

# **A Traffic Flow Prediction Method Based on Temporal Embedding and Lightweight Attention**

Yi Gong

*College of Computer Science and Technology, Shandong University of Technology, China*

**ABSTRACT :** Traffic flow prediction is a fundamental task in intelligent transportation systems. To address limitations of existing methods in exploiting temporal periodic priors, modeling critical temporal dependencies, and stably training attention branches, this paper proposes a Chrono Graph Network (ChronoGraphNet). First, a unified temporal feature embedding is constructed based on the pronounced intraday periodicity of traffic flow, mapping temporal indices to learnable representations and injecting them into the backbone feature space. Second, a lightweight temporal attention mechanism employs a shared  $1 \times 1$  convolution to jointly generate query, key, and value representations, reducing the structural redundancy and additional parameter overhead of independent projections while enhancing critical temporal dependency modeling. Furthermore, a progressive attention activation and dynamic fusion strategy gradually increases the contribution of the attention branch through a freeze-unlock mechanism, epoch-level warm-up, and batch-level fine-tuning. Experiments on PeMS04, PeMS08, and METR-LA show that ChronoGraphNet achieves competitive performance in both overall and multi-horizon prediction, while ablation results confirm the effectiveness and complementarity of the three core components.

**KEYWORDS -** graph neural network, intelligent transportation, machine learning, temporal feature embedding, traffic flow prediction

## **I. INTRODUCTION**

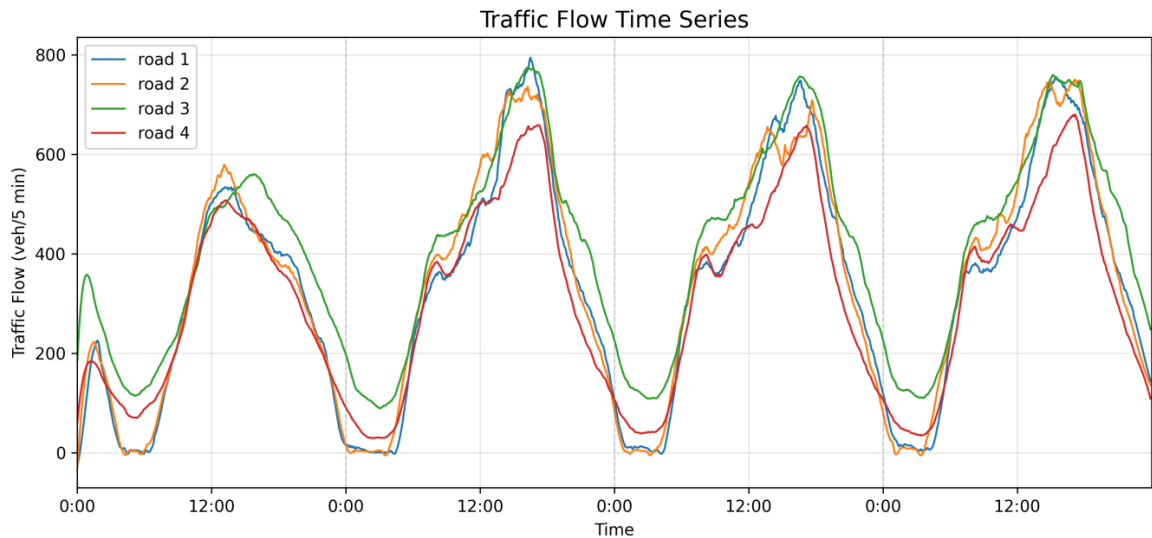
With the continuous expansion of urban road networks and the sustained growth of motor-vehicle travel demand, road transportation systems are facing increasing operating pressure, with traffic congestion, reduced travel efficiency, and rising safety risks becoming increasingly prominent. Against this background, Intelligent Transportation Systems (ITS)<sup>[1]</sup> have become an important technical means of improving traffic efficiency and management. As a fundamental research task in ITS, traffic flow prediction infers future road-network states from historical traffic observations and provides decision support for traffic control, travel guidance, and resource allocation, thereby attracting extensive attention from both academia and engineering practice<sup>[2]</sup>.

In recent years, advances in sensing technologies, the Internet of Things, and big-data processing have gradually shifted traffic data acquisition from offline statistics toward real-time monitoring, providing a strong data foundation for traffic forecasting research. However, traffic data are nonlinear, highly dynamic, spatially and temporally correlated, and distinctly periodic. Fig. 1 illustrates the traffic-flow variations of four representative monitoring nodes in the PeMS04 dataset over four consecutive days. All four nodes exhibit clear intraday periodic fluctuations, indicating the importance of incorporating temporal periodic information into traffic flow prediction.

These practical demands and technical challenges have promoted the application of deep learning to traffic flow prediction. Spatio-Temporal

Graph Neural Networks (STGNNs)<sup>[5]</sup> model spatial dependencies through graph convolution and extract temporal features using recurrent structures or attention mechanisms, substantially improving forecasting performance. Nevertheless, existing methods still leave room for improvement. On the one hand, many methods rely mainly on implicit temporal encoding or multi-branch temporal feature concatenation to model temporal priors, resulting in

insufficient use of traffic periodicity or relatively redundant structures. On the other hand, although attention structures can enhance long-range temporal dependency modeling, they often introduce additional parameters and computational overhead. Moreover, jointly optimizing the attention branch and backbone network from the beginning of training may lead to training oscillations and unstable convergence.



