCN: A multi-scale temporal dual graph convolution network for traffic flow prediction," *Sensors*, vol. 23, no. 2, p. 841, Jan. 2023, doi: 10.3390/s23020841.

[14] F. Wei, X. Li, Y. Guo, Z. Wang, Q. Li, and X. Ma, "Flow direction level traffic flow prediction based on a GCN-LSTM combined model," *Intell. Autom. Soft Comput.*, vol. 37, no. 2, pp. 2001–2018, 2023, doi: 10.32604/iasc.2023.035799.

[15] C. Wang, L. Wang, S. Wei, Y. Sun, B. Liu, and L. Yan, "STN-GCN: Spatial and temporal normalization graph convolutional neural networks for traffic flow forecasting," *Electronics*, vol. 12, no. 14, p. 3158, Jul. 2023, doi: 10.3390/electronics12143158.

[16] J. Zhao, X. Xiong, Q. Zhang, and D. Wang, "Extended multi-component gated recurrent graph convolutional network for traffic flow prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 25, no. 5, pp. 4634–4644, May 2024, doi: 10.1109/TITS.2023.3322745.

[17] J. Yan, L. Zhang, Y. Gao, and B. Qu, "GECRAN: Graph embedding based convolutional recurrent attention network for traffic flow prediction," *Expert Syst. Appl.*, vol. 256, Dec. 2024, Art. no. 125001.

[18] G. Zhang, G. Yuan, D. Cheng, L. Liu, J. Li, and S. Zhang, "Disentangled contrastive learning for fair graph representations," *Neural Netw.*, vol. 181, Jan. 2025, Art. no. 106781.

[19] C. Wang, R. Tian, J. Hu, and Z. Ma, "A trend graph attention network for traffic prediction," *Inf. Sci.*, vol. 623, pp. 275–292, Apr. 2023, doi: 10.1016/j.ins.2022.12.048.

[20] D. Xia, B. Shen, J. Geng, Y. Hu, Y. Li, and H. Li, "Attention-based spatial–temporal adaptive dual-graph convolutional network for traffic flow forecasting," *Neural Comput. Appl.*, vol. 35, no. 23, pp. 17217–17231, Aug. 2023, doi: 10.1007/s00521-023-08582-1.

[21] W. Zhang, R. Yao, X. Du, Y. Liu, R. Wang, and L. Wang, "Traffic flow prediction under multiple adverse weather based on self-attention mechanism and deep learning models," *Phys. A, Stat. Mech. Appl.*, vol. 625, Sep. 2023, Art. no. 128988, doi: 10.1016/j.physa.2023.128988.

[22] J. Zuo, K. Zeitouni, Y. Taher, and S. Garcia-Rodriguez, "Graph convolutional networks for traffic forecasting with missing values," *Data Mining Knowl. Discovery*, vol. 37, no. 2, pp. 913–947, Mar. 2023, doi: 10.1007/s10618-022-00903-7.

[23] X. Kong, W. Zhou, G. Shen, W. Zhang, N. Liu, and Y. Yang, "Dynamic graph convolutional recurrent imputation network for spatiotemporal traffic missing data," *Knowl.-Based Syst.*, vol. 261, Feb. 2023, Art. no. 110188, doi: 10.1016/j.knosys.2022.110188.

[24] G. Zhang, S. Zhang, and G. Yuan, "Bayesian graph local extrema convolution with long-tail strategy for misinformation detection," *ACM Trans. Knowl. Discovery Data*, vol. 18, no. 4, pp. 1–21, May 2024.

[25] Bharti, P. Redhu, and K. Kumar, "Short-term traffic flow prediction based on optimized deep learning neural network: PSO-Bi-LSTM," *Phys. A, Stat. Mech. Appl.*, vol. 625, Sep. 2023, Art. no. 129001, doi: 10.1016/j.physa.2023.129001.

[26] Y. Li, M. Liang, H. Li, Z. Yang, L. Du, and Z. Chen, "Deep learning-powered vessel traffic flow prediction with spatial–temporal attributes and similarity grouping," *Eng. Appl. Artif. Intell.*, vol. 126, Nov. 2023, Art. no. 107012, doi: 10.1016/j.engappai.2023.107012.

[27] S. K. Singh, S. Chauhan, A. Alsafrani, M. Islam, H. I. Sherazi, and I. Ullah, "Optimizing healthcare data quality with optimal features driven mutual entropy gain," *Expert Syst.*, vol. 42, no. 2, p. 13737, Feb. 2025.

[28] R. Pandey, M. Koranga, S. N. Thakur, H. Khan, S. K. Singh, and R. N. Ravikumar, "Securing vehicle-to-grid communications: A cyber-physical approach," in *Optimized Energy Management Strategies for Electric Vehicles*. Hershey, PA, USA: IGI Global, 2025, pp. 301–318.

[29] R. Zhang, S. Mao, and Y. Kang, "A novel traffic flow prediction model: Variable order fractional grey model based on an improved grey evolution algorithm," *Expert Syst. Appl.*, vol. 224, Aug. 2023, Art. no. 119943, doi: 10.1016/j.eswa.2023.119943.

[30] Z. Cheng, J. Lu, H. Zhou, Y. Zhang, and L. Zhang, "Short-term traffic flow prediction: An integrated method of econometrics and hybrid deep learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 6, pp. 5231–5244, Jun. 2022, doi: 10.1109/TITS.2021.3052796.

