STFFormer-GCN: Spatial–Temporal Fusion Transformer Based Graph Convolutional Network for Traffic Flow Prediction

Guoqing Teng
Chongqing University of Science and Technology
Han Wu
Chongqing University of Science and Technology
Yixing Wang
Chongqing University of Science and Technology
Ao He
Chongqing University of Science and Technology
Yangsheng Long
Chongqing University of Science and Technology
Meng Zhao
zhaomeng@cqust.edu.cn

Chongqing University of Science and Technology

Research Article

Keywords: Traffic prediction, graph convolutional network, transformer, spatial–temporal fusion.

Posted Date: October 17th, 2024

DOI: https://doi.org/10.21203/rs.3.rs-5259013/v1

Additional Declarations: No competing interests reported.

STFFormer-GCN: Spatial–Temporal Fusion Transformer Based Graph Convolutional Network for Traffic Flow Prediction

Guoqing Teng[1], Han Wu[1], Yixing Wang[1], Ao He[1], Yangsheng Long[1], Meng Zhao[1*]

[1]Chongqing University of Science and Technology, 20 East University Road, Chongqing, 401331, Shapingba District, China.

*Corresponding author(s). E-mail(s): zhaomeng@cqust.edu.cn;

Abstract

Traffic flow prediction plays a crucial role in improving traffic management efficiency and optimizing traffic resource allocation. However, existing traffic flow prediction models are often difficult to accurately capture complex spatial–temporal dependencies. To this end, this paper proposes a spatial–temporal fusion Transformer based graph convolutional network (STFFormer-GCN) for traffic flow prediction, aiming to address the issues of long- and short-term dynamic spatial–temporal relationships, traffic propagation delays, and spatial–temporal feature fusion. First, we propose a multi-scale time convolution module employing recursion and sliding windows to model long- and short-term temporal dependencies. Second, a new dynamic relational convolution matrix is proposed for learning dynamic spatial dependencies that change over time and road structure. Then, we propose a time-delay cycle module that combines the current traffic flow and spatially propagated time delay information to model the propagation process of traffic flow in space. Finally, we employ Transformer's encoder to fuse dynamic spatial–temporal features with time delayed spatial–temporal features. We conducted extensive experiments on four real datasets and the results show that STFFormer-GCN achieves state-of-the-art performance,especially the MAPE value of the PeMS08 dataset experiment is improved by 4.05%. In addition, we conducted an ablation study and a single-step prediction performance study to evaluate the contribution of each component to the prediction accuracy of the model.

Keywords: Traffic prediction, graph convolutional network, transformer, spatial–temporal fusion.

1 Introduction

In recent years, the issue of traffic congestion has become increasingly serious with a skyrocketing urban population and the development of transportation infrastructure. Intelligent Transportation Systems (ITS) combine the Internet of Things (IoT) and intelligent algorithms to more efficiently plan transportation resources, relieve traffic congestion, and improve the overall efficiency of traffic flow through the collection and processing of data from multiple sources [1]. A variety of data collection and processing techniques provide us with an extensive traffic dataset, encompassing not only traditional traffic metrics such as volume and speed, but also modern metrics related to environmental conditions, including weather, accidents, and other real-time incidents. The key challenge is how to effectively integrate and utilize the temporal characteristics of this data with the complex spatial topology of urban networks. [2]. Early research relied extensively on traditional statistical and machine learning models, including vector autoregression (VAR) [3], the autoregressive integrated moving average (ARIMA) model [4], and support vector regression (SVR) [5]. However, these methods mainly focus on dependencies in the temporal dimension while ignoring the spatial correlations between different nodes. When dealing with complex data containing spatial correlations, these methodologies often leading to poor accuracy and reliability performance, and cannot fully capture the spatial characteristics. With the rapid development of deep learning technology, its application in the field of traffic flow prediction has become a hot trend. Convolutional Neural Networks (CNNs) are mainly used to process grid-based traffic flow data [6]; Graph Convolutional Networks (GCNs) capture spatial correlations by aggregating information from neighboring nodes and perform convolution operations using graph structures to extract node features and reveal node relationships [7]; Recurrent Neural Networks (RNNs) are used to capture dynamics and temporal dependencies in time series [8]; and Temporal Convolutional Networks (TCNs) extract temporal dependencies by stacking convolutional layers [9]. However, each network has its limitations. For instance, CNNs are more suitable for processing structured data but with poor performance when dealing with sequential data; RNNs may encounter the issue of exploding or vanishing gradients and are computationally inefficient when handling long-term sequences; TCNs are optimized to copy with sequential data by expanding the receptive field through causal and dilation convolutions, but their performance is deeply rely on proper hyperparameter tuning and may be less efficient than RNNs in dealing with very complex time-dependent patterns; the performance of GCNs is significantly influenced by the quality and structure of the input graph. Traditional GCNs often assume a static graph structure, which can be limiting when dealing with dynamically changing graph structures.Therefore, in order to comprehensively extract spatial–temporal features and obtain more accurate traffic flow predictions, researchers often combine the advantages of multiple network models. Specifically, researchers utilize Graph Convolutional Networks (GCNs) to extract spatial features of traffic data within the graph domain. Meanwhile, the temporal features of traffic data are captured using Recurrent Neural Networks (RNNs) and their variants, such as Gated Recurrent Units (GRUs), or Temporal Convolutional Networks (TCNs). However, four major challenges may be faced.

Firstly, as shown by node A and node D in Fig. 1b, how to deal with similar traffic behavior and spatial dependencies occurring in two distant locations under the influence of various aspects such as the city can planning and traffic management policies. Meanwhile, the spatial dependencies of transport networks can change dynamically. For example, as shown in Fig. 1d, the correlation between node C and node D is weaker during morning rush hour and stronger during other hours. Nevertheless, most existing methods model spatial dependencies statically. Widely used GCNs are based on static adjacency matrices, which are calculated by predefined distance functions or similarity measures and remain unchanged during the model training process, so they cannot dynamically adapt to changes in the input data [10]. Traffic flow is affected by time, weather, traffic accidents, road construction and other factors. Node relationships are constantly changing, making it difficult for static matrix models to adequately capture these dynamic changes.

Secondly, the traffic flow has significant periodicity, as shown in Fig. 1c. Although RNN-based methods can handle sequential data, they often suffer from the challenge of gradient disappearance or explosion in long-term dependency problems. These challenges arise because the gradient gradually decreases or increases during backpropagation, making learning difficult. In addition, RNNs tend to focus on recent data and ignore important information from previous periods, potentially limiting the model's accuracy and effectiveness in predicting long-term sequences. One of the main disadvantages of CNN-based methods is the fixed receptive field size. The size of a convolutional layer's receptive field is determined by the size of the convolution kernel and the number of layers, meaning that CNNs may not be able to capture longer-term dependencies when it comes to long-term temporal relationships.

Thirdly, the spread of spatial information between locations in a transportation system can be affected by time delays. For example, if a traffic accident occurs in one location, it will take several minutes (with a delay) to affect traffic conditions in neighboring locations [11], such as: B. the neighboring nodes A and B in Fig. 1e. However, GCNs typically assume that the spatial structure of the graph is static, and this static assumption may result in a model that does not accurately reflect the actual properties of spatial information propagation, especially in the short period after an accident.