**Figure 1. Daily periodic variation of traffic flow at representative monitoring nodes in PeMS04**

To address these issues, this paper proposes the Chrono Graph Network (ChronoGraphNet) and improves traffic forecasting from three aspects: temporal-prior injection, temporal-dependency enhancement, and training stability. The main contributions are as follows:

(1) A unified temporal feature embedding is proposed. Based on the periodic pattern of 288 time slots per day, an intraday periodic index is constructed and mapped through a learnable embedding into the residual-channel space, thereby explicitly introducing temporal prior information.

(2) A lightweight temporal attention mechanism is designed. A shared  $1 \times 1$  convolution jointly generates query, key, and value representations, and temporal correlations are computed along the time dimension. This design enhances temporal-dependency modeling while reducing structural redundancy and additional parameter overhead introduced by independent projections.

(3) A progressive attention activation and dynamic fusion strategy is proposed. Through staged freeze-unlock control, epoch-level warm-up, and batch-level fine-tuning, the attention branch is changed from direct parallel participation in optimization to staged and progressive involvement in training.

## II. RELATED WORK

Traffic flow prediction has evolved from early statistical modeling to complex spatio-temporal feature learning. Traditional statistical and machine-learning methods are structurally simple and relatively interpretable, but have limited capability in modeling complex nonlinear relationships and dynamic spatio-temporal dependencies. Deep-learning methods can automatically extract complex features through multilayer nonlinear mappings and have achieved substantial improvements in prediction accuracy, but further optimization is still needed in model complexity, long-range dependency modeling, and training stability. In line

with the focus of this study, the following subsections briefly review spatio-temporal feature fusion and attention-based methods.

### 2.1 Spatio-Temporal Feature Fusion Methods

Traffic flow data contain both complex temporal dependencies and spatial correlations, and modeling either dimension alone is insufficient to capture their coupling. Spatio-temporal feature fusion has therefore become a mainstream direction in traffic forecasting. Li et al.<sup>[6]</sup> proposed the Diffusion Convolutional Recurrent Neural Network (DCRNN), which abstracts traffic evolution as a diffusion process on a directed graph, models spatial dependencies using bidirectional random walks, and captures temporal dependencies through an encoder-decoder architecture. Yu et al.<sup>[5]</sup> proposed STGCN, which characterizes spatio-temporal coupling through separated and alternating spatio-temporal convolutional modules. Wu et al.<sup>[7]</sup> proposed Graph WaveNet, which learns latent spatial relationships through an adaptive adjacency matrix and enlarges the temporal receptive field with dilated causal convolutions, thereby improving long-horizon forecasting.

Building on these studies, the Spatial-Temporal Synchronous Graph Convolutional Network (STSGCN)<sup>[8]</sup> captures local spatio-temporal correlations through synchronous modeling; the Adaptive Graph Convolutional Recurrent Network (AGCRN)<sup>[9]</sup> models fine-grained spatial dependencies through node-adaptive parameter learning and data-adaptive graph generation; and the Dynamic Spatial-Temporal Aware Graph Neural Network (DSTAGNN)<sup>[10]</sup> improves spatial modeling flexibility using a data-driven dynamic spatio-temporal aware graph. These studies demonstrate that jointly modeling temporal dynamics and road-network spatial relationships is an important means of improving traffic forecasting performance.

### 2.2 Attention-Based Methods

Attention mechanisms assign different weights to inputs according to their relevance to the current task, thereby emphasizing more informative feature representations<sup>[11]</sup>. Velićević et al.<sup>[12]</sup> proposed the Graph Attention Network (GAT), which adaptively weights the contributions of neighboring nodes through self-attention. Zheng et al.<sup>[13]</sup> proposed GMAN, which uses multiple

attention modules to establish relationships between historical observations and target prediction sequences. Guo et al.<sup>[14]</sup> proposed ASTGCN, which models dynamic spatio-temporal interactions in traffic sequences using spatial and temporal attention. Subsequent studies further improved spatio-temporal representation through local-global temporal attention, gated temporal self-attention, and Transformer-based models with spatio-temporal adaptive embedding<sup>[15]</sup>.

Overall, although existing studies have made substantial progress in temporal modeling, spatial dependency characterization, and spatio-temporal fusion, three limitations remain. First, some methods rely primarily on implicit temporal encoding or multi-branch temporal feature concatenation, leading to insufficient use of periodic information. Second, strengthening temporal dependency representation by expanding attention structures often increases the number of parameters and computational overhead. Third, existing studies typically allow the attention branch and the backbone network to participate in joint optimization from the beginning, which may cause training oscillations and unstable convergence. Accordingly, this work investigates unified temporal feature embedding, lightweight temporal attention, and progressive attention activation with dynamic fusion.

## III. CHRONOGRAPH NET MODEL

### 3.1 Problem Definition and Overall Framework

Represent the traffic network as the graph  $G = (V, E, A)$ , where  $V$  denotes the set of traffic monitoring nodes,  $E$  denotes the connections between nodes, and  $A$  is the adjacency matrix describing spatial relationships in the road network.

