Journal of Computers Vol. 36 No. 4, August 2025, pp. 31-45
https://doi.org/10.63367/19911599202508360400 2

A Long Term Transformer-based Spatiotemporal Graph Attention Network for Traffic Flow Forecasting

Lin Xiao[1*] and Hongchao Chen[2]

[1] Fujian University of Technology,
Fuzhou 350118, China
66246297@qq.com

[2] Fujian Provincial Key Laboratory of Big Data Mining and Applications,
Fuzhou 350118, China
chenhongchao2023@163.com

Received 23 July 2024; Revised 13 November 2024; Accepted 16 December 2024

Abstract. Traffic flow prediction is the key to accurate urban traffic control and the basis for developing intelligent transportation systems. Recent studies have made substantial progress in traffic prediction by modelling complex spatiotemporal graph topology and considering sensors as road network nodes. However, the current spatiotemporal graph neural network model is limited by its structure. It can only utilize short-range traffic flow data and cannot effectively extract the long-term trend of complex traffic flow and periodic features in traffic patterns. To address the above problems, we propose a Transformer-based long-term traffic flow prediction framework, "Transformer-based spatiotemporal graph attention network". First, the model utilizes the Transformer coding layer to learn compressed and context-rich subsequence temporal representations from long-term sequences. Then, the model designs a multi-scale gated temporal convolution module to identify and extract long-term trend features of traffic flow from the subsequence time representations. Next, the model constructs a multi-granularity random graph attention module to capture the periodic features of traffic flow from the subsequence time representations and extracts the short-term trend features present in the long-time series using the STGNN model. Finally, the model fuses the extracted long-term trends, periodic features and short-term trends to obtain the final prediction results. Experimental results on two real-world traffic flow datasets show that the model outperforms the baseline model and makes accurate long-term predictions.

Keywords: traffic forecasting, intelligent transportation systems, graph attention, periodic features

1 Introduction

With the rapid development of urbanization, Intelligent Transportation Systems (ITS) play an important role in people's daily travel experience, and one of the key components of ITS is the traffic flow prediction system. In ITS [1], the main function of traffic prediction is to use historical traffic data captured by sensors to predict future traffic conditions [2, 3]. Accurate traffic prediction enables drivers to strategize their travel routes in advance, reducing road delays. Also, it assists authorities in deploying personnel in advance to manage traffic in congested areas.

Predicting traffic flow faces great challenges due to its complex spatial dependence and nonlinear temporal dynamics. Traditional traffic flow prediction techniques, such as ARIMA [4] and VAR [5], require strict assumptions on the traffic dataset. However, the complexity of real-world data often makes these assumptions untenable, thus severely limiting the applicability of these models. In addition, these traditional methods are insufficient to capture complex nonlinear time dependencies, nor can they represent the spatial dependencies inherent in transportation networks.

Urban planners have much traffic data in the era of advanced sensing and data processing technologies. Due to the rapid development of deep learning, many methods for traffic flow prediction using deep learning and big data have emerged, significantly improving prediction accuracy compared to traditional methods. The outstanding performance proves the great potential of data-driven systems based on deep learning in predicting traffic flow. These methods are currently the most widely used prediction techniques. Earlier spatiotemporal prediction

* Corresponding Author

techniques utilizing CNNs to capture spatial interdependencies outperformed traditional methods [6, 7]. Given that traffic networks are essentially non-Euclidean graph structures, researchers have begun to use graph neural networks (GNNs) to model spatial relationships in traffic networks. Combining GNNs with RNNs, one-dimensional (1D) CNNs, and other temporal models allows the development of spatiotemporal graph neural network models for accurate traffic flow prediction [8, 9].

Recent studies have shown that Spatio-Temporal Graph Neural Networks (STGNN) successfully predict traffic flow trends by simultaneously considering both variation over time and spatial relationships [10-14]. Examples of cutting-edge spatiotemporal graphical neural networks (STGNN) include D2STGNN [11], Graph WaveNet [12], MTGNN [13], and GTS [15]. However, it is worth noting that such models usually have limitations. They tend to rely on short-term historical series as model inputs, e.g., they typically restrict traffic flow data to 12 time steps or 1 hour. However, the long-term trends and cyclical features present in historical traffic data are often not adequately captured by the information contained in short-term data. STEP [10] investigated a potential pre-training method to increase the receptive field of STGNNs in order to obtain longer temporal information. PatchTST [16] extends the range of historical dependencies by segmenting the time series into subsequence patches and employing a channel-independent strategy. However, the above models are unable to accurately capture long-term trends and cyclical features in long history series. As shown in Fig. 1, the green box indicates the traffic flow trend on Wednesdays, while the red box indicates the traffic flow trend on Saturdays, and the traffic flow trends in the same color boxes all exhibit a high degree of similarity. In addition, the traffic flow trends for Wednesday and Thursday within the same week also show a high degree of similarity. Therefore, we can infer that the daily and weekly traffic flow trends exhibit significant periodicity. Considering the long-term trend characteristics and periodicity features in long historical series can directly improve the accuracy of traffic flow prediction.

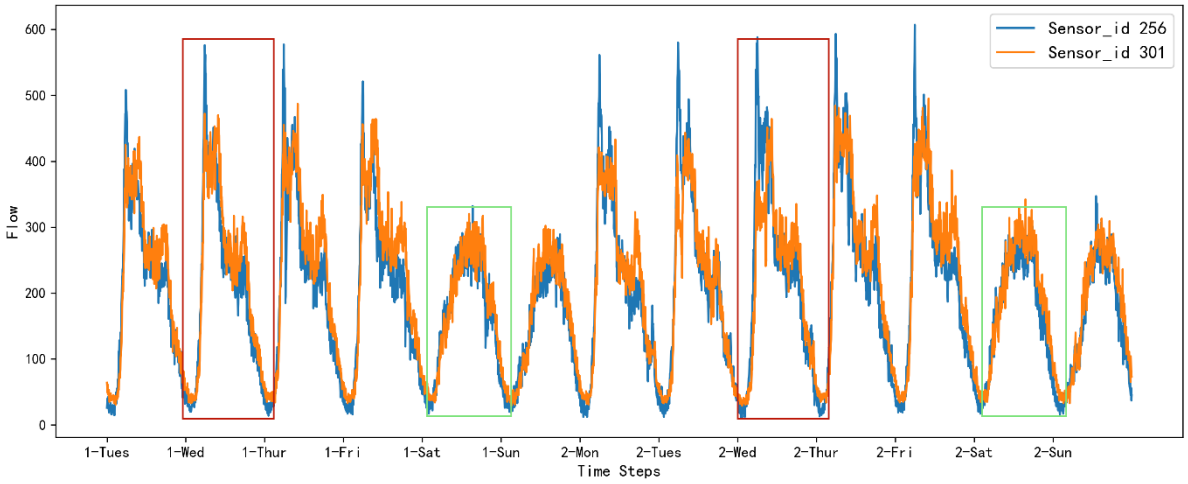

Fig. 1. Example of historical time series data of traffic flow

To address the problem of insufficient information embedded in short-term historical series, we incorporate extended historical series into the traffic flow prediction model. Specifically, we use the traffic flow data from the previous two weeks as input for the model. This allows the model to accurately predict future traffic flow trends from the rich information of the long-term historical series. Second, in order to solve the problem of not being able to effectively extract long-term trends and cyclical features from long-term historical series. We propose a spatiotemporal graph attention network based on a long-term Transformer. It can not only capture short-term trends in long-term historical series but also effectively extract long-term trend features and periodic features from long-term historical series. The contribution of this paper can be briefly described as follows:

- We propose a framework for traffic flow prediction that combines the short-term trends, long-term trends, and periodic features of extensive historical sequences and can accurately identify and extract long-term trends and periodic features from long-term historical sequences.
- We propose a multi-scale gated temporal convolution module that can identify and extract features of long-term trends in traffic flows from hour-level subsequences temporal representations.