Finally, a key challenge is to fully exploit the complementary advantages of these features to achieve a balance between different spatial–temporal features. Simple feature fusion methods such as direct splicing or summing can combine different spatial–temporal features together. However, these methods are often only shallow fusion, which cannot deeply explore and exploit the unique advantages of the two different feature extraction methods. It is difficult to achieve deep feature interaction and fusion. To address the above challenges, this paper proposes a novel STFFormer-GCN model that combines Graph Convolutional Network (GCN) and Transformer. The model is able to simultaneously model the long- and short-term temporal characteristics of different nodes and effectively capture the dynamic spatial dependencies between nodes. By combining delay information, the model is able to perform more accurate long-term traffic flow prediction. Meanwhile, we improve the accuracy of prediction by using an

encoder with an attention mechanism to fuse spatial–temporal features. The following are the main contributions of this paper:

1.We propose an STFFormer-GCN model based on a spatial–temporal self-attention mechanism, which utilizes the Transformer encoder structure to fuse dynamic spatial–temporal features and time-delayed spatial–temporal features, thus solving the problems of spatial–temporal feature fusion to effectively solve long- and short-term dynamic spatial–temporal relationships and delays in traffic propagation.

2.We propose a novel multi-scale time convolution module (MSTCM) that uses two different time convolution methods combining recursion and sliding windows to model long-term and short-term temporal dependencies.

3.Considering that the spatial dependencies between nodes may vary with time and road structure, we propose a new Dynamic Relational Convolution Matrix (DRCM) to capture the dynamic spatial dependencies at different points in time.

4.We propose a new Delay Cycle Module (DCM) to more accurately simulate the propagation of traffic in space by combining the current traffic flow and spatially propagated time delay information and weighting the outputs of different time steps.

5.We conducted experiments on four real traffic datasets (PeMS03, PeMS04, PeMS07 and PeMS08) to validate the effectiveness of the STFFormer-GCN model under different traffic patterns. Compared with the current state-of-the-art methods, the experimental results show that our model has significant improvements in prediction accuracy and efficiency. In addition, the model also shows significant advantages in one-hour single-step long-term prediction evaluation.

2 Related works

2.1 Traffic flow prediction

In previous studies, researchers used statistically based methods to predict traffic flow such as Historical Averaging (HA) [12] and AutoRegressive Integrated Moving Average models (ARIMA) [4]. These models are mainly based on linear dependencies and are therefore poorly suited to traffic scenarios where traffic flow exhibits nonlinear and dynamically changing characteristics. The researchers then proposed machine learning-based methods such as Support Vector Regression (SVR) [5], Support Vector Machines (SVM) [13] and Relative Density Based Knowledge Extraction (RDKE) [14] and K-Nearest Neighbor (K-NN) [15]. These methods perform better when modeling nonlinear data, but both the training and prediction phases can require significant computational resources, especially when dealing with large datasets and high-dimensional data that are computationally intensive. In recent years, deep learning-based methods have been widely used in the field of traffic flow prediction. In order to fully exploit the spatial features of traffic networks, some researchers have used CNNs to capture the adjacencies between traffic networks and RNNs along the time axis for prediction. Lv et al [16] proposed the LC-RNN framework for traffic flow prediction in large scale traffic networks, which combines CNNs and RNNs to model the spatial–temporal features of traffic data. However, CNN-based methods can only deal with regular network structures such as images. Such methods cannot effectively represent the spatial relationships of road networks with irregular traffic. Due to the

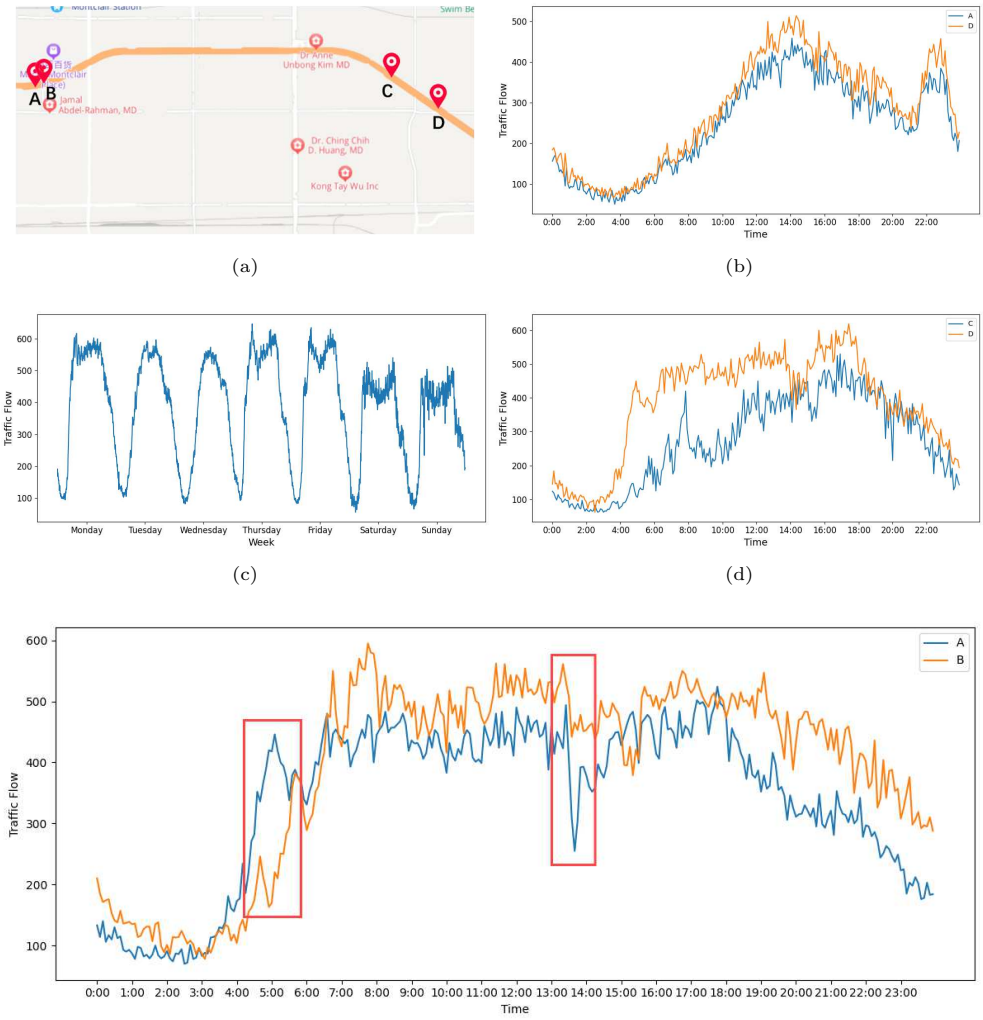

Fig. 1: Traffic condition diversity display. (a) Map of traffic network node distribution. (b) Long-range spatial dependencies. (c) Periodicity of data. (d) Dynamic spatial dependencies. (e) Propagation delay of neighborhood.

irregular shape of traffic road networks, they cannot be represented as a grid, which limits the application of CNN in traffic flow prediction.

2.2 Traffic prediction with graph-based structure

In recent years, deep graph convolutional networks have been widely used to predict traffic flow [17]. Researchers began using GCN to extract spatial features of non-Euclidean traffic networks. For example, Zhao et al. combined GCN [18] and Gated Recurrent Unit (GRU) to build a Temporal Graph Convolutional Network (T-GCN) model for traffic prediction. STGCN [19] used one-dimensional convolution instead of RNN to simulate long-term temporal relationships by stacking multiple layers, and then used GCN to extract spatial features. Guo et al. [20] proposed identifying spatial associations through multiple views with three different graph adjacency matrices. However, this approach treats spatial dependencies in spatial–temporal data as static and invariant and cannot model dynamic spatial relationships. Bai et al. [21] addressed the issue of adaptive adjacency matrix and proposed a method based on the automatic inference of hidden spatial dependencies from the data, but due to the fixed parameters, the method could only model the static structure of the traffic network after training.