[31] M. Méndez, M. G. Merayo, and M. Núñez, "Long-term traffic flow forecasting using a hybrid CNN-BiLSTM model," *Eng. Appl. Artif. Intell.*, vol. 121, May 2023, Art. no. 106041, doi: 10.1016/j.engappai.2023.106041.

[32] Z. Yang and C. Wang, "Short-term traffic flow prediction based on AST-MTL-CNN-GRU," *IET Intell. Transp. Syst.*, vol. 17, no. 11, pp. 2205–2220, Nov. 2023, doi: 10.1049/ITR2.12400.

[33] R. He, Y. Xiao, X. Lu, S. Zhang, and Y. Liu, "ST-3DGMR: Spatio-temporal 3D grouped multiscale ResNet network for region-based urban traffic flow prediction," *Inf. Sci.*, vol. 624, pp. 68–93, May 2023, doi: 10.1016/j.ins.2022.12.066.

[34] Y. Y. Pu, W. H. Wang, Q. Zhu, and P. P. Chen, "Urban short-term traffic flow prediction algorithm based on CNN-ResNet-LSTM model," *Beijing Youdian Daxue Xuebao/J. Beijing Univ. Posts Telecommun.*, vol. 43, no. 5, p. 9, 2020, doi: 10.13190/j.jbupt.2019-243.

[35] X. Ren, H. Mosavat-Jahromi, L. Cai, and D. Kidston, "Spatio-temporal spectrum load prediction using convolutional neural network and ResNet," *IEEE Trans. Cognit. Commun. Netw.*, vol. 8, no. 2, pp. 502–513, Jun. 2022, doi: 10.1109/TCCN.2021.3139030.

[36] H. Wang, J. Chen, Z. Fan, Z. Zhang, Z. Cai, and X. Song, "ST-ExpertNet: A deep expert framework for traffic prediction," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 7, pp. 7512–7525, Jul. 2022, doi: 10.1109/TKDE.2022.3196936.

[37] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2018, pp. 1–16.

[38] C. Song, Y. Lin, S. Guo, and H. Wan, "Spatial–temporal synchronous graph convolutional networks: A new framework for spatial–temporal network data forecasting," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 914–921, doi: 10.1609/aaai.v34i01.5438.

[39] M. Li and Z. Zhu, "Spatial–temporal fusion graph neural networks for traffic flow forecasting," in *Proc. 35th AAAI Conf. Artif. Intell.*, 2021, pp. 4189–4196, doi: 10.1609/aaai.v35i5.16542.

[40] G. Jin et al., "Automated dilated spatio-temporal synchronous graph modeling for traffic prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 8, pp. 8820–8830, 2022.

[41] H. Li, J. Liu, S. Han, J. Zhou, T. Zhang, and C. L. Philip Chen, "STFGCN: Spatial–temporal fusion graph convolutional network for traffic prediction," *Expert Syst. Appl.*, vol. 255, Dec. 2024, Art. no. 124648.

[42] 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, doi: 10.1609/aaai.v33i01.3301922.

[43] S. Guo, Y. Lin, H. Wan, X. Li, and G. Cong, "Learning dynamics and heterogeneity of spatial–temporal graph data for traffic forecasting," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 11, pp. 5415–5428, Nov. 2022, doi: 10.1109/TKDE.2021.3056502.

[44] J. Jiang, C. Han, W. X. Zhao, and J. Wang, "PDFormer: Propagation delay-aware dynamic long-range transformer for traffic flow prediction," in *Proc. AAAI*, vol. 37, Washington, DC, USA, Jun. 2023, pp. 4365–4373, doi: 10.1609/aaai.v37i4.25556.

[45] Y. Zhu, Y. Ye, X. Zhao, and J. J. Q. Yu, "DiffTraj: Generating GPS trajectory with diffusion probabilistic model," in *Proc. Adv. Neural Inf. Process. Syst.*, Jan. 2023, pp. 65168–65188.

[46] H. Wen et al., "DiffSTG: Probabilistic spatio-temporal graph forecasting with denoising diffusion models," in *Proc. 31st ACM Int. Conf. Adv. Geographic Inf. Syst.*, Nov. 2023, pp. 1–12, doi: 10.1145/3589132.3625614.

[47] Z. Shao et al., "Decoupled dynamic spatial–temporal graph neural network for traffic forecasting," *Proc. VLDB Endowment*, vol. 15, no. 11, pp. 2733–2746, Jul. 2022, doi: 10.14778/3551793.3551827.

[48] A. Wang, Y. Ye, X. Song, S. Zhang, and J. J. Q. Yu, "Traffic prediction with missing data: A multi-task learning approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 4, pp. 4189–4202, Apr. 2023, doi: 10.1109/TITS.2022.3233890.

[49] I. Laña, I. I. Olabarrieta, M. Vélez, and J. D. Ser, "On the imputation of missing data for road traffic forecasting: New insights and novel techniques," *Transp. Res. C, Emerg. Technol.*, vol. 90, pp. 18–33, May 2018, doi: 10.1016/j.trc.2018.02.021.

[50] N. Siddique, S. Paheding, C. P. Elkin, and V. Devabhaktuni, "U-Net and its variants for medical image segmentation: A review of theory and applications," *IEEE Access*, vol. 9, pp. 82031–82057, 2021, doi: 10.1109/ACCESS.2021.3086020.

[51] G. P. Meyer, "An alternative probabilistic interpretation of the Huber loss," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 5261–5269, doi: 10.1109/CVPR46437.2021.00522.

[52] 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, Sep. 2020, doi: 10.1109/TITS.2019.2935152.