- We propose a multi-granularity random graph attention module that can extract periodic features of traffic flow from hour-level subsequences temporal representations of the previous two weeks and days.
- We assess the proposed structure by using two genuine traffic datasets. The experimental results demonstrate that our suggested model surpasses the baseline model in all prediction ranges.

Overall, we propose a network framework for traffic flow prediction and design a multi-scale gated temporal convolutional module and a multi-granularity random graph attention module in this framework. The multi-scale gated temporal convolution module can accurately capture the long-term trend of traffic flow from long-term historical sequences. The multi-granularity random graph attention module can effectively extract periodic features from long-term historical sequences. This allows the framework to effectively solve the problem of being unable to extract long-term trends and periodic features from long-term historical sequences.

Subsequent portions of this work are organized as follows: Section 2 presents a comprehensive summary of previous techniques employed in forecasting traffic patterns. Section 3 presents a formal description of the objective of predicting traffic flow and explains the overarching framework and design specifics. The proposed approach is evaluated in Section 4 through the execution of a variety of experimental sets. The final section of this work is Section 5, which provides potential avenues for additional research.

2 Related Work

The majority of the first methods employed to predict traffic flow rely on statistical techniques, including the historical average (HA) model [17], the ARIMA model [4], and the Kalman filter [18]. The historical average (HA) model relies on the inherent periodic characteristics of time series data and performs predictive analysis by counting the average values of corresponding time points in the past. ARIMA combines the autoregression (AR) and moving average (MA) models while handling the non-stationarity of time series through differencing. The Kalman filter realizes state estimation through two steps: prediction and update of system state. Early traffic flow prediction models generally tend to use time series analysis techniques. However, they often ignore the complex interaction effects between roads and the significant spatial correlation of traffic flow data.

Recently, there has been a growing trend in the field of traffic flow prediction to use neural networks to capture spatial and temporal correlations of traffic flows and to train models using large datasets. This approach has shown superior performance compared to earlier methods [19]. In some technical applications, Convolutional Neural Networks (CNNs) have been used to capture and model spatial correlations in traffic data in the form of grids [20, 21]. Although, these methods can capture the spatial correlation of traffic flows using Euclidean distance and spatial location between roads. However, they do not adequately consider the topology between roads. The graph structure, on the other hand, can effectively represent the topological relationship of the road network, and this method is not only intuitive but also reflects the connectivity between roads. The distance between roads can be portrayed by the weight parameters of the edges in the graph. Therefore, Graph Neural Networks (GNN) have successfully addressed the shortcomings of early deep learning techniques in dealing with non-Euclidean data structures. Nowadays, graph neural networks (GNNs) have been widely used for several tasks, including spatial and temporal feature prediction [22-24].

Spatio-temporal graph neural networks (STGNNs) are regarded as the most promising method because of their exceptional performance. They combine graph neural networks (GNNs) [25] with time series models [26, 27] to effectively simulate both spatial and temporal interdependence. STGNNs that exploit convolution operations to capture temporal features in a parallel manner encompass STGCN [9], Graph WaveNet [12], MTGNN [13], and StemGNN [14]. These models integrate graph convolutional networks and gated temporal convolutional networks, along with their corresponding variations. To better represent the intricate temporal characteristics, STGNNs like DCRNN [8], ST-MetaNet [28], TGCN [29] and AGCRN [30] employ convolutional networks and recurrent neural networks [26, 31], together with their corresponding variations, to capture the temporal dimension. Furthermore, it is essential to mention that the attention mechanism has been extensively employed in diverse spatiotemporal prediction models, such as GMAN [32] and ASTGCN [33].

D2STGNN [11] is the latest advancement in STGNN design. It improves modelling performance by separating diffuse and intrinsic traffic information. This is especially important considering the unique properties of traffic data. In addition, MegaCRN [34] employs a meta-learning approach that allows the model to adapt to various graphical architectures and time series patterns. This approach can enhance the generalization of the model by modifying its meta-parameters. MAGNN [35] employs a multi-scale feature extraction approach that utilizes graph convolutional layers at different scales to capture dependencies at different temporal granularities. Despite

the significant progress made by STGNN, its complexity is still high due to the need to address both temporal and spatial dependencies at each step. Due to the increased complexity, STGNNs cannot handle long-term historical temporal data. Most STGNNs are only capable of handling short-term historical time data, which typically covers the previous hour and consists of 12-time steps. As a result, contemporary STGNNs face difficulties in recognizing and mining these long-term trends and periodic features that occur over weeks. In order to learn spatial dependencies from long-term historical time series, STEP [10] provides the graph structure learning module with subsequence representations derived via Transformer. PatchTST [16] proposes a new approach that extends the range of historical dependencies by dividing lengthy time-series data into smaller chunks and employing a channel-independent strategy. Crossformer [36, 37] aims to extract traffic flow features from long-time history series by converting the past into segment embedding using the Transformer model. However, these models do not adequately consider the long-term trends and cyclical features unique to long sequences.

Despite significant advances in STGNN (e.g., D2STGNN and MegaCRN) and Transformer-based approaches (e.g., STEP, PatchTST, and Crossformer), these models are either incapable of dealing with long-term historical time series data or are unable to capture long-term trends and periodic features efficiently. To address these issues, our work introduces long-term historical time series and designs two new modules that can effectively extract long-term trends and periodic features. This approach allows for more comprehensive mining of traffic flow patterns from long-term historical series, surpassing the limitations of existing methods.

Compared with the D2STGNN and MegaCRN models, our approach achieves a more comprehensive utilization of time series information by introducing long-term historical sequences up to two weeks, which enhances the model's ability to capture long-term traffic flow patterns. Compared with the STEP method, our method can extract rich information from long-term time series by using multi-scale gated temporal convolution modules and achieves effective extraction of long-term trend features. The PatchTST and Crossformer methods do not model the cyclical features of traffic flow. Compared with the PatchTST and Crossformer methods, our method can effectively extract periodic features from long time series by specially designing the multi-granularity random graph attention module, which makes the model perform better when facing more extended time series (e.g., two-week data).

3 Methodology

The entire model framework is shown in Fig. 2. After the long-term historical sequence data is input into the model, it will first be divided into multiple hour-level subsequences. Next, we use Transformer encoding layers to obtain the contextual representation of the hour-level subsequences from the long-term historical sequence. After that, all hour-level subsequences will be passed to the multi-scale gated temporal convolution module and the multi-granularity random graph attention module, and the last hour-level subsequence will be passed to the STGNN module. In the end, the outcomes of the three modules are merged via feature fusion to provide the final predictions of the model.

3.1 Transformer Encoding Layers

At the T_hth time step, the historical dataset $\mathcal{X}$ is denoted as $\mathcal{X} = [X_{t-T_h+1}, \ldots, X_{t-1}, X_t] \in R^{T_h \times N \times C}$ where X_t is a matrix of size $R^{N \times C}$ and represents the observation data of N sensors at time t. The initial procedure is to divide $\mathcal{X}$ into P consecutive segments, described as a set S: $S = [S_0, S_1, \ldots, S_P]$. Each segment $S_P \in R^{L \times N \times C}$ consists of L data points, each of which corresponds to one hour of data. Specifically, each hour contains 12 time-step observations. If the number of data points in the time series does not reach the preset threshold $T_h = P \times L$, the model will expand its length to the standard specification of $P \times L$ by adding zero values to the left side of the input data. Subsequently, the sequence S will be input into the Transformer encoding layer, the fundamental purpose of which is to enable the model to effectively learn compressed, context-rich hour-level subsequence representations from long time series.