With the successful application and continuous development of Transformer in natural language processing and computer vision [22], researchers have begun to integrate Transformer into traffic flow prediction. For example, ProSTformer [23] reduces computational complexity by decomposing and processing spatial–temporal dependencies using tensor reordering techniques and patch merging. TransGAT [24] uses a dynamic attention mechanism to capture dynamic spatial dependencies by assigning different weights based on real-time traffic data. STGAFormer [25] and PDFormer [11] combine Transformer encoders with GCN to achieve better results. STGAFormer and PDFormer use the Transformer to learn complex spatial–temporal correlations hidden in the traffic network and capture dynamic spatial-temporal dependencies. STGAFormer utilizes a distance-based spatial self-attention module to extract similar features when the distance exceeds a certain threshold, while focusing on key features when the distance falls below it. PDFormer employs two distinct mask matrices to emphasize spatial dependencies for both short and far views. Overall, Transformers excel at capturing intricate spatial-temporal relationships within traffic networks, thereby enhancing the accuracy of traffic flow predictions. However, the aforementioned approaches largely overlook the integration of different spatial-temporal features. In addition, most of them have difficulty performing well in long-term predictions.

3 Problem definition

The central goal of traffic flow prediction is to use historical traffic data collected from sensors at each node of a road network to predict future traffic flows at those locations. Therefore, specific topological relationships exist in the traffic data. Specifically, we model the traffic network as a graph $G = (V, E, A)$, where V is the set of nodes representing the sources of traffic sequences and E is the set of edges between nodes representing the physical or functional connectivity of the traffic flow sequences, and $A \in R^{N \times N}$ is the adjacency matrix that describes node-to-node proximity, e.g., based on traffic network distance or traffic sequence similarity. This graphical representation

not only represents the topology of the traffic network but also captures the spatial dependencies between traffic flow.

Our proposed prediction model uses a function f_θ to predict appearance of the traffic flow in a future τ time step by using historical traffic flow data from the past T time step and structural information of the graph G. This prediction issue can be formulated as follows:

$$\{X_{:,t+1}, X_{:,t+2}, \ldots, X_{:,t+\tau}\} = f_\theta(X_{:,t}, X_{:,t-1}, \ldots, X_{:,t-T+1}; G) \tag{1}$$

where, $X_{:,t} \in R_{N\times 1}$ denotes the vector of traffic flows for all nodes at time-step t, and θ represents all the learnable parameters in the model, highlighting the fact that the model not only relies on the historical information of the time-series data, but also makes use of the spatial information of the traffic network in making future traffic flow predictions.

4 Methodology

The overall framework of the STFFormer-GCN model proposed in this paper is shown in Fig. 2. The model contains the following five modules: multi-scale time convolution module, dynamic spatial convolution module, time-delay cycle module, spatial–temporal fusion module, and output layer.

First, we propose a multi-scale time convolution module for learning long-term and short-term temporal dependencies. Then, a dynamic spatial convolution module is employed to capture spatial dependencies directly from traffic flow data without relying on predefined road network topologies. The time-delay cycle module learns the congestion propagation and fluctuation effects of traffic flow by incorporating delay information, thus adapting to the complex dynamics of traffic flow propagation under different scenarios. Subsequently, the dynamic and delayed spatial–temporal features are fused by the Transformer encoder-based spatial–temporal fusion module. Finally, the output layer adjusts the feature dimensions to match the predicted demand.

4.1 Multi-Scale Time Convolution Module

We propose a novel multi-scale time convolution module (MSTCM) specifically designed to capture long- and short-term temporal dependencies in traffic flow data. Unlike traditional models rely only on a sliding window design, MSTCM achieves efficient extraction of various temporal features through the synergy of two convolutional layers and residual connection. The long-term feature extraction convolutional layer processes the time series data in a recursive manner and gradually merges the data at each time step to efficiently capture the long-term dependencies. The short-term feature extraction convolutional layer focuses on modeling relationships between neighboring time points, thereby improving the short-term expressiveness of the model. The main advantages of this two-layer convolutional design are:

- Recursive processing of data: The long-term feature extraction convolutional layer connects to the previous hidden state at each time step, and gradually merges the

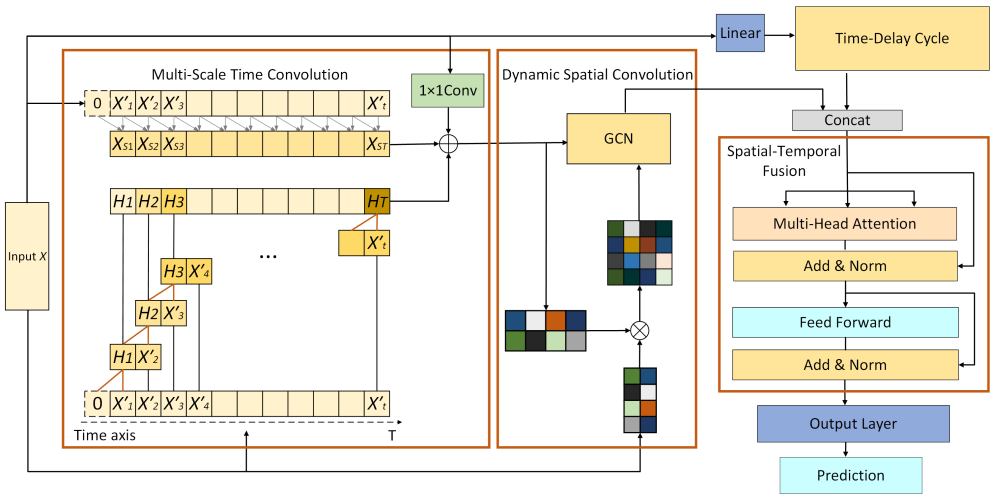

Fig. 2: The framework of STFFormer-GCN.

data from each time step, allowing the model to accumulate and integrate previous information, effectively capturing long-term dependencies.

- Modeling neighboring relationships: The short-term feature extraction convolutional layer performs convolution operations on the entire time series, focusing on modeling relationships between neighboring time points. This improves the model's ability to capture short-term dependencies and effectively capture subtle changes between local points in time.

The two convolutional layers maintain the same time dimension of inputs and outputs through the use of zero padding, which helps in fully preserving information and thus ensures that the model performs well when processing time series data of different lengths. The calculation formula is as follows:

$$X' = concat(\mathbf{0}, X) \in R^{1\times(T+1)\times N} \tag{2}$$

$$H_0 = X'\left[:, 0, :\right] \in R^{1\times 1\times N} \tag{3}$$

$$C_t = X'\left[:, t, :\right] \in R^{1\times 1\times N} \tag{4}$$

$$P_t = [H_{t-1}, C_t] \in R^{1\times 2\times N} \tag{5}$$

$$H_t = Conv_1(P_t) \in R^{1\times N} \tag{6}$$

where the input traffic flow feature matrix is $X \in R^{1\times T\times N}$. The Conv_1 convolution is a two-dimensional standard convolution with a convolution kernel size of 2×1 and a time step of $t = 1, ..., 12$. The long-term feature extraction convolutional layer enhances the long-term trend of temporal modeling by combining the output of the previous convolution kernel with the input at the current time. The output X_L of

the long-term feature extraction convolutional layer is obtained by splicing the hidden states of all time steps,as shown in Fig. 2:

$$X_L = [H_1, H_2, \ldots, H_T] \tag{7}$$

The short-term feature extraction convolutional layer uses a standard two-dimensional convolution with a convolution kernel size of 2×1 to extract short-term features between adjacent time points, which are calculated as follows:

$$X_S = Conv_2\left(X'\right) \tag{8}$$

Finally, the outputs of the two convolutional layers and the residual connection (conv3 convolution kernel is 1×1) are summed to get the final output:

$$X_Z = X_L + X_S + Conv_3\left(X'\right) \tag{9}$$

4.2 Dynamic Spatial Convolution Module

Existing GCN traffic prediction models are rely on predefined adjacency matrices A, which are calculated via distance functions or similarity measures. However, these methods have limitations because they are based solely on static neighborhood relationships, which only model the static structure of roads and cannot adapt to changes in the input data. To address this problem, we propose a new Dynamic Relational Convolution Matrix (DRCM). The dynamically generated adjacency matrix can be adjusted in real time according to the changes in input features, and the dynamic structural changes of the road are captured by the features processed by the multi-scale time convolution module. At the same time, the spatial attention mechanism can adaptively assign weights between nodes, so that the model can capture the dynamic relationships between nodes. The specific calculation process is as follows:

$$X_{i1} = X_Z W_Z \tag{10}$$

$$X_{i2} = X W_X \tag{11}$$

$$A = softmax\left(ReLU\left(X_{i2} X_{i1}^T\right)\right) + I_N \tag{12}$$

where I_N is the unit matrix, $X \in R_{N \times T}$ denotes the original traffic features, and $X_Z \in R_{N \times T}$ denotes the features after multi-scale convolution processing. W_Z and W_X are the learnable weight matrices used to map X and X_Z, respectively, to the new feature space. $W_Z \in R^{T \times d}$ and $W_X \in R^{T \times d}$, d are the new feature dimensions. Combined with the dynamic relational convolution matrix, the new dynamic spatial convolution module is implemented to learn the dynamic spatial relationship between nodes. The formula of GCN in dynamic spatial convolution is as follows:

$$Z = A X_Z \Theta + b \tag{13}$$

The output of this layer is $y_{s1} = Z$, $A \in R_{N\times N}$ is the dynamic relational convolution matrix of the graph, and $\Theta \in R_{T\times F}$ and $b \in R_F$ denote learnable weights and biases, respectively.

4.3 Time-Delay Cycle Module

In the real world, when an accident occurs on one road, it may take some time before traffic on the neighboring roads to be affected. To effectively model this phenomenon, we resort to the concept of delay-aware features, which successfully extracts the delay-related information R_t [11]. Our proposed time-delay cycle module is inspired by the Real-Time Gated Linear Loop Unit (RG-LRU) [26], which recursively combines the current traffic flow with the delay information. The time-delay cycle module merges the traffic flow with the acquired delay information, and the model is able to comprehensively consider the historical traffic flow information, thereby improving the accuracy of the prediction. The model weights the input information at different time steps by calculating the time delay attenuation factor a_t to better reflect the process of spatial information propagation, converting the input X into a high-dimensional representation through a fully connected layer.$X'' \in R^{N\times T\times D}$.

First, it is important to control the update rate of the hidden state, which determines how much of the hidden state of the previous moment is retained in the current moment and regulates the decaying effect of historical information in the time loop:

$$r_t = \sigma(h_{t-1} \cdot W_r + b_r) \tag{14}$$

Then, the delay information gating factor i_t is calculated for the current time step input to selectively introduce historical traffic pattern information to more accurately reflect the current traffic conditions:

$$i_t = \sigma(R_t W_t + b_t) \tag{15}$$

Next, the attention weight at is calculated to adaptively adjust the decay rate of the historical information in the hidden state according to r_t, so that the dynamic changes of traffic flow can be more effectively captured in time series:

$$a_t = a^{cr_t} \tag{16}$$

Finally, the hidden state h_t is updated to dynamically fuse the historical hidden state and the current traffic information by combining the delay information and current inputs. This updating process allows the model to accurately capture traffic flow trends and provide highly accurate traffic flow predictions:

$$h_t = a_t \odot h_{t-1} + \sqrt{1 - a_t^2} \odot (i_t \odot X_t'') \tag{17}$$

The output of this layer is denoted as $y_{s2} = h_t$, where W_r, b_r, W_t and b_t are the trainable parameters. We parameterize the parameter a in equation (16) as $a = \sigma(\Lambda)$, where Λ is the learnable parameter, and transform it by a nonlinear function σ (sigmoid

function). This ensures that $0 \leq a \leq 1$ and thus ensures stable maintenance of the hidden state. The variable c is a constant with a value of 8. At the beginning of training, we initialize Λ so that a^c is evenly distributed between 0.9 and 0.999.

4.4 Spatial-Temporal Fusion Module

In order to efficiently utilize the captured spatial–temporal dependencies of y_{s1} and y_{s2}, we employ the spatial–temporal fusion module to fuse the learned dynamic spatial–temporal features with the time delayed spatial–temporal features. We utilize the powerful feature capture and representation capabilities of Transformer's encoder to efficiently integrate the two types of feature data. The module contains a multi-head self-attention layer, a feed-forward neural network layer, and a layer normalization layer. First, we introduce the multi-head self-attention layer to enhance the model's ability to learn the importance of different spatial–temporal information. The formula for attention calculation is shown below:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{18}$$

In the multiple attention mechanism, the features are decomposed into, Q, K, and V, which denote the query, key, and value of all the nodes, respectively. The weights are computed using dot product and softmax functions and these weights are used in weighted value matrix to capture information in parallel in different subspaces. Then the residual connections are summed with the original inputs through the outputs of the attention layer and normalized by layers to enhance the stability of the model training. Subsequently, the feedforward network acts independently on the outputs of each location, increasing the nonlinear expressiveness of the model through two linear transformations and an activation function. Finally, the output of the feedforward network is again residual concatenated and layer normalized, and we stack the encoder structure of the Transformer by three layers to get the final feature-fused output y_z.

4.5 Output layer

In order to adapt the output dimension of the spatial–temporal encoder layer to the prediction needs, we introduce a multilayer perceptron (MLP) as the output layer of the model. This MLP structure consists of three fully connected layers and two nonlinear activation functions, and with this design, the model can nonlinearly transform the spatial–temporal fused features to the expected prediction dimension. In addition, in order to minimize the accumulation of errors during the prediction process, we employ a multi-step prediction strategy to ensure that the final output dimension matches the length of the prediction time window. Such an approach not only optimizes the prediction accuracy, but also enhances the model's adaptability when dealing with time-series data.

4.6 Loss function

This loss function consists of two parts: the first part $\parallel y_t - \widehat{y}t \parallel$ is used to quantify the error between the real traffic flow and the predicted value; the second part λLreg is the $L2$ regularization term, which is used by adjusting the hyper-parameter λ to enhance the model's generalization ability and avoid the overfitting issue. Where y_t and $\widehat{y}_t$ denote the real traffic flow and the predicted flow, respectively.

$$loss = \parallel y_t - \widehat{y}_t \parallel + \lambda L_{reg} \tag{19}$$

5 Experiments

5.1 Datasets

Four publicly available datasets are used in this paper to validate the predictive performance of the proposed model, which are derived from datasets from the California Traffic Management System; PeMS03, PeMS04, PeMS07and PeMS08, which generates 288 data points per day by measuring freeway traffic in California in real time every 5 minutes. This dataset was used as the data source for the study because it encompasses multiple freeway networks and provides a wide range of traffic flow data. The different datasets represent different regions and traffic conditions and provide a more complete reflection of traffic patterns. The specific parameters of the dataset are shown in Table 1.