At each discrete time step  $t$ , the dynamic operating state of the traffic network is represented as a graph signal  $\mathbf{X}_t \in \mathbb{R}^{N \times F}$ , where  $F$  is the node feature dimension, typically including multidimensional traffic-state variables such as traffic flow, travel speed, and traffic density.

Given the historical state sequence over the past  $T$  time steps  $\{\mathbf{X}_{t-T+1}, \dots, \mathbf{X}_{t-1}, \mathbf{X}_t\}$ , together with the traffic network graph  $G$ , traffic forecasting aims to learn a mapping function  $f$  that nonlinearly maps the historical sequence to the future sequence:

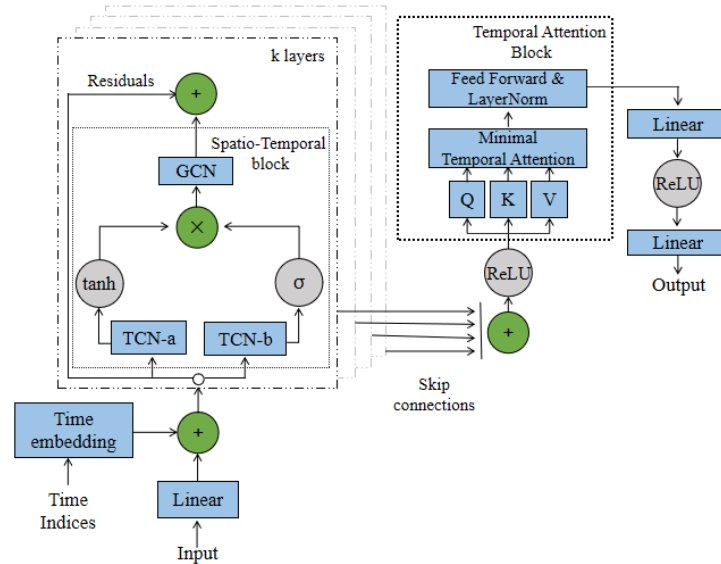
$$\{\mathbf{X}_{t+1}, \dots, \mathbf{X}_{t+H}\} = f(\{\mathbf{X}_{t-T+1}, \dots, \mathbf{X}_t\}, G) \quad (1)$$

where  $H$  denotes the prediction horizon,  $\{\mathbf{X}_{t+1}, \dots, \mathbf{X}_{t+H}\}$  is the traffic-state sequence output by the model for the next  $H$  time steps. This study uses the past 12 time steps to predict the next 12 time steps, i.e., one hour of historical traffic observations to forecast the following hour. The data are chronologically divided into training, validation, and test sets at a ratio of 6:2:2.

ChronoGraphNet mainly consists of unified temporal feature embedding, a spatio-temporal graph convolution backbone, lightweight temporal attention, and progressive attention activation with dynamic fusion. First, the unified temporal feature embedding explicitly injects intraday periodic training by controlling when the attention branch is activated and how its fusion strength changes;

information into the original traffic-state representation to provide temporal priors for subsequent spatio-temporal feature extraction. The spatio-temporal graph convolution backbone then extracts temporal features and models spatial dependencies. Next, lightweight temporal attention further captures correlations among critical temporal positions in the high-level feature space. Finally, dynamic fusion controls the contribution of attention-enhanced features to the final prediction. The main network architecture of ChronoGraphNet is shown in Fig. 2. Progressive attention activation and dynamic fusion mainly operate during model

therefore, this strategy is not shown as a separate network layer in Fig. 2.



**Figure 2. Main network architecture of ChronoGraphNet**

### 3.2 Unified Temporal Feature Embedding

Traffic flow exhibits clear periodic patterns in the temporal dimension. Because the experimental data are sampled at 5-min intervals, one day contains 288 time slots. The positions of the input time steps are first periodicized to construct a standardized intraday periodic index:

$$I_t = t \bmod 288 \quad (2)$$

where  $I_t$  denotes the intraday periodic index corresponding to time step  $t$ , and  $t$  denotes the position of the time step in the input sequence. A learnable temporal embedding layer is then used to represent the discrete periodic positions and map

them to vector representations with the same dimensionality as the residual channels of the backbone network:

$$\mathbf{E}_t = \text{Emb}(I_t) \quad (3)$$

where  $\text{Emb}(\cdot)$  denotes the temporal embedding function, and  $\mathbf{E}_t$  denotes the temporal embedding vector generated from the periodic index  $I_t$ .

The input data are then mapped to the feature space of the backbone network. The temporal embedding vector is broadcast along the node dimension to match the shape of the backbone

features, after which element-wise addition is used to inject temporal information.

### 3.3 Spatio-Temporal Graph Convolution Backbone

The temporally enhanced input is fed into the spatio-temporal graph convolution backbone. The backbone contains multiple stacked spatio-temporal graph convolution layers, each mainly composed of gated temporal convolution, graph convolution, residual connections, batch normalization, and skip connections. Gated temporal convolution extracts local dynamic patterns from traffic sequences, while graph convolution propagates and aggregates spatial dependencies among road nodes using the topology support matrix and adaptive adjacency relationships.