The Transformer encoding layer first embeds the input data, that is, applies a linear transformation to it, converting it into the latent space.

$$U_j^i = W \cdot S_j^i + b\,, \tag{1}$$

where $W \in R^{d\times(LC)}$ and $b \in R^d$ are learnable parameters, $U_j^i \in R^d$ is the model input vector, and d is the hidden dimension.

Next, we add sequential information using a learnable positional encoding layer:

$$U^0 = U + Pos\,. \tag{2}$$

Finally, we use two transformer blocks as our Transformer Blocks. For the lth Transformer Blocks:

$$H^{l+1} = \mathrm{LayerNorm}\left(H^l + \mathrm{MSA}\left(H^l\right)\right), \tag{3}$$

$$H^l = \mathrm{LayerNorm}\left(U^l + \mathrm{FFN}\left(U^l\right)\right), \tag{4}$$

where MSA refers to a network architecture that processes information through multiple attention heads, and FFN is an architecture that applies a fully connected feedforward network to each position of the input separately. For more information, see the original Transformer paper [27].

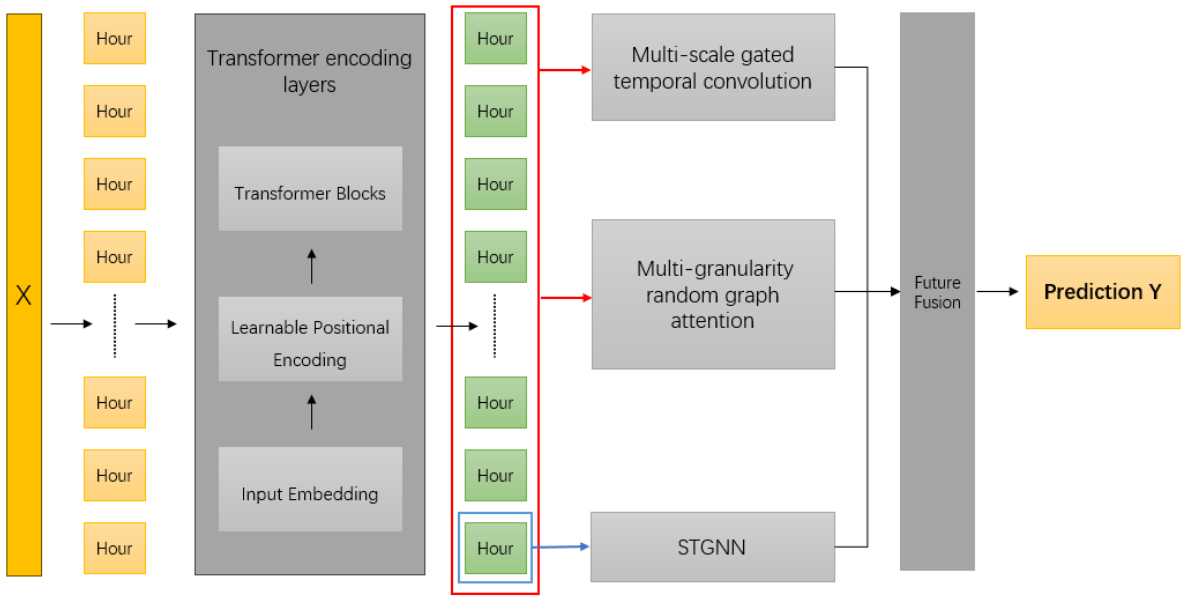

Fig. 2. The overall framework of the model

3.2 Multi-scale Gated Temporal Convolutional Network

The sparseness of short-term historical data makes it difficult for existing models to predict complex future traffic flow changes. In contrast, the long-term trend features contained in long-term historical series can help models predict future traffic flow changes more accurately. Therefore, we design a multi-scale gated time convolution module that can identify and extract long-term trend features of traffic flow from long-term historical series.

The multi-scale gated temporal convolutional network is constructed with two dilated initial layers (DILs). One DIL employs the tanh activation function as a filter, while the other DIL utilizes the sigmoid activation function as a gating mechanism. This arrangement effectively controls the flow of information from the preceding layer to the subsequent module. Furthermore, each DIL utilizes a collection of typical 1D causal dilated convolutions to capture temporal characteristics at different scales. Hence, this module can handle lengthy time series

and extract temporal features at several scales. The architecture of the multi-scale gated temporal convolutional network is illustrated in Fig. 3.

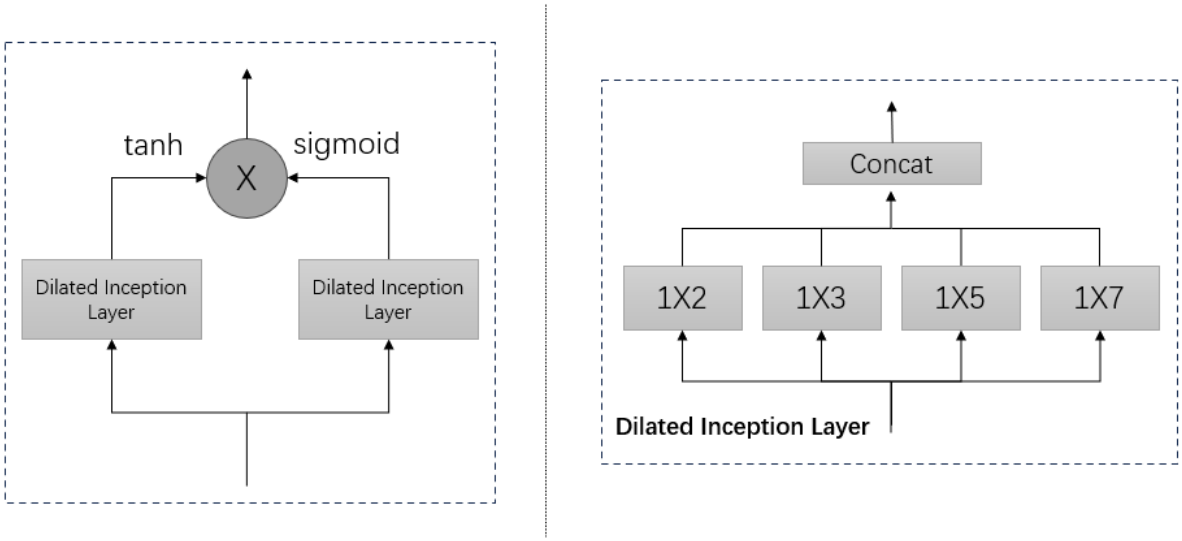

Fig. 3. The composition structure of the multi-scale gated temporal convolutional network