Table 1: Dataset Overview

Dataset	Sensors	Edges	Time Range	Time Steps
PeMS03	358	547	09/2018–11/2018	26,208
PeMS04	307	340	01/2018–02/2018	16,992
PeMS07	883	866	05/2017–08/2017	28,224
PeMS08	170	295	07/2016–08/2016	17,856

5.2 Baselines

STFFormer-GCN achieved striking improvements on the PeMS03, PeMS04, PeMS07, and PeMS08 datasets compared to the previous 25 baselines, which can be broadly categorized into the following three categories:

- Conventional time series prediction techniques: HA[12], ARIMA[4], VAR[3], SVR[5].
- Graph neural network approaches: DSANet[27], STGCN[19], DCRNN[28], ASTGCN[20], MSTGCN[29], STSGCN[30], GraphWaveNet[31], AGCRN[21], MTGNN[32], STFGNN[33], STGODE[34], STGNCDE[35], Z-GCNETs[36], DSTAGNN[37], GDGCN[38].
- Methods incorporating Transformer technology: STTN[39], GMAN[40], TFormer[41], ASTGNN[42], PDFormer[11], STGAFormer[24].

5.3 Experimental settings

In this experiment, all experiments are performed on NVIDIA RTX 4090. The training, validation and test sets for each dataset are divided in 6:2:2. All training and testing use a 60-minute time window of data to predict traffic at 5-minute intervals over the next 60 minutes, i.e., 12 data points (T=12) are used to predict data for the next 12 observations (H=12). We optimize all the models using the Adam optimizer for a maximum of 500 epochs with a learning rate of 0.001 and the batchsize is set to 32.

5.4 Evaluation metrics

We use three metrics in the experiments: (1) Mean Absolute Error (MAE), (2) Mean Absolute Percentage Error (MAPE), and (3) Root Mean Squared Error (RMSE). Missing values are excluded when calculating these metrics.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|, \tag{20}$$

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \times 100\%, \tag{21}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}, \tag{22}$$

Where y_i and $\hat{y}_i$ are respectively the real traffic flow data at node i and the predicted value of the same node given by our model in a time step, and n is the number of nodes. When we calculate the overall performance of all nodes in the network over the next hour, we need to average the next 12 time steps.

5.5 Comparison results

To validate the performance of our model, we conducted experiments on four traffic datasets to compare the prediction performance of the STFFormer-GCN model with other baseline models. Referring to the official documents of most standard baseline models and published research results [24], Table 2 shows the comparison of the prediction performance of different models on these four datasets.

It is obvious that methods that incorporate multiple spatial–temporal features significantly outperform traditional linear-time prediction models because they are able to more fully utilize the dynamic spatial–temporal and delayed spatial–temporal feature characteristics in traffic data. In addition, two mainstream methods for traffic prediction are utilizing GCN and Transformer, both of which have achieved remarkable results. The work in this paper is also based on GCN and Transformer for traffic flow prediction and effectively combines dynamic spatial–temporal features and time delayed spatial–temporal features at the fusion stage of the model through the mechanism of multi-attention.The excellent performance of the STFFormer-GCN model in various metrics stems from the following aspects:

- Utilizeing multi-scale time convolution modele captures short-term and long-term temporal dependence, emphasizing on dealing with periodic features of traffic flow data at different time scales.
- Utilizeing dynamic spatial convolution module we propose DRCM to capture the spatial dependence between nodes by dynamically adjusting the adjacency matrix.
- Utilizeing the time-delay cycle module, we combine the time-delay information with the current traffic flow input to simulate the propagation process of traffic flow in space, which improves the long-term prediction effect.
- The transformer encoder-based fusion module is utilised to achieve deep fusion of multi-temporal and spatial features of the traffic flow, thus further improving the prediction performance of the model.

5.6 Ablation study

To further investigate the validity of the different modules of the STFFormer-GCN model, we conducted ablation experiments based on the PeMS08 dataset. This model was compared to the following variants: (1) STFFormer-GCN-M:This variant removes the multi-scale time convolution module ,thereby not accounting for time periodicity. (2) STFFormer-GCN-D: This variant removes the dynamic relational convolution matrix for dynamically generating adjacency matrices, using only adjacency matrices based on road connectivity relationships. (3) STFFormer-GCN-T: This variant removes the delay information gating factor i_t from the time-delay cycle module. (4) STFFormer-GCN-F: This variant removes the spatial–temporal fusion module, directly concatenates the dynamic spatial–temporal features and delayed spatial–temporal features, and then passes them to the dimension adjustment to the output layer.

As shown in Fig. 3, the STFFormer-GCN model showed excellent performance on all metrics. By introducing the spatio-temporal fusion module, the MAE is significantly improved, realizing an improvement of 17.93%. This highlights the need for fusing different spatial-temporal features as it effectively captures the relationship between dynamic and time delayed spatial-temporal features. Specifically, the spatial–temporal fusion module captures complex patterns and dependencies in traffic flow by integrating feature data from different space-time contexts. This fusion not only increases the prediction accuracy but also improves the model's generalization ability across different scenarios.

The multi-scale time convolution module also achieved significant results by combining long and short period information, especially in reducing the RMSE. This module can capture both short-term fluctuations and long-term trends, improving the overall stability and accuracy of forecasts. In addition, the time-delay cycle module played a crucial role in reducing the MAE value. Given the sensitivity of MAE to outliers, its significant decline highlights the advantages of the time-delay cycle module in dealing with the time delay of spatial information propagation. This module effectively captures the time delay effect in traffic flow, leading to more accurate predictions of future traffic conditions.