The output of each layer is also passed to a skip branch. After alignment along the temporal dimension, the skip outputs are accumulated layer by layer to form a high-level spatio-temporal representation **ResFinal**. This design integrates features extracted at different depths and provides a foundation for subsequent temporal attention enhancement.

### 3.4 Lightweight Temporal Attention Mechanism

To further capture correlations among critical temporal positions in high-level features, a lightweight temporal attention mechanism is introduced at the output of the backbone network. Unlike standard self-attention, which uses independent projections to generate query, key, and value, the proposed module applies a shared  $1 \times 1$  convolution to jointly project the input features and generate the three representations in a single operation. Let the high-level backbone feature be  $\mathbf{X} \in \mathbb{R}^{B \times C \times N \times L}$ , where  $B$ ,  $C$ ,  $N$ , and  $L$  denote batch size, number of channels, number of nodes, and temporal length, respectively. The mapping is expressed as:

$$[\mathbf{Q}, \mathbf{K}, \mathbf{V}] = \text{Conv}_{1 \times 1}(\mathbf{X}) \quad (4)$$

where  $\mathbf{Q}$ ,  $\mathbf{K}$ , and  $\mathbf{V}$  denote the query, key, and value matrices, respectively. After obtaining  $\mathbf{Q}$ ,  $\mathbf{K}$ , and  $\mathbf{V}$ , the channel dimension is rearranged according to the number of attention heads, and scaled dot-product correlations are computed along the temporal dimension in each subspace to obtain the attention weights:

$$\mathbf{A} = \text{Softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d}}\right) \quad (5)$$

where  $\mathbf{A}$  denotes the temporal attention weight matrix, and  $d$  denotes the channel dimension of each attention head. The obtained attention weights are then used to aggregate the value vectors, yielding the temporal attention-enhanced feature:

$$\Delta\mathbf{X} = \mathbf{A}\mathbf{V} \quad (6)$$

where  $\Delta\mathbf{X}$  denotes the incremental feature output by the temporal attention module. The shared convolution jointly generates query, key, and value representations in a single projection, thereby reducing structural redundancy and additional parameter overhead caused by three independent projections while retaining temporal dependency modeling capability. The module is complementary to the unified temporal feature embedding: the former learns data-driven temporal correlations in the high-level feature space, whereas the latter provides explicit periodic priors at the input.

### 3.5 Progressive Attention Activation and Dynamic Fusion

To reduce interference from the attention branch during the early optimization of the backbone network, a staged "freeze-unlock" mechanism is first adopted. At initialization, the learnable parameters of the lightweight temporal attention module are temporarily frozen, allowing the model to first learn basic spatio-temporal patterns using the temporal feature embedding and the spatio-temporal graph convolution backbone. An automatic activation threshold  $E_{\text{auto}}$  is defined for the attention branch. As training proceeds, when epoch  $e$  reaches the preset unfreezing threshold for the first time, the attention branch is automatically unlocked and participates in subsequent joint training. This stage-control process is summarized as:

$$\text{Status}_{\text{attn}} = \begin{cases} \text{Frozen,} & e < E_{\text{auto}} \\ \text{Active,} & e \geq E_{\text{auto}} \end{cases} \quad (7)$$

During the Frozen stage, the attention branch remains frozen, reducing disturbances to the backbone feature distribution caused by randomly initialized attention weights. After the attention branch is unlocked, an epoch-level warm-up and batch-level fine-tuning strategy is used to progressively increase the contribution of the attention-enhanced features during optimization. Let the original backbone feature be  $\mathbf{X}_{\text{base}}$ , the feature enhanced by lightweight temporal attention

be  $\mathbf{X}_{\text{attn}}$ , and the final fused feature be  $\mathbf{X}_{\text{out}}$ . The fusion is expressed as:

$$\mathbf{X}_{\text{out}} = (1 - \alpha_{e,b}) \cdot \mathbf{X}_{\text{base}} + \alpha_{e,b} \cdot \mathbf{X}_{\text{attn}} \quad (8)$$

where the fusion coefficient  $\alpha_{e,b}$  increases dynamically with the training process and is jointly determined by the current epoch  $e$  and batch-step count  $b$ . A small fusion coefficient of 0.01 is used at the beginning of training so that the model mainly relies on the backbone to learn basic spatio-temporal patterns. As training proceeds, the coefficient is gradually increased to 0.03, allowing the attention branch to participate in feature optimization more smoothly. When the model enters a relatively stable stage, the coefficient is further increased and capped at 0.05 to strengthen the supplementary modeling of critical temporal dependencies while avoiding training oscillations caused by an excessive weakening the basic representation learned by the backbone, thereby balancing training stability and feature enhancement.

$$\alpha_{e,b} = \begin{cases} 0.01, & e < 5 \\ 0.01 + 0.02 \cdot \frac{e-4}{6}, & 5 \leq e < 10 \\ 0.03, & e \geq 10, b < 2000 \\ 0.03 + 0.02 \cdot \frac{b-2000}{2000}, & e \geq 10, 2000 \leq b < 4000 \\ 0.05, & e \geq 10, b \geq 4000 \end{cases} \quad (9)$$

#### IV. EXPERIMENTS AND RESULTS

##### 4.1 Datasets and Experimental Settings

Experiments are conducted on three real-world traffic datasets: PeMS04, PeMS08, and METR-LA. The datasets differ in road-network scale, topological complexity, and temporal coverage, allowing the model to be evaluated under different traffic scenarios. All data are chronologically divided into training, validation, and test sets at a ratio of 6:2:2, and a multi-step setting using the previous 12 time steps to predict the next 12 time steps is adopted. Table 1 summarizes the basic information of the three datasets.