Given an input $H \in R^{C\times N\times T}$, C is the number of channels, N is the number of sequences, and T is the length of the sequence. Taking the first DIL layer as an example, we first extract multi-scale time features in long-term historical sequences by a set of 1D causally dilated convolutional layers (without padding). Among them, the convolution kernels [k_1, k_2, k_3, k_4] of the causally dilated convolutional layers are gradually increased. At the same time, we map the channels to C/4 (divide the channels equally according to the number of convolutional layers):

$$H_1 \in R^{(C/4)\times N\times (T-k_1+1)} = \mathrm{Conv}_{1\times k_1}(H)\,. \tag{5}$$

$$H_2 \in R^{(C/4)\times N\times (T-k_2+1)} = \mathrm{Conv}_{1\times k_2}(H)\,. \tag{6}$$

$$H_3 \in R^{(C/4)\times N\times (T-k_3+1)} = \mathrm{Conv}_{1\times k_3}(H)\,. \tag{7}$$

$$H_4 \in R^{(C/4)\times N\times (T-k_4+1)} = \mathrm{Conv}_{1\times k_4}(H)\,. \tag{8}$$

Then, taking the feature with the least time dimension as the standard (H_4 has the least time dimension because its convolution kernel is the largest), we intercept H_1, H_2, H_3 in the time dimension to keep their time dimensions consistent:

$$H_1 \in R^{(C/4)\times N\times (T-k_4+1)} = H_1[\ldots, -(T-k_4+1):]\,. \tag{9}$$

$$H_2 \in R^{(C/4)\times N\times (T-k_4+1)} = H_2[\ldots, -(T-k_4+1):]\,. \tag{10}$$

$$H_3 \in R^{(C/4)\times N\times (T-k_4+1)} = H_3[\ldots, -(T-k_4+1):]\,. \tag{11}$$

Next, the four multi-scale feature values are concatenated on the channel to restore the same number of channels as the input:

$$H_L \in R^{C\times N\times(T-k_4+1)} = \mathrm{Concat}\left(H_1, H_2, H_3, H_4\right) . \tag{12}$$

Ultimately, two fully connected layers are employed to regain the original temporal dimension of the input:

$$H_L = \mathrm{MLP}\left(H_L\right), \tag{13}$$

where $H_L \in R^{C\times N\times T}$ is the result produced by the initial DIL layer.

By employing the identical approach as described earlier, we obtain the resultant output of the second DIL layer $H_R \in R^{C\times N\times T}$. Next, we create a gating unit using the sigmoid function and a filter using the tanh activation function. The gating unit regulates the amount of information that the filter transfers to the subsequent modules:

$$H_{Long} \in R^{C\times N\times T} = \tanh\left(H_L\right) \odot \mathrm{sigmoid}\left(H_R\right) . \tag{14}$$

3.3 Multi-granularity Random Graph Attention Network

Periodicity is an important feature of traffic flow data. Traffic flow often exhibits weekly and daily periodicity, and different times within the same period may show very similar trends. To extract the periodic features of traffic flow data, we design a multi-granularity random graph attention module. This module can extract periodic features from the hour-level subsequences representations of the previous two weeks and previous days.

The multi-granular random graph attention module first generates two adjacency matrices, namely the hour-level adjacency matrix and the day-level adjacency matrix. The hour-level adjacency matrix represents the pairwise relationship between two hour-level subsequences R_h, and each node is connected by checking the periodic pattern. The same hour-level subsequences R_h on different days will be linked together. In this process, the corresponding position weight is set to 1, while the position weights of other nodes are set to 0. The day-level adjacency matrix represents the link between two day-level subsequences R_d. It is convolved by the time-level subsequences R_h within a day. Each node in the day-level adjacency matrix considers the periodic pattern. The same day-level subsequences R_h in different weeks will be linked together. This means that the corresponding position weight is set to 1, while the position weights of other nodes are set to 0.

Next, the multi-granular random graph attention module constructs a trainable attention weight matrix A_r. This weight matrix is not formed interactively but is randomly generated. Through this weight matrix, the correlation between periods of different days and days of different weeks can be learned. Thus, the periodic features in the time-level adjacency matrix and the day-level adjacency matrix can be extracted.

Finally, the attention matrix A_r is added to the hour-level adjacency matrix R_h and the day-level adjacency matrix R_d to generate the final adjacency matrices A_h and A_d. A_h and A_d are expressed as:

$$A_h = A_r + R_h . \tag{15}$$

$$A_d = A_r + R_d . \tag{16}$$

To capture more periodic features and stabilize the learning process, we further extend the multi-granularity random graph attention into a multi-head mechanism. We set the number of multi-head attention to K, and the final output is as follows:

$$H_{Period} = MLP\left(\sigma\left(A_h W H_h\right) \| \sigma\left(A_d W H_d\right)\right), \tag{17}$$

$$H_{Periodicity} = \mathrm{Concat}\left(H^1_{Period}, \ldots, H^K_{Period}\right), \tag{18}$$

Where ‖ represents concatenation and $H_{Periodicity} \in R^{T\times N\times C}$ represents the output of multi-granularity random graph attention module.

3.4 STGNN

Since the hour-level subsequences obtained from the long-term historical time series are not detailed enough, the model does not fully extract the short-term trend features of traffic flow. Furthermore, since there is a significant temporal connection between future and past short-term traffic flows, it is imperative to estimate the short-term trend independently.

STGNNs have been extensively proven to effectively extract short-term trend characteristics from short-term sequences, as evidenced by multiple studies [8, 9, 28]. Typically, STGNN models accomplish comprehensive feature extraction through a two-step process. Initially, people acquire spatial features and temporal features by means of a spatial learning network and a temporal learning network, correspondingly. Next, they combine the two features using a specific spatiotemporal fusion structure. Considering the acknowledged strengths of STGNNs, we choose to utilize the already established STGNN, Graph WaveNet [12], to extract the short-term trend features of traffic flow. H_{short}:

$$H_{Short} = \mathrm{STGNN}\left(A, H_{last}\right), \tag{19}$$

where H_{last} represents the last subsequence in the long historical sequence, that is the nearest subsequence at the hour level to the current moment. A is the adjacency matrix, whereas STGNN() denotes the implemented Graph WaveNet model.

3.5 Feature Fusion

By modeling long-term historical data, we obtain long-term trend features, periodic features, and short-term trend features. Finally, by combining the three features, we can get the final prediction result of the model:

$$\hat{\mathcal{Y}} = \mathrm{MLP}\left(H_{Long} || H_{Periodicity} || H_{Short}\right), \tag{20}$$

where ‖ represents concatenation. The objective of the traffic flow prediction task is to minimize the discrepancy between the model's output and the actual value. To do this, we use the average error as the loss function, which is defined as:

$$\mathcal{L}_{\mathrm{mae}} = \mathcal{L}(\hat{\mathcal{Y}}, \mathcal{Y}) = \frac{1}{T_f NC} \sum_{j=1}^{T_f} \sum_{i=1}^{N} \sum_{k=1}^{C} \left| \hat{\mathcal{Y}}_{ik}^{i} - \mathcal{Y}_{jk}^{i} \right|. \tag{21}$$

4 Experiments

This section presents experiments on two real public transportation datasets to illustrate the effectiveness of our method in predicting traffic flow. In Section 4.1, we give a comprehensive explanation of the experimental datasets. Next, we introduce the evaluation metrics we use in Section 4.2. In addition, in Section 4.3, we describe the basic details of the baseline model. In Section 4.4, we evaluate the effectiveness of our model by comparing it with the baseline model. Finally, in Section 4.5, we conduct ablation experiments on the modules and analyze their impact on the model performance to verify the effectiveness of our proposed modules.

4.1 Datasets

To thoroughly assess the effectiveness of our method, we have chosen two real public transportation flow datasets, namely PEMS04 and PEMS08. These datasets consist of numerous time intervals and hundreds of sensors. Table 1 provides a comprehensive description of the two datasets.