Table 2: Comparison Table of Model Performance

Model	PeMS03			PeMS04			PeMS07			PeMS08		
	MAE	RMSE	MAPE(%)	MAE	RMSE	MAPE(%)	MAE	RMSE	MAPE(%)	MAE	RMSE	MAPE(%)
HA	31.58	52.39	33.78	38.03	59.24	27.88	45.12	65.64	24.51	34.86	59.24	27.88
ARIMA	35.41	47.59	33.78	33.73	48.80	24.18	38.17	59.27	19.46	31.09	44.32	22.73
VAR	23.65	38.26	24.51	24.54	38.61	17.24	50.22	75.63	32.22	19.19	29.81	13.10
SVR	27.40	26.46	44.51	28.66	44.59	19.15	32.97	50.15	15.43	23.25	36.15	14.71
DSANet	21.29	34.55	23.21	22.79	35.77	16.03	31.36	49.11	14.43	17.14	26.96	11.32
STGCN	17.55	30.42	17.34	21.16	34.89	13.83	25.33	39.34	11.21	17.50	27.09	11.29
DCRNN	17.99	30.31	18.34	21.22	33.44	14.17	25.22	38.61	11.82	16.82	26.36	10.92
ASTGCN	17.34	29.56	17.21	22.93	35.22	16.56	24.01	37.87	10.73	18.25	28.06	11.64
MSTGCN	19.54	31.93	23.86	23.96	37.21	14.33	29.00	43.73	14.30	19.00	29.15	12.38
STSGCN	17.48	29.21	16.78	21.19	33.65	13.90	24.26	39.03	10.21	17.13	26.80	10.96
GraphWaveNet	19.12	32.77	18.89	39.66	31.72	17.29	26.39	41.50	11.97	18.28	30.05	12.15
AGCRN	15.98	28.25	15.23	19.83	32.26	12.97	22.37	36.55	9.12	15.95	25.22	10.09
MTGNN	15.85	26.23	15.55	19.08	31.56	12.96	20.82	34.09	9.03	15.40	24.93	10.17
STFGNN	16.77	28.34	16.30	19.83	31.88	13.02	22.07	35.80	9.21	16.64	26.22	10.60
STGODE	16.50	27.84	16.69	20.84	32.82	13.77	22.59	37.54	10.14	16.81	25.97	10.62
STGNCDE	15.57	27.09	15.06	19.21	31.09	12.76	20.53	33.84	8.80	15.45	24.81	9.92
Z-GCNETs	16.64	28.15	16.39	19.50	31.61	12.78	21.77	35.17	9.25	15.76	25.11	10.01
DSTAGNN	15.57	27.21	14.68	19.30	31.46	12.70	21.42	34.51	9.01	15.67	24.77	9.94
GDGCN	14.66	**24.30**	13.94	18.44	29.79	12.52	20.15	33.21	8.50	14.82	23.87	9.35
STTN	17.25	28.32	17.25	19.48	31.91	13.63	21.34	34.59	9.93	15.48	24.97	10.34
GMAN	16.87	27.92	18.23	19.14	31.60	13.19	20.96	34.10	9.05	15.31	24.92	10.13
TFormer	15.03	25.32	15.36	18.92	31.35	12.71	20.75	34.06	8.97	15.19	24.88	9.93
ASTGNN	14.78	25.00	14.79	18.60	30.91	12.36	20.62	34.00	8.86	15.00	24.70	9.50
PDFormer	14.94	25.39	15.82	18.32	29.97	12.10	19.83	32.87	8.53	13.58	23.51	9.05
STGAFormer	14.56	24.94	14.69	18.18	29.78	11.98	19.65	32.62	8.45	13.06	22.43	8.87
STFFormer-GCN	**14.28**	25.30	**13.90**	**18.05**	**29.67**	**11.62**	**19.35**	**32.11**	**8.25**	**12.81**	**21.84**	**8.51**

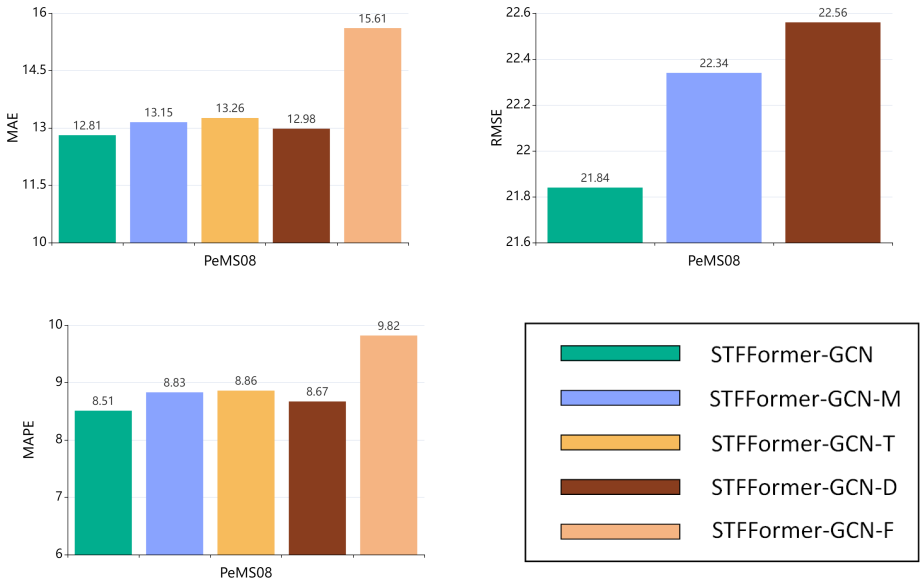

Fig. 3: Ablation study of key designs in STFFormer-GCN on the PeMS08 dataset.

In terms of spatial feature extraction, the dynamic spatial convolution module contributed significantly to the predictive capabilities of the model. This module captures current spatial dependencies by modeling traffic flow inputs at different times using DRCM. It adaptively adjusts the spatial strength of connectivity between nodes to accommodate traffic flow characteristics at different times, thus more accurately reflecting the dynamic changes in spatial characteristics.

5.7 Evaluation of model single-step prediction performance

In order to deeply analyze the prediction performance of STFFormer-GCN at different time steps, we conducted detailed experiments. Specifically, we calculated the MAE, RMSE and MAPE for the next 12 time steps and plotted the results as shown in Fig. 4. To ensure the fairness of the comparison, we cite the published research results [24] as the benchmark.

As can be seen from Fig. 4, STFFormer-GCN model exhibits certain limitations in short-term forecasting (e.g., 1 to 4 steps), with its MAE, RMSE, and MAPE metrics being slightly higher than those of existing models in some cases. This reflects some shortcomings of the model in capturing short-term temporal dependence. However, accurately capturing trends and changes over long periods of time is usually more important for traffic flow prediction, especially in scenarios such as decision making and resource scheduling, where the stability and accuracy of long term predictions are particularly critical.

As the prediction time steps increase, STFFormer-GCN significantly outperforms the existing models in longer time steps (e.g., 6 to 12 steps) with significantly lower

MAE, RMSE, and MAPE metrics than the other methods. This demonstrates the model's superior ability to handle long-time temporal dependence and to effectively fuse and capture long-term trends and spatial–temporal dependencies in traffic data. The synergistic effect of the multi-scale time convolution module, dynamic spatial convolution module, and time-delayed cyclic module enables STFFormer-GCN to show stronger adaptability and stability in long-term prediction tasks, which further validates its superior performance in long-term prediction.

5.8 Evaluation of Dynamic Relational Convolution Matrix

In order to demonstrate the effectiveness of our model in capturing dynamic spatial correlations at different times, we conducted comparative experiments using two model variants, STFFormer-GCN-D and STFFormer-GCN-A. STFFormer-GCN-D employs an adjacency matrix based on road connectivity, while STFFormer-GCN-A utilizes an adaptive adjacency matrix proposed by Bai et al., replacing our proposed DRCM. It is important to note that the parameters of the adaptive adjacency matrix are fixed post-training, hence both are considered static adjacency matrices. The performance comparison on the PeMS08 dataset is shown in Fig. 5. The DRCM-based model shows obvious advantages in the three metrics of RMSE, MAE and MAPE, which further proves the effectiveness of our proposed DRCM in modeling dynamic spatial relationships.

6 Conclusion

Our proposed traffic flow prediction model combining Graph Convolutional Network and Transformer methods shows clear superiority in solving the traffic flow prediction task. With the proposed multi-scale time convolution module and dynamic spatial convolution module, we successfully capture the dynamic temporal and dynamic spatial dependencies in traffic data. In addition, the design of the time-delay cycle module effectively models the time delay in the propagation of spatial information, which allows the model to more accurately reflect the variation of actual traffic flow. Extensive experimental results on four real-world datasets show that our model outperforms existing benchmark models in multiple evaluation metrics, demonstrating strong predictive capabilities and wide applicability. These experiments confirm the effectiveness of our model and demonstrate its superior performance in processing complex spatial–temporal data. Future research will focus on further optimizing the STFFormer-GCN model to consider more factors affecting traffic flow, such as weather, special events, etc., to improve the model's predictive ability in more complex environments.

Acknowledgements. This work was supported by the Research Foundation of Chongqing University of Science and Technology under Grant ckrc2019032.