The model is trained using the Adam optimizer with an initial learning rate of 0.002 and a weight decay of  $1 \times 10^{-4}$ . The training batch size is 16, while the validation and test batch sizes are both 64.

contribution of attention features. Accordingly, the dynamic fusion coefficient  $\alpha_{e,b}$  is defined in Equation (9).

The dynamic fusion coefficient  $\alpha_{e,b}$  is constrained to the range 0.01-0.05 and increases progressively throughout training. Here,  $e$  denotes the current training epoch, and  $b$  denotes the accumulated number of batch steps after entering batch-level fine-tuning. At the beginning of training,  $\alpha_{e,b}=0.01$  and the attention branch contributes only a small weight to feature fusion. During epoch-level warm-up,  $\alpha_{e,b}$  increases linearly from 0.01 to 0.03; during batch-level fine-tuning,  $\alpha_{e,b}$  further increases smoothly from 0.03 to 0.05 and remains fixed after reaching the upper bound. Because the fusion ratio of the attention branch is kept small, attention-enhanced information can be gradually introduced without

MultiStepLR is used to decay the learning rate at epochs 40, 70, and 90 with a decay factor of 0.5. The maximum number of training epochs is 100. Early stopping based on validation MAE is applied: training stops and the best model is saved when the validation MAE improves by less than 0.01 for 12 consecutive epochs. The lightweight temporal attention module uses two attention heads. MAE, RMSE, and MAPE are adopted as evaluation metrics

**Table 1. Basic information of the experimental datasets**

| Dataset | Nodes | Edges | Time Steps | Time Span          |
|---------|-------|-------|------------|--------------------|
| PeMS04  | 307   | 340   | 16992      | 2018.1.1–2018.2.28 |

| Dataset | Nodes | Edges | Time Steps | Time Span          | Dataset | Nodes | Edges | Time Steps | Time Span          |
|---------|-------|-------|------------|--------------------|---------|-------|-------|------------|--------------------|
| PeMS08  | 170   | 295   | 17856      | 2016.7.1–2016.8.31 | METR-LA | 207   | 1313  | 29280      | 2012.3.1–2012.6.30 |

**Table 2. Comparison of Prediction Performance Between ChronoGraphNet and Baseline Models on PeMS04 and PeMS08 Datasets**

| Model                        | PeMS04         |              |              | PeMS08         |              |             |
|------------------------------|----------------|--------------|--------------|----------------|--------------|-------------|
|                              | MAE<br>MAPE(%) | RMSE         |              | MAE<br>MAPE(%) | RMSE         |             |
| HA <sup>[18]</sup>           | 38.03          | 59.24        | 27.88        | 34.86          | 52.04        | 24.07       |
| VAR <sup>[19]</sup>          | 24.54          | 38.61        | 17.24        | 19.19          | 29.81        | 13.10       |
| GRU-ED <sup>[20]</sup>       | 23.68          | 39.27        | 16.44        | 22.00          | 36.23        | 13.33       |
| DSANet <sup>[21]</sup>       | 22.79          | 35.77        | 16.03        | 17.14          | 26.96        | 11.32       |
| STGCN <sup>[5]</sup>         | 21.16          | 34.89        | 13.83        | 17.50          | 27.09        | 11.29       |
| ASTGCN <sup>[14]</sup>       | 22.93          | 35.22        | 16.56        | 18.25          | 28.06        | 11.64       |
| STSGCN <sup>[8]</sup>        | 21.19          | 33.65        | 13.90        | 17.13          | 26.86        | 10.96       |
| DCRNN <sup>[6]</sup>         | 19.74          | 31.29        | 13.46        | 16.82          | 26.36        | 10.92       |
| Graph WaveNet <sup>[7]</sup> | 24.89          | 39.66        | 17.29        | 18.28          | 30.05        | 12.15       |
| DSTCGCN <sup>[22]</sup>      | <u>19.01</u>   | <u>31.03</u> | <u>12.64</u> | <u>15.18</u>   | <u>24.49</u> | <u>9.68</u> |
| TraverseNet <sup>[23]</sup>  | 19.86          | 31.54        | 14.38        | 15.68          | 24.62        | 10.87       |
| DGCRN <sup>[24]</sup>        | 20.39          | 32.34        | 14.64        | 16.22          | 26.10        | 12.06       |
| DSTAGNN <sup>[10]</sup>      | 19.30          | 31.46        | 12.70        | 15.67          | 24.77        | 9.94        |
| RGSL <sup>[25]</sup>         | 19.19          | 31.14        | 12.69        | 15.49          | 24.80        | 9.96        |
| ChronoGraphNet               | <b>18.43</b>   | <b>29.94</b> | <b>12.34</b> | <b>14.45</b>   | <b>23.66</b> | <b>9.22</b> |

#### 4.2 Comparison with Baseline Models

To evaluate the overall forecasting performance of ChronoGraphNet, it is compared with HA<sup>[18]</sup>, VAR<sup>[19]</sup>, GRU-ED<sup>[20]</sup>, DSANet<sup>[21]</sup>, STGCN<sup>[5]</sup>, ASTGCN<sup>[14]</sup>, STSGCN<sup>[8]</sup>, DCRNN<sup>[6]</sup>, Graph WaveNet<sup>[7]</sup>, DSTCGCN<sup>[22]</sup>, TraverseNet<sup>[23]</sup>, DGCRN<sup>[24]</sup>, DSTAGNN<sup>[10]</sup>, and RGSL<sup>[25]</sup>.