- PEMS04. The collection is obtained from 307 sensors located on 29 freeways in California. PEMS-04 includes data from 307 sensors that were gathered over a span of two months, specifically from January 1, 2018 to February 28, 2018. Information on traffic movement is recorded at 5-minute intervals, resulting in a total of 16,992 data points.
- PEMS08. The collection comprises traffic data obtained from 170 sensors located on 8 freeways in California. PEMS08 dataset comprises data from 170 sensors gathered during a two-month timeframe spanning from July 1, 2016 to August 31, 2016. Additionally, traffic flow data is captured at 5-minute intervals, resulting in a total of 17,833 time slices.

During the experiment, we normalized the two sets of highway traffic data, PEMS04 and PEMS08, to scale the data values to a uniform magnitude, which is helpful for model training and generalization. After that, we allocated about 60% of the data for training, 20% of the data for testing, and the remaining 20% for validation.

Table 1. Detailed description of the data set

Dataset	Nodes	Time range	Time interval/min	Time slices
PEMS04	307	2018/01/01-2018/02/28	5	16992
PEMS08	170	2016/07/01-2016/08/31	5	17833

4.2 Evaluation Metrics

Three standard metrics are employed to assess the performance of all baselines: mean absolute error (MAE), root mean square error (RMSE), and mean absolute percentage error (MAPE). MAE measures prediction accuracy, RMSE is particularly responsive to outliers, and MAPE quantifies the relative deviation between predicted and actual values. The equations for these metrics are as follows:

$$MAE = \frac{1}{n}\sum_{t=1}^{n}\left|Y_t - \hat{Y}_t\right| . \tag{22}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n}\left(Y_t - \hat{Y}_t\right)^2} . \tag{23}$$

$$MAPE = \frac{1}{n}\sum_{t=1}^{n}\frac{\left|Y_t - \hat{Y}_t\right|}{Y_t} \times 100\% . \tag{24}$$

4.3 Baseline Models

We selected various baselines with official public codes, including traditional methods, typical deep learning methods, and recent state-of-the-art works.

- HA [17]: The Historical Average (HA) model is a straightforward and efficient time series forecasting technique that predicts future values by computing the mean of all previous observations.
- VAR [5]: The VAR model believes that each variable is not only influenced by its previous values but also by the past values of all other variables. It is well-suited for capturing intricate interconnections in time series data.
- LSTM [38]: LSTM solves the gradient vanishing and gradient exploding problems by introducing memory units and gate mechanisms, effectively capturing long-term dependencies.

- DCRNN [8]: DCRNN introduces diffusion convolution, the idea of which is to extend the convolution operation to the graph and consider the diffusion process between nodes to capture the spatial dependencies in the graph structure.
- Graph WaveNet [12]: The Graph WaveNet model provides a powerful approach to modeling spatiotemporal data by combining causal convolutions with an adaptive graph learning module.
- ASTGCN [33]: ASTGCN integrates the Graph Convolutional Network (GCN) and the attention mechanism to accurately capture intricate relationships in spatiotemporal data.
- STSGCN [39]: STSGCN efficiently captures intricate local spatiotemporal correlations via a well-crafted spatiotemporal synchronization modeling approach.
- GMAN [32]: GMAN utilizes an encoder-decoder design, with numerous spatiotemporal attention blocks in both the encoder and decoder. This allows for the modeling of the influence of spatiotemporal elements on traffic conditions.
- MTGNN [13]: MTGNN presents a comprehensive graph neural network framework specifically tailored for multivariate time series data.
- DGCRN [40]: The DGCRN model provides an effective method for modeling spatiotemporal data by combining dynamic graph convolutional networks and recurrent neural networks.
- STEP [10]: STEP proposes an innovative method that combines pre-training and ST-GNN, enabling the model to learn richer spatiotemporal features.
- PatchTST [16]: PatchTST performs specific optimization on time series data, dividing the time series data into several small time slices (patches) and then processing them through the Transformer model.

4.4 Experimental Results

To evaluate the prediction accuracy of the proposed model, we conduct experiments on two real traffic datasets and compare the model with thirteen other baseline models. Table 2 and Table 3 show the prediction results of all models on the two datasets. As shown in Table 2 and Table 3, our method outperforms the baseline methods in all indicators of both datasets, indicating the effectiveness of our method.

Statistical Analysis Methods HA and VAR perform the worst because they have strong assumptions about the data, such as stationarity or linearity. LSTM is a classic recurrent neural network that performs poorly because it only considers temporal features. Thanks to the strong ability of the attention mechanism to capture long-term dependencies, GMAN performs well in long-term prediction. DCRNN and Graph WaveNet are two typical spatiotemporal graph neural networks. Even compared with many newer models, such as ASTGCN and STSGCN, their performance is still very good. Due to the introduction of long-term historical time series and modeling of long-term trends, PatchTST and STEP both have good prediction performance. However, their prediction performance is weaker than our model's due to the lack of modeling of the periodic features of traffic flow.

Table 2. Experimental results of algorithm prediction performance on PEMS04

Methods	Horizon 3			Horizon 6			Horizon 12		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
HA	28.92	42.69	20.31%	33.73	49.37	24.01%	46.97	67.43	35.11%
VAR	21.94	34.30	16.42%	23.72	36.58	18.02%	26.76	40.28	20.94%
LSTM	21.42	33.37	15.32%	25.83	39.10	20.35%	36.41	50.73	29.92%
DCRNN	20.34	31.94	13.65%	23.21	36.15	15.70%	29.24	44.81	20.09%
STGCN	19.35	30.76	12.81%	21.85	34.43	14.13%	26.97	41.11	16.84%
Graph WaveNet	18.15	29.24	12.27%	19.12	30.62	13.28%	20.69	33.02	14.11%
ASTGCN	20.15	31.43	14.03%	22.09	34.34	15.47%	26.03	40.02	19.17%
STSGCN	19.41	30.69	12.82%	21.83	34.33	14.54%	26.27	40.11	14.71%
MTGNN	18.22	30.13	12.47%	19.27	32.21	13.09%	20.93	34.49	14.02%
GMAN	18.28	29.32	12.35%	18.75	30.77	12.96%	19.95	30.21	12.97%
DGCRN	18.27	28.97	12.36%	19.39	30.86	13.42%	21.09	33.59	14.94%
STEP	17.47	28.46	12.10%	18.24	29.81	12.49%	19.34	31.52	13.26%
PatchTST	17.52	28.67	11.96%	18.23	29.91	12.34%	19.41	31.71	13.09%
OURS	17.43	28.38	11.62%	18.22	29.53	12.24%	19.24	31.27	12.91%

Table 3. Experimental results of algorithm prediction performance on PEMS08

Methods	Horizon 3			Horizon 6			Horizon 12		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE
HA	23.52	34.96	14.72%	27.67	40.89	17.37%	39.28	56.74	25.17%
VAR	19.52	29.73	12.54%	22.25	33.30	14.23%	26.17	38.97	17.32%
LSTM	17.38	26.27	12.63%	21.22	31.97	17.32%	30.69	43.96	25.72%
DCRNN	15.64	25.48	10.04%	17.88	27.63	11.38%	22.51	34.21	14.17%
STGCN	15.30	25.03	9.88%	17.69	27.27	11.03%	25.46	33.71	13.34%
Graph WaveNet	14.02	22.76	8.95%	15.24	24.22	9.57%	16.67	26.77	10.86%
ASTGCN	16.48	25.09	11.03%	18.66	28.17	12.23%	22.83	33.68	15.24%
STSGCN	15.45	24.39	10.22%	16.93	26.53	10.84%	19.50	30.43	12.27%
MTGNN	14.24	22.43	9.02%	15.30	24.32	9.58%	16.85	26.93	10.57%
GMAN	13.80	22.88	9.41%	14.62	24.02	9.57%	15.72	25.96	10.56%
DGCRN	13.89	22.07	9.19%	14.92	23.99	9.85%	16.73	26.88	10.84%
STEP	13.24	21.37	8.71%	14.04	24.03	9.46%	15.01	25.89	9.90%
PatchTST	13.31	21.65	8.65%	14.09	23.93	9.41%	15.53	26.36	10.05%
OURS	12.95	21.09	8.35%	13.68	22.72	9.37%	14.55	24.25	9.88%