References

[1] Medina-Salgado, B., Sánchez-DelaCruz, E., Pozos-Parra, P., Sierra, J.E.: Urban

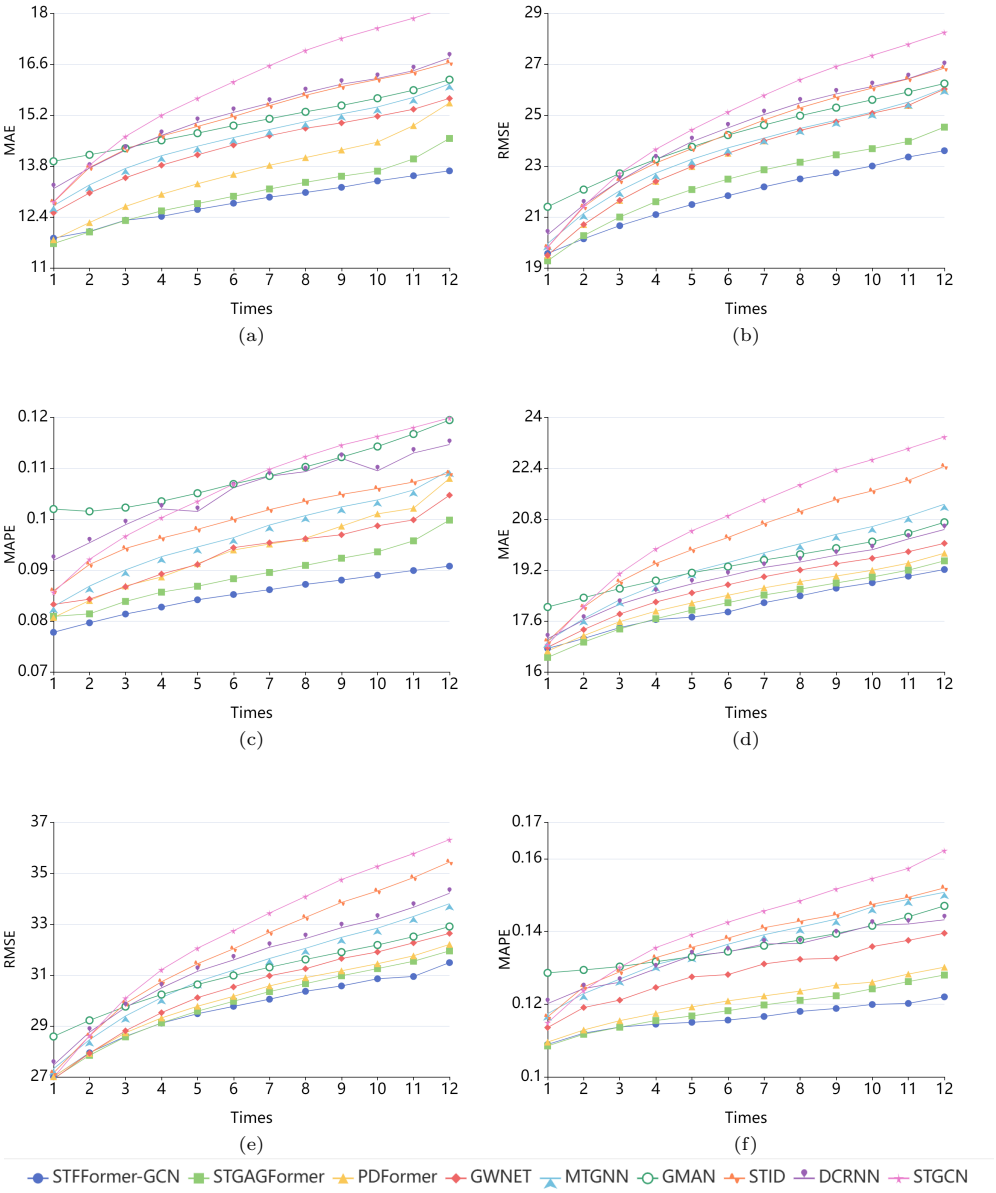

Fig. 4: Comparison of single-step prediction on different datasets. (a) MAE on PeMS08 (b) RMSE on PeMS08 (c) MAPE on PeMS08 (d) MAE on PeMS04 (e) RMSE on PeMS04 (f) MAPE on PeMS04

traffic flow prediction techniques: A review. Sustainable Computing: Informatics

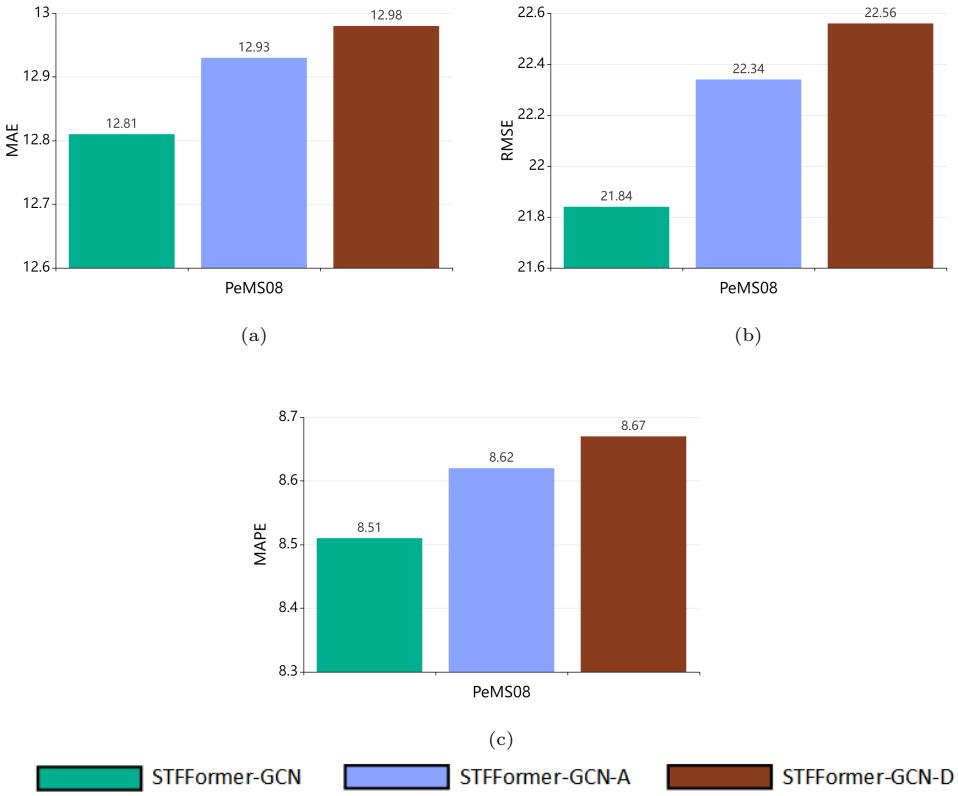

Fig. 5: Traffic flow prediction comparison of STFFormer-GCN models with three types of adjacency matrix construction. (a) Variation in MAE for different variants on the PeMS08; (b) Variation in RMSE; (c) Variation in MAPE.

and Systems **35**, 100739 (2022)

[2] Zhang, Y.: Short-term traffic flow prediction methods: A survey **1486**(5), 052018 (2020). IOP Publishing

[3] Lu, Z., Zhou, C., Wu, J., Jiang, H., Cui, S.: Integrating granger causality and vector auto-regression for traffic prediction of large-scale wlans. KSII Transactions on Internet and Information Systems (TIIS) **10**(1), 136–151 (2016)