The results show that ChronoGraphNet

achieves an MAE of 18.43, an RMSE of 29.94, and a MAPE of 12.34% on PeMS04, and an MAE of 14.45, an RMSE of 23.66, and a MAPE of 9.22% on PeMS08. Under the current experimental setting, ChronoGraphNet achieves superior overall forecasting performance compared with most of the baseline models.

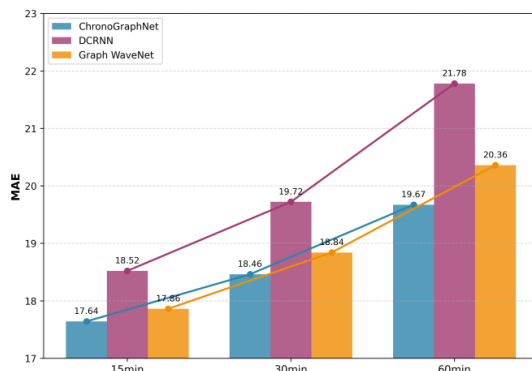
**Table 3. Prediction Performance Comparison Between ChronoGraphNet and Baseline Models on PeMS04 and METR-LA Datasets with Different Time Steps**

| Dataset / H | Metric   | VAR <sup>[19]</sup> | STGCN <sup>[5]</sup> | AGCRN <sup>[9]</sup> | GMA <sup>N[13]</sup> | MTGNN <sup>N[26]</sup> | DCRN <sup>N[6]</sup> | Graph WaveNet <sup>t[7]</sup> | ChronoGraphNet |
|-------------|----------|---------------------|----------------------|----------------------|----------------------|------------------------|----------------------|-------------------------------|----------------|
| PeMS04 H3   | MAE      | 21.94               | 18.74                | 18.52                | 18.27                | 18.65                  | 18.52                | <u>17.86</u>                  | <b>17.64</b>   |
|             | RMSE     | 34.30               | 29.84                | 29.79                | 29.25                | 30.13                  | 29.49                | <u>28.84</u>                  | <b>28.58</b>   |
|             | MAPE (%) | 16.42               | 14.42                | <u>12.31</u>         | 12.66                | 13.32                  | 12.68                | 12.36                         | <b>11.84</b>   |
| PeMS04 H6   | MAE      | 23.72               | 19.64                | 19.45                | <u>18.81</u>         | 19.48                  | 19.72                | 18.84                         | <b>18.46</b>   |
|             | RMSE     | 36.58               | 31.34                | 31.45                | 30.85                | 32.02                  | 31.27                | <u>30.46</u>                  | <b>30.01</b>   |
|             | MAPE (%) | 18.02               | 13.27                | <u>12.82</u>         | 13.25                | 14.08                  | 13.41                | 12.92                         | <b>12.34</b>   |
| PeMS04 H12  | MAE      | 26.76               | 21.12                | 20.64                | <u>20.01</u>         | 20.96                  | 21.78                | 20.36                         | <b>19.67</b>   |
|             | RMSE     | 40.28               | 33.53                | 33.31                | <b>31.32</b>         | 34.66                  | 34.03                | 32.76                         | <u>31.79</u>   |
|             | MAPE (%) | 20.94               | 14.22                | 13.74                | <u>13.40</u>         | 14.96                  | 14.92                | 13.98                         | <b>13.22</b>   |
| METR-LA H3  | MAE      | 4.42                | <u>2.77</u>          | 2.88                 | <b>2.69</b>          | 2.87                   | 2.80                 | <b>2.69</b>                   | <b>2.69</b>    |
|             | RMSE     | 7.89                | 5.38                 | 5.74                 | <b>5.15</b>          | 5.58                   | 5.55                 | 5.18                          | <u>5.16</u>    |
|             | MAPE (%) | 10.20               | 7.30                 | 7.62                 | <u>6.90</u>          | 7.70                   | 7.41                 | <b>6.86</b>                   | 6.96           |
| METR-LA H6  | MAE      | 5.41                | 3.15                 | 3.47                 | 3.07                 | 3.23                   | 3.12                 | <u>3.05</u>                   | <b>3.04</b>    |
|             | RMSE     | 9.13                | 6.45                 | 7.24                 | 6.22                 | 6.58                   | 6.49                 | <u>6.17</u>                   | <b>6.14</b>    |
|             | MAPE (%) | 12.70               | 8.80                 | 9.57                 | 8.37                 | 9.00                   | 8.73                 | <b>8.19</b>                   | <u>8.34</u>    |
| METR-LA H12 | MAE      | 6.52                | 3.60                 | 4.59                 | 3.53                 | 3.62                   | <b>3.44</b>          | <u>3.49</u>                   | <b>3.44</b>    |
|             | RMSE     | 10.11               | 7.60                 | 9.40                 | 7.37                 | 7.51                   | 7.35                 | <u>7.23</u>                   | <b>7.18</b>    |
|             | MAPE (%) | 15.80               | 10.50                | 12.70                | 10.01                | 10.38                  | 10.07                | <b>9.87</b>                   | <u>9.90</u>    |