To assess the advantages of the method in long-term forecasting, we performed more sophisticated tests. We calculated the Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Mean Absolute Percentage Error (MAPE) for the model and several benchmark models over the subsequent 12 time intervals. We then plotted the changes in these metrics. Fig. 4 and Fig. 5 show the results. From the figure, it can be observed that the OURS method (the model proposed in the paper) outperforms the other baseline models for all time steps (1 to 12) and all metrics (MAE, RMSE, MAPE). Its advantages are mainly reflected in the following:

- Stability for long-term prediction: the absolute errors at different time steps are smaller than those of other methods, and the error growth rate is the lowest, reflecting its advantages for long-term prediction.
- Adaptation to different datasets: consistent performance is maintained on both datasets (PEMS04 and PEMS08).

The model's excellent performance is mainly attributed to two modules: multi-scale gated temporal convolution and multi-granularity random graph attention. The multi-scale gated temporal convolution module can capture the long-term trend features in the long-term time series. The multi-granularity random graph attention module can effectively extract periodic features in long-term historical time series. These modules enable the model to better utilize the rich information in the long-term historical time series and thus stay ahead in long-term forecasting.

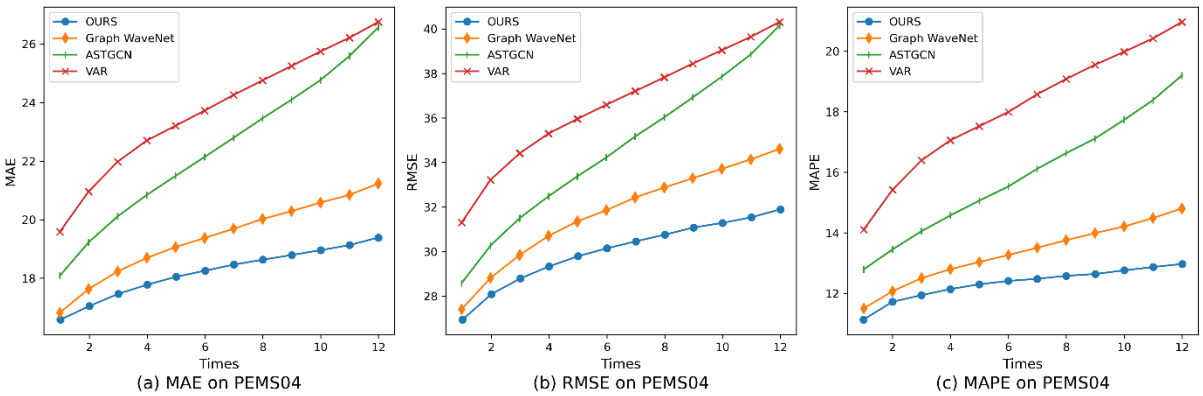

Fig. 4. Performance changes of the model at different time steps on the PEMS04 dataset

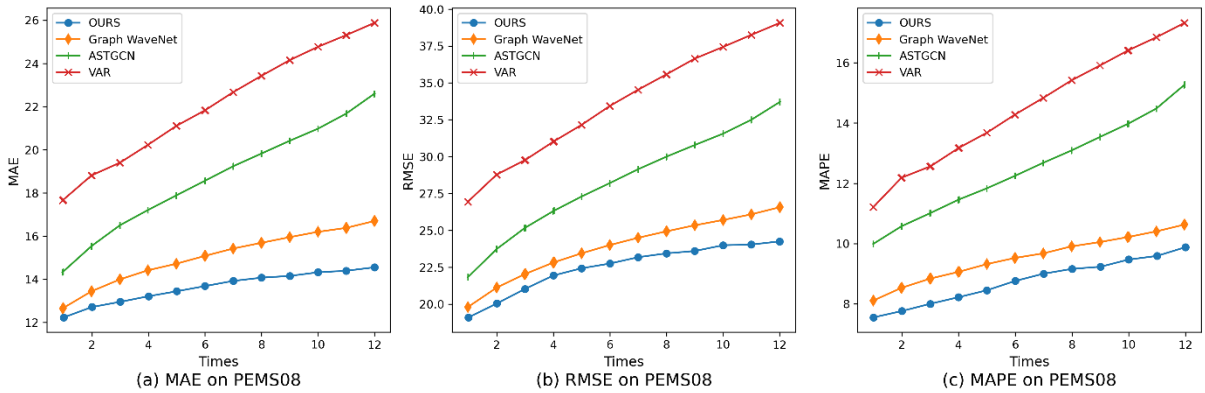

Fig. 5. Performance changes of the model at different time steps on the PEMS08 dataset

We display the forecast outcomes of the four nodes and juxtapose them with the actual data on the PEMS04 and PEMS08 datasets. As depicted in Fig. 6, the upper part of the figure is the prediction result of Sensor 108 and Sensor 178 on the PEMS04 dataset from January 2 to 4, 2018, both of which are weekends. Based on the results, it is evident that the model demonstrates a highly satisfactory level of prediction accuracy across the board. Still, due to the large random noise, the prediction of some local details may be inaccurate. The lower part of the figure is the prediction result of Sensor 148 and Sensor 44 on July 2 to 4, 2016, on the PEMS08 dataset, both of which are weekdays. Due to the weekdays, there is a lot of noise in the traffic signals, which makes the prediction task very challenging. Nevertheless, the prediction results indicate that our model generates sensible forecasts, demonstrating its resilience to traffic noise.