[4] Williams, B.M., Hoel, L.A.: Modeling and forecasting vehicular traffic flow as a seasonal arima process: Theoretical basis and empirical results. Journal of transportation engineering **129**(6), 664–672 (2003)

[5] Dhiman, H.S., Deb, D., Guerrero, J.M.: Hybrid machine intelligent svr variants

for wind forecasting and ramp events. Renewable and Sustainable Energy Reviews **108**, 369–379 (2019)

[6] LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86**(11), 2278–2324 (1998)

[7] Yu, B., Lee, Y., Sohn, K.: Forecasting road traffic speeds by considering area-wide spatio-temporal dependencies based on a graph convolutional neural network (gcn). Transportation research part C: emerging technologies **114**, 189–204 (2020)

[8] Zaremba, W., Sutskever, I., Vinyals, O.: Recurrent neural network regularization. arXiv preprint arXiv:1409.2329 (2014)

[9] Li, W., Wang, X., Zhang, Y., Wu, Q.: Traffic flow prediction over muti-sensor data correlation with graph convolution network. Neurocomputing **427**, 50–63 (2021)

[10] Sun, L., Liu, M., Liu, G., Chen, X., Yu, X.: Fd-tgcn: Fast and dynamic temporal graph convolution network for traffic flow prediction. Information Fusion **106**, 102291 (2024)

[11] Jiang, J., Han, C., Zhao, W.X., Wang, J.: Pdformer: Propagation delay-aware dynamic long-range transformer for traffic flow prediction **37**(4), 4365–4373 (2023)

[12] Hamed, M.M., Al-Masaeid, H.R., Said, Z.M.B.: Short-term prediction of traffic volume in urban arterials. Journal of Transportation Engineering **121**(3), 249–254 (1995)

[13] Sun, Y., Leng, B., Guan, W.: A novel wavelet-svm short-time passenger flow prediction in beijing subway system. Neurocomputing **166**, 109–121 (2015)

[14] Tang, Y., Pan, Z., Hu, X., Pedrycz, W., Chen, R.: Knowledge-induced multiple kernel fuzzy clustering. IEEE Transactions on Pattern Analysis and Machine Intelligence (2023)

[15] Sun, B., Cheng, W., Goswami, P., Bai, G.: Flow-aware wpt k-nearest neighbours regression for short-term traffic prediction, 48–53 (2017). IEEE

[16] Lv, Z., Xu, J., Zheng, K., Yin, H., Zhao, P., Zhou, X.: Lc-rnn: A deep learning model for traffic speed prediction. **2018**, 27 (2018)

[17] Jiang, W., Luo, J.: Graph neural network for traffic forecasting: A survey. Expert systems with applications **207**, 117921 (2022)

[18] Zhao, L., Song, Y., Zhang, C., Liu, Y., Wang, P., Lin, T., Deng, M., Li, H.: T-gcn: A temporal graph convolutional network for traffic prediction. IEEE transactions on intelligent transportation systems **21**(9), 3848–3858 (2019)

[19] Yu, B., Yin, H., Zhu, Z.: Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. arXiv preprint arXiv:1709.04875 (2017)

[20] Guo, S., Lin, Y., Feng, N., Song, C., Wan, H.: Attention based spatial-temporal graph convolutional networks for traffic flow forecasting **33**(01), 922–929 (2019)

[21] Bai, L., Yao, L., Li, C., Wang, X., Wang, C.: Adaptive graph convolutional recurrent network for traffic forecasting. Advances in neural information processing systems **33**, 17804–17815 (2020)

[22] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)

[23] Yan, X., Gan, X., Tang, J., Zhang, D., Wang, R.: Prostformer: Progressive space-time self-attention model for short-term traffic flow forecasting. IEEE Transactions on Intelligent Transportation Systems (2024)

[24] Wang, T., Ni, S., Qin, T., Cao, D.: Transgat: A dynamic graph attention residual networks for traffic flow forecasting. Sustainable Computing: Informatics and Systems **36**, 100779 (2022)

[25] Geng, Z., Xu, J., Wu, R., Zhao, C., Wang, J., Li, Y., Zhang, C.: Stgaformer: Spatial–temporal gated attention transformer based graph neural network for traffic flow forecasting. Information Fusion **105**, 102228 (2024)

[26] De, S., Smith, S.L., Fernando, A., Botev, A., Cristian-Muraru, G., Gu, A., Haroun, R., Berrada, L., Chen, Y., Srinivasan, S., et al.: Griffin: Mixing gated linear recurrences with local attention for efficient language models. arXiv preprint arXiv:2402.19427 (2024)

[27] Huang, S., Wang, D., Wu, X., Tang, A.: Dsanet: Dual self-attention network for multivariate time series forecasting, 2129–2132 (2019)

[28] Li, Y., Yu, R., Shahabi, C., Liu, Y.: Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. arXiv preprint arXiv:1707.01926 (2017)

[29] Guo, S., Lin, Y., Feng, N., Song, C., Wan, H.: Attention based spatial-temporal graph convolutional networks for traffic flow forecasting **33**(01), 922–929 (2019)

[30] Song, C., Lin, Y., Guo, S., Wan, H.: Spatial-temporal synchronous graph convolutional networks: A new framework for spatial-temporal network data forecasting. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 914–921 (2020)

[31] Wu, Z., Pan, S., Long, G., Jiang, J., Zhang, C.: Graph wavenet for deep spatial-temporal graph modeling (2019)

[32] Wu, Z., Pan, S., Long, G., Jiang, J., Chang, X., Zhang, C.: Connecting the dots: Multivariate time series forecasting with graph neural networks, 753–763 (2020)

[33] Li, M., Zhu, Z.: Spatial-temporal fusion graph neural networks for traffic flow forecasting **35**(5), 4189–4196 (2021)

[34] Fang, Z., Long, Q., Song, G., Xie, K.: Spatial-temporal graph ode networks for traffic flow forecasting, 364–373 (2021)

[35] Choi, J., Choi, H., Hwang, J., Park, N.: Graph neural controlled differential equations for traffic forecasting **36**(6), 6367–6374 (2022)

[36] Chen, Y., Segovia, I., Gel, Y.R.: Z-gcnets: Time zigzags at graph convolutional networks for time series forecasting, 1684–1694 (2021). PMLR

[37] Lan, S., Ma, Y., Huang, W., Wang, W., Yang, H., Li, P.: Dstagnn: Dynamic spatial-temporal aware graph neural network for traffic flow forecasting, 11906–11917 (2022). PMLR

[38] Xu, Y., Han, L., Zhu, T., Sun, L., Du, B., Lv, W.: Generic dynamic graph convolutional network for traffic flow forecasting. Information Fusion **100**, 101946 (2023)

[39] Xu, M., Dai, W., Liu, C., Gao, X., Lin, W., Qi, G.-J., Xiong, H.: Spatial-temporal transformer networks for traffic flow forecasting. arXiv preprint arXiv:2001.02908 (2020)

[40] Zheng, C., Fan, X., Wang, C., Qi, J.: Gman: A graph multi-attention network for traffic prediction **34**(01), 1234–1241 (2020)

[41] Yan, H., Ma, X., Pu, Z.: Learning dynamic and hierarchical traffic spatiotemporal features with transformer. IEEE Transactions on Intelligent Transportation Systems **23**(11), 22386–22399 (2021)

[42] Guo, S., Lin, Y., Wan, H., Li, X., Cong, G.: Learning dynamics and heterogeneity of spatial-temporal graph data for traffic forecasting. IEEE Transactions on Knowledge and Data Engineering **34**(11), 5415–5428 (2021)