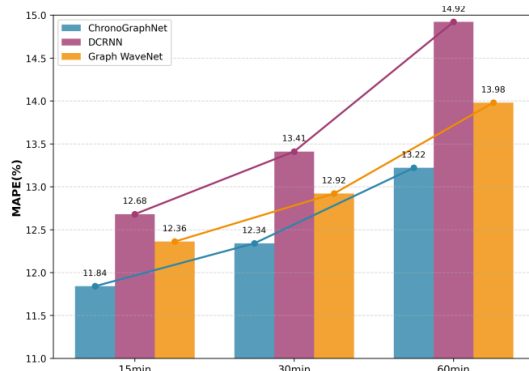
#### 4.3 Analysis at Different Prediction Horizons

To further evaluate performance over different forecasting ranges, 15-min, 30-min, and 60-min prediction results are examined and

compared with VAR<sup>[19]</sup>, STGCN<sup>[5]</sup>, AGCRN<sup>[9]</sup>, GMA<sup>N[13]</sup>, MTGNN<sup>[26]</sup>, DCRNN<sup>[6]</sup>, and Graph WaveNet<sup>[7]</sup>.



(a) MAE comparison



(b) MAPE comparison

**Figure 3. MAE and MAPE comparison of ChronoGraphNet, DCRNN, and Graph WaveNet at different prediction horizons**

On PeMS04, the MAE values of ChronoGraphNet for 15-min, 30-min, and 60-min prediction are 17.64, 18.46, and 19.67, respectively, while the corresponding MAPE values are 11.84%, 12.34%, and 13.22%. As the prediction horizon increases, the errors of all models generally increase, but ChronoGraphNet still maintains good performance for medium- and long-horizon forecasting. On METR-LA, ChronoGraphNet remains highly competitive in MAE and RMSE across different horizons, with several metrics achieving the best or tied-best results; however, its MAPE is slightly inferior to that of individual baselines at some horizons. This indicates that the degree of performance improvement varies across datasets and evaluation metrics. For a more intuitive view of the error trends over increasing prediction horizons, Fig. 3 further compares the MAE and MAPE of ChronoGraphNet, DCRNN, and Graph WaveNet.

As the prediction horizon increases, the errors of all three models rise, while ChronoGraphNet generally maintains lower error values and preserves good forecasting performance over longer horizons.

**4.4 Ablation Study**

To evaluate the effectiveness of the core components of ChronoGraphNet, ablation experiments are conducted on PeMS04. TE denotes unified temporal feature embedding, Attn denotes lightweight temporal attention, and Prog. denotes progressive attention activation and dynamic fusion.

**Table 4. Ablation results of the core components of ChronoGraphNet on PeMS04**

| Configuration               | MAE   | RMSE  | MAPE(%) |
|-----------------------------|-------|-------|---------|
| ChronoGraphNet              | 18.43 | 29.94 | 12.34   |
| w/Attn, w/o TE, w/o Prog.   | 18.66 | 30.09 | 12.70   |
| w/Attn+Prog., w/o TE        | 18.46 | 30.01 | 12.64   |
| w/Attn+TE, w/o Prog.        | 18.44 | 29.98 | 12.60   |
| w/o Attn, w/o TE, w/o Prog. | 18.72 | 30.15 | 12.91   |

Without the three enhancement components, the model obtains an MAE of 18.72. Adding only lightweight temporal attention reduces the MAE to 18.66. Further introducing temporal feature embedding without the progressive strategy lowers the MAE to 18.44. When progressive attention activation and dynamic fusion are added to the configuration containing only the attention branch, the MAE decreases from 18.66 to 18.46, the RMSE from 30.09 to 30.01, and the MAPE from 12.70% to 12.64%, indicating that the training strategy provides an independent optimization benefit. The complete ChronoGraphNet further achieves the lowest MAE of 18.43, while also obtaining the best RMSE and MAPE. These results show that unified temporal feature embedding, lightweight temporal attention, and progressive attention activation with dynamic fusion each contribute positively to model performance and act complementarily in the complete model.

## V. CONCLUSION

This paper proposes ChronoGraphNet to address insufficient exploitation of temporal periodic priors, the cost of modeling critical temporal dependencies, and the instability of jointly training attention branches in traffic flow prediction. The model explicitly introduces intraday periodic information through unified temporal feature embedding, enhances critical temporal dependencies using lightweight temporal attention, and controls the training and fusion of the attention branch through progressive attention activation and dynamic fusion.

Experiments on the PeMS04, PeMS08, and METR-LA datasets demonstrate that ChronoGraphNet achieves good forecasting performance in both overall and multi-horizon prediction. Ablation experiments further show that all three core components make positive contributions and act complementarily in the complete model. Meanwhile, the degree of improvement varies across datasets and prediction horizons. Future work may further investigate dynamic spatial relationship modeling, fusion of external factors, adaptation to abnormal traffic states, and cross-region generalization.