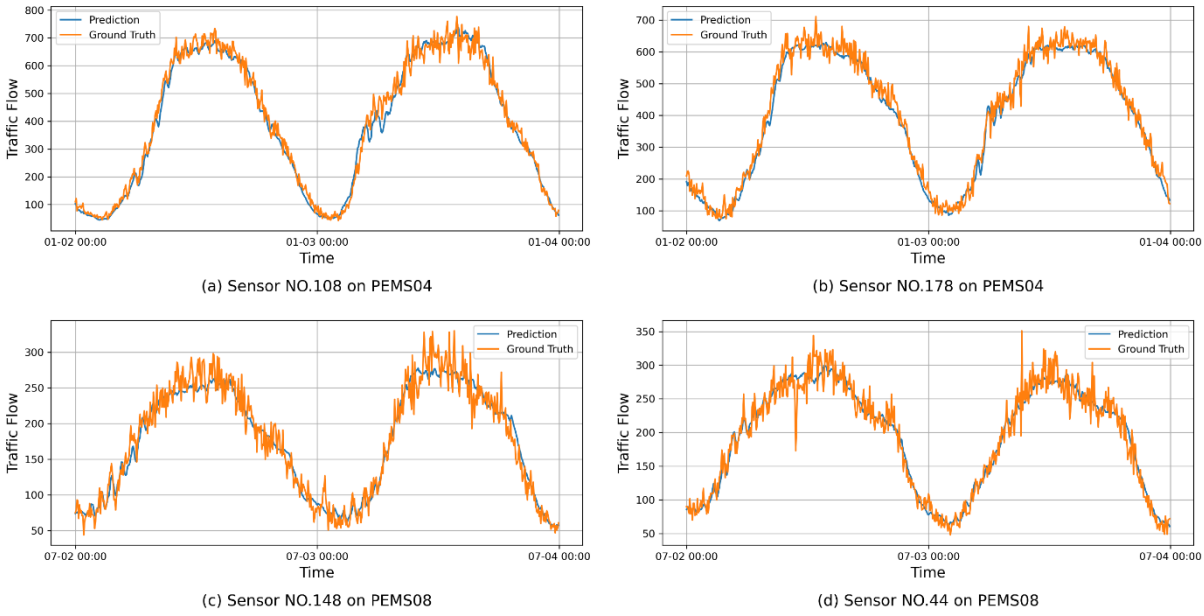

Fig. 6. Prediction results of model on different nodes within two days

4.5 Ablation Studies

To assess the efficacy of the proposed components outlined in this research, we performed ablation experiments and evaluated their performance on the PEMS04 dataset. Two variants were designed: ① w/o MSGTC: the model lacking the multi-scale gated temporal convolution module; ② w/o MGRGA: the model lacking the

multi-granularity random graph attention, and the precise information is displayed in Table 4. Fig. 7 displays the outcomes of the ablation experiment. The original model shows the best performance in all three evaluation indicators on the PEMS04 dataset. The figure clearly demonstrates that the model's performance is inferior to the original model in all three evaluation indicators when the multi-scale gated temporal convolution and the multi-granularity random graph attention are not included. This further confirms that explicitly modelling the long-term trend and periodic characteristics of traffic flow enhances the predictive performance of the model.

Table 4. Model comparison of ablation experiments

Model	Multi-scale gated temporal convolution	Multi-granularity random graph attention
w/o MSGTC	×	✓
w/o MGRGA	✓	×
Original	✓	✓

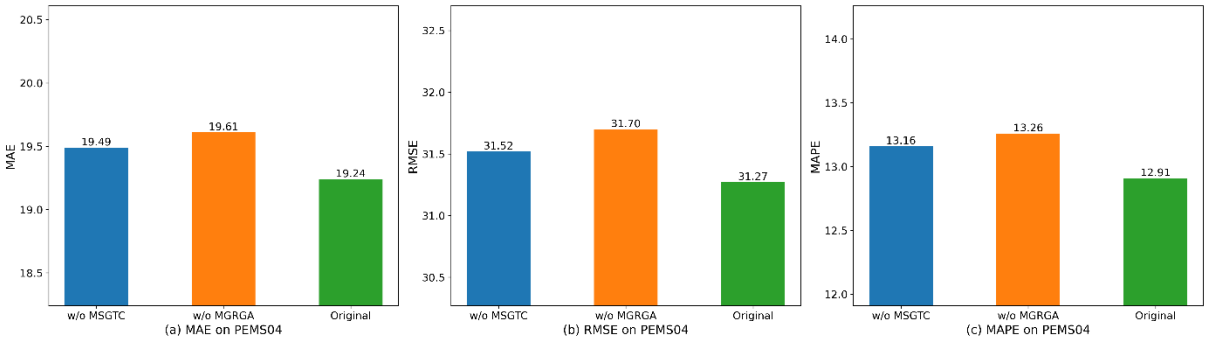

Fig. 7. Ablation experiment on PEMS04 dataset

In order to further analyze the superiority of the multi-scale gated temporal convolution module and multi-granularity random graph attention module designed in this paper compared with other method modules. We conducted two sets of ablation experiments on the PEMS08 dataset. In the first set of ablation experiments, we replaced the multi-scale gated temporal convolution with dilated causal convolution and multi-scale Dilated causal convolution, keeping the other modules of the model unchanged. In the second set of ablation experiments, we also replaced the Multi-granularity random graph attention with self-attention and random graph attention, respectively, keeping the other modules of the model unchanged. The results are shown in Table 5 and Table 6.

It can be seen from Table 5 that the models using dilated causal convolution and multi-scale dilated causal convolution will produce larger errors than the model using multi-scale gated temporal convolution because they cannot capture the long-term trend of traffic flow from the long-term historical series well.

Table 5. Performance comparison of different temporal convolution modules

Module	MAE	RMSE	MAPE
Dilated causal convolution	19.49	31.52	13.16%
Multi-scale Dilated causal convolution	19.37	31.41	13.09%
Multi-scale gated temporal convolution	19.24	31.27	12.91%

Table 6 shows that the prediction effect of multi-granularity random graph attention is better than self-attention and random graph attention. This is because multi-granularity random graph attention can effectively extract periodic features from long-term historical sequences.

The results in Table 5 and Table 6 prove the excellence of multi-scale gated temporal convolution and multi-granularity random graph attention proposed in this paper.

Table 6. Performance comparison of different attention modules

Module	MAE	RMSE	MAPE
Self-attention	19.61	31.70	13.26%
Random graph attention	19.33	31.48	13.03%
Multi-granularity random graph attention	19.24	31.27	12.91%

5 Conclusion

This research presents a novel framework for predicting traffic flow, namely A Long Term Transformer-based spatiotemporal graph attention network. The framework is designed to accurately predict traffic flow by leveraging long-term trends and periodic features in historical time series data. The model first uses the Transformer encoding layer to effectively learn compressed, context-rich hour-level subsequence representations from long historical time series. Next, the multi-scale gated temporal convolution module is employed to extract the specific features of traffic flow's long-term trend from the representation of hour-level subsequences. Subsequently, the multi-granularity random graph attention module is used to capture the periodic features from the hour-level subsequence data encompassing the previous two weeks and days. Ultimately, the long-term trend features, periodic features, and short-term trend features are combined to yield the ultimate forecast findings. The proposed model demonstrates a substantial enhancement in prediction accuracy when compared to earlier cutting-edge models, as evidenced by experimental findings from two real-world traffic datasets. Furthermore, comprehensive ablation research and visualization experiments emphasize the significance of each module used in the model for precise traffic forecasts. However, the model performs relatively poorly in terms of reasoning time and applicability, and cannot capture the dynamic spatiotemporal correlation of traffic flow well. Therefore, our future research goals will first study the cross-domain applicability of the model architecture, especially its applicability in weather forecasting and wind forecasting. Secondly, we will simplify the component modules of the model to reduce the reasoning time. Finally, we will explore the dynamic characteristics of traffic flow so that the model can capture the dynamic spatiotemporal correlation of traffic flow well.

6 Acknowledgement

This work is supported by the National Natural Science Foundation of China (62376059), and the project is funded by the Fujian Provincial Department of Finance (GY-Z23012).

References

[1] S. Rahmani, A. Baghbani, N. Bouguila, Z. Patterson, Graph neural networks for intelligent transportation systems: A survey, IEEE Transactions on Intelligent Transportation Systems 24(8)(2023) 8846–8885.
[2] Y. Wang, C. Jing, S. Xu, T. Guo, Attention based spatiotemporal graph attention networks for traffic flow forecasting, Information Sciences 607(2022) 869–883.
[3] C. Zheng, X. Fan, S. Pan, H. Jin, Z. Peng, Z. Wu, C. Wang, S. Y. Philip, Spatio-temporal joint graph convolutional networks for traffic forecasting, IEEE Transactions on Knowledge and Data Engineering 36(1)(2023) 372–385.
[4] G. Yu, C. Zhang, Switching arima model based forecasting for traffic flow, in: Proc. 2004 International Conference on Acoustics, Speech, and Signal Processing, 2004.
[5] S.R. Chandra, H. Al-Deek, Predictions of freeway traffic speeds and volumes using vector autoregressive models, Journal of Intelligent Transportation Systems 13(2)(2009) 53–72.
[6] J. Zhang, Y. Zheng, D. Qi, Deep spatio-temporal residual networks for citywide crowd flows prediction, in: Proc. 2017 AAAI conference on artificial intelligence, 2017.
[7] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, W.-C. Woo, Convolutional lstm network: A machine learning approach for precipitation nowcasting. <https://arxiv.org/abs/1506.04214>, 2015 (accessed 15.07.2024).
[8] Y. Li, R. Yu, C. Shahabi, Y. Liu, Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. <https://arxiv.org/abs/1707.01926>, 2017 (accessed 28.05.2024).
[9] B. Yu, H. Yin, Z. Zhu, Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. <https://arxiv.org/abs/1709.04875>, 2017 (accessed 20.07.2024).

[10] Z. Shao, Z. Zhang, F. Wang, Y. Xu, Pre-training enhanced spatial-temporal graph neural network for multivariate time series forecasting, in: Proc. 2022 International Conference on Knowledge Discovery and Data Mining, 2022.
[11] Z. Shao, Z. Zhang, W. Wei, F. Wang, Y. Xu, X. Cao, C.S. Jensen, Decoupled dynamic spatial-temporal graph neural network for traffic forecasting. <https://arxiv.org/abs/2206.09112>, 2022 (accessed 25.06.2024).
[12] Z. Wu, S. Pan, G. Long, J. Jiang, C. Zhang, Graph wavenet for deep spatial-temporal graph modeling. <https://arxiv.org/abs/1906.00121>, 2019 (accessed 10.07.2024).
[13] Z. Wu, S. Pan, G. Long, J. Jiang, X. Chang, C. Zhang, Connecting the dots: Multivariate time series forecasting with graph neural networks, in: Proc. 2020 International Conference on Knowledge Discovery and Data Mining, 2020.
[14] D. Cao, Y. Wang, J. Duan, C. Zhang, X. Zhu, C. Huang, Y. Tong, B. Xu, J. Bai, J. Tong, Spectral temporal graph neural network for multivariate time-series forecasting, Advances in neural information processing systems 33(2020) 17766–17778.
[15] C. Shang, J. Chen, J. Bi, Discrete graph structure learning for forecasting multiple time series. <https://arxiv.org/abs/2101.06861>, 2021 (accessed 06.06.2024).
[16] Y. Nie, N.H. Nguyen, P. Sinthong, J. Kalagnanam, A time series is worth 64 words: Long-term forecasting with transformers. <https://arxiv.org/abs/2211.14730>, 2022 (accessed 24.05.2024).
[17] B.L. Smith, M.J. Demetsky, Traffic flow forecasting: comparison of modeling approaches, Journal of transportation engineering 123(4)(1997) 261–266.
[18] J.W. Gao, Z.W. Leng, B. Zhang, X. Liu, G.Q. Cai, The application of adaptive kalman filter in traffic flow forecasting, Advanced Materials Research 680(2013) 495–500.
[19] W. Jiang, J. Luo, Graph neural network for traffic forecasting: A survey, Expert Systems with Applications 207(2022) 117921.
[20] X. Ma, Z. Dai, Z. He, J. Ma, Y. Wang, Y. Wang, Learning traffic as images: A deep convolutional neural network for large-scale transportation network speed prediction, Sensors 17(4)(2017) 818.
[21] Y. Liu, H. Zheng, X. Feng, Z. Chen, Short-term traffic flow prediction with conv-lstm, in: Proc. 2017 International Conference on Wireless Communications and Signal Processing (WCSP), 2017.
[22] T.N. Kipf, M. Welling, Variational graph auto-encoders. <https://arxiv.org/abs/1611.07308>, 2016 (accessed 15.06.2024).
[23] M. Zhang, Z. Cui, M. Neumann, Y. Chen, An end-to-end deep learning architecture for graph classification, in: Proc. 2018 AAAI conference on artificial intelligence, 2018.
[24] H. Li, J. Cao, J. Zhu, Y. Liu, Q. Zhu, G. Wu, Curvature graph neural network, Information Sciences 592(2022) 50–66.
[25] T.N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks. <https://arxiv.org/abs/1609.02907>, 2016 (accessed 30.06.2024).
[26] I. Sutskever, O. Vinyals, Q.V. Le, Sequence to sequence learning with neural networks. <https://arxiv.org/abs/1409.3215>, 2014 (accessed 20.06.2024).
[27] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need. <https://arxiv.org/abs/1706.03762>, 2017 (accessed 11.06.2024).
[28] Z. Pan, Y. Liang, W. Wang, Y. Yu, Y. Zheng, J. Zhang, Urban traffic prediction from spatio-temporal data using deep meta learning, in: Proc. 2019 International Conference on Knowledge Discovery & Data mining, 2019.
[29] L. Zhao, Y. Song, C. Zhang, Y. Liu, P. Wang, T. Lin, M. Deng, H. Li, T-GCN: A temporal graph convolutional network for traffic prediction, IEEE transactions on intelligent transportation systems 21(9)(2019) 3848–3858.
[30] L. Bai, L. Yao, C. Li, X. Wang, C. Wang, Adaptive graph convolutional recurrent network for traffic forecasting, Advances in neural information processing systems 33(2020) 17804–17815.
[31] K. Cho, B. Van, D. Bahdanau, Y. Bengio, On the properties of neural machine translation: Encoder-decoder approaches. <https://arxiv.org/abs/1409.1259>, 2014 (accessed 05.07.2024).
[32] C. Zheng, X. Fan, C. Wang, and J. Qi, Gman: A graph multi-attention network for traffic prediction, in: Proc. 2020 AAAI conference on artificial intelligence, 2020.
[33] S. Guo, Y. Lin, N. Feng, C. Song, H. Wan, Attention based spatial-temporal graph convolutional networks for traffic flow forecasting, in: Proc. 2019 AAAI conference on artificial intelligence, 2019.
[34] R. Jiang, Z. Wang, J. Yong, P. Jeph, Q. Chen, Y. Kobayashi, X. Song, T. Suzumura, S. Fukushima, Megacrn: Meta-graph convolutional recurrent network for spatio-temporal modeling. <https://arxiv.org/abs/2212.05989>, 2022 (accessed 02.06.2024).
[35] L. Chen, D. Chen, Z. Shang, B. Wu, C. Zheng, B. Wen, W. Zhang, Multi-scale adaptive graph neural network for multivariate time series forecasting, IEEE Transactions on Knowledge and Data Engineering 35(10)(2023) 10748–10761.
[36] Y. Zhang, J. Yan, Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting, in: Proc. 2023 International Conference on Learning Representations, 2023.
[37] Z. Chen, M. Ma, T. Li, H. Wang, Long sequence time-series forecasting with deep learning: A survey, Information Fusion 97(2023) 101819.
[38] S. Hochreiter, Long Short-term Memory, Neural Computation MIT-Press 9(8)(1997) 1735–1780.
[39] C. Song, Y. Lin, S. Guo, Spatial-temporal synchronous graph convolutional networks: A new framework for spatial-temporal network data forecasting, in: Proc. 2022 AAAI conference on artificial intelligence, 2020.
[40] F. Li, J. Feng, H. Yan, Dynamic graph convolutional recurrent network for traffic prediction: Benchmark and solution, ACM Transactions on Knowledge Discovery from Data 17(1)(2023) 1–21.