## REFERENCES

- [1] Yin C T, Xiong Z, Chen H, et al. A literature survey on smart cities[J]. *Science China Information Sciences*, 2015, 58(10): 1-18.
- [2] Mao Y, Zhou H, Chen L, et al. A survey on spatio-temporal prediction: From transformers to foundation models[J]. *ACM Computing Surveys*, 2025, 58(4): 1-36.
- [3] Tedjopurnomo D A, Bao Z, Zheng B, et al. A survey on modern deep neural network for traffic prediction: Trends, methods and challenges[J]. *IEEE Transactions on Knowledge and Data Engineering*, 2020, 34(4): 1544-1561.
- [4] Yin X, Wu G, Wei J, et al. Deep learning on traffic prediction: Methods, analysis, and future directions[J]. *IEEE Transactions on Intelligent Transportation Systems*, 2021, 23(6): 4927-4943.
- [5] Yu B, Yin H, Zhu Z. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting[J]. *arXiv preprint arXiv:1709.04875*, 2017.
- [6] Li Y, Yu R, Shahabi C, et al. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting[J]. *arXiv preprint arXiv:1707.01926*, 2017.
- [7] Wu Z, Pan S, Long G, et al. Graph wavenet for deep spatial-temporal graph modeling[J]. *arXiv preprint arXiv:1906.00121*, 2019.
- [8] Song C, Lin Y, Guo S, et al. Spatial-temporal synchronous graph convolutional networks: A new framework for spatial-temporal network data forecasting[C]//*Proceedings of the AAAI conference on artificial intelligence*. 2020, 34(01): 914-921.
- [9] Bai L, Yao L, Li C, et al. Adaptive graph convolutional recurrent network for traffic forecasting[J]. *Advances in neural information processing systems*, 2020, 33: 17804-17815.
- [10] Lan S, Ma Y, Huang W, et al. Dstagnn: Dynamic spatial-temporal aware graph neural network for traffic flow forecasting[C]//*International conference on machine learning*. PMLR, 2022: 11906-11917.
- [11] Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate[J]. *arXiv preprint arXiv:1409.0473*, 2014.
- [12] Veličković P, Cucurull G, Casanova A, et al. Graph attention networks[J]. *arXiv preprint arXiv:1710.10903*, 2017.
- [13] Zheng C, Fan X, Wang C, et al. Gman: A graph multi-attention network for traffic prediction[C]//*Proceedings of the AAAI conference on artificial intelligence*. 2020, 34(01): 1234-1241.
- [14] Guo S, Lin Y, Feng N, et al. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting[C]//*Proceedings of the AAAI conference on artificial intelligence*. 2019, 33(01): 922-929.
- [15] He X, Zhang W, Li X, et al. Tea-gcn: Transformer-enhanced adaptive graph convolutional network for traffic flow forecasting[J]. *Sensors*, 2024, 24(21): 7086.
- [16] Geng Z, Xu J, Wu R, et al. STGAFormer: Spatial - temporal gated attention transformer based graph neural network for traffic flow forecasting[J]. *Information Fusion*, 2024, 105: 102228.
- [17] Liu H, Dong Z, Jiang R, et al. Staformer: Spatio-temporal adaptive embedding makes vanilla transformer SOTA for traffic forecasting[J]. *arXiv preprint arXiv:2308.10425*, 2023.
- [18] Jing L, Wei G. A Summary of Traffic Flow Forecasting Methods[J]. *Journal of Highway and Transportation Research and Development*, 2004, 21: 82-85.
- [19] Schimbschi F, Moreira-Matias L, Nguyen V X, et al. Topology-regularized universal vector autoregression for traffic forecasting in large urban areas[J]. *Expert Systems with Applications*, 2017, 82: 301-316.
- [20] Cho K, Van Merriënboer B, Gulçehre Ç, et al. Learning phrase representations using RNN encoder - decoder for statistical machine translation[C]//*Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 2014: 1724-1734.
- [21] Huang S, Wang D, Wu X, et al. DSANet: Dual self-attention network for multivariate time series forecasting[C]//*Proceedings of the 28th ACM international conference on information and knowledge management*. 2019: 2129-2132.

- [22] Wu B, Chen L. Dstcgen: Learning dynamic spatial-temporal cross dependencies for traffic forecasting[J]. arXiv preprint arXiv:2307.00518, 2023.
- [23] Wu Z, Zheng D, Pan S, et al. TraverseNet: Unifying space and time in message passing for traffic forecasting[J]. IEEE Transactions on Neural Networks and Learning Systems, 2022, 35(2): 2003-2013.
- [24] Li F, Feng J, Yan H, et al. Dynamic graph convolutional recurrent network for traffic prediction: Benchmark and solution[J]. ACM Transactions on Knowledge Discovery from Data, 2023, 17(1): 1-21.
- [25] Yu H, Li T, Yu W, et al. Regularized graph structure learning with semantic knowledge for multi-variables time-series forecasting[J]. arXiv preprint arXiv:2210.06126, 2022.
- [26] Wu Z, Pan S, Long G, et al. Connecting the dots: Multivariate time series forecasting with graph neural networks[C]//Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining. 2020: 753-763.