Hindawi
Journal of Advanced Transportation
Volume 2023, Article ID 8256907, 13 pages
https://doi.org/10.1155/2023/8256907

Research Article

Multibranch Adaptive Fusion Graph Convolutional Network for Traffic Flow Prediction

Xin Zan[1] **and Jasmine Siu Lee Lam**[2]

[1]*Antai College of Economics & Management, Shanghai Jiao Tong University, Shanghai 200030, China*
[2]*School of Civil and Environmental Engineering, Nanyang Technological University, Singapore*

Correspondence should be addressed to Jasmine Siu Lee Lam; sllam@ntu.edu.sg

Received 30 November 2022; Revised 4 May 2023; Accepted 17 May 2023; Published 13 June 2023

Academic Editor: Yajie Zou

Urban road networks have complex spatial and temporal correlations, driving a surge of research interest in spatial-temporal traffic flow prediction. However, prior approaches often overlook the temporal-scale differentiation of spatial-temporal features, limiting their ability to extract complex structural information. In this work, we design the multibranch adaptive fusion graph convolutional network (MBAF-GCN) that explicitly exploits the prior spatial-temporal characteristics at different temporal scales, and each branch is responsible for extracting spatial-temporal features at a specific scale. Besides, we design the spatial-temporal feature fusion (STFF) module to refine the prediction results. Based on the multibranch complementary features, the module adopts a coarse-to-fine fusion strategy, incorporating different spatial-temporal scale features to obtain recalibrated prediction results. Finally, we evaluate the MBAF-GCN using two real-world traffic datasets. Experimentally, the newly designed multibranch can efficaciously utilize the prior information of different temporal scales. Our MBAF-GCN achieved better performance in the comparative model, indicating its potential and validity.

1. Introduction

The rise in vehicle numbers has resulted in increased pressure on urban traffic and travelers. To improve traffic efficiency and reduce congestion, it is crucial to develop an effective and accurate traffic flow forecasting method [1]. Accurate traffic forecasts enable transportation agencies to improve road network capacity by setting up tidal sections and adjusting traffic signals dynamically to alleviate traffic congestion. Travelers can plan their routes with the foresight of road traffic conditions, while online car platform companies (such as DiDi and Uber) can anticipate traffic demand [2]. Traffic speed prediction has a wide range of applications in traffic management and control centers, including traffic monitoring, road condition broadcasting, and traffic control. It can help the traffic management center to better understand the current traffic situation and take corresponding measures in a timely manner to reduce traffic congestion and improve travel experience. Moreover, it can also be applied to traffic navigation and route planning to improve traffic efficiency and reduce energy consumption. Therefore, traffic state prediction is necessary.

Traffic flow prediction methods can be categorized into multistep and one-step prediction, depending on the forecasting temporal interval. Multistep prediction aims to predict long-term future road network information, while one-step prediction only indicates the future state in the next temporal step. Both methods require modeling the spatial-temporal correlation among nodes in a dynamic road network. Despite efforts to improve prediction accuracy, several challenges still require further exploration [3–5]. For instance, how can we better model complex spatial-temporal movement patterns among traffic data? How can we achieve high precision in long-term or multistep ahead traffic forecasting? How can we use complementary features or factors, such as weather changes, holidays, and traffic accidents, to enhance forecasting accuracy and robustness? This paper aims to effectively use complementary time-scale

features to improve the modeling of complex spatial-temporal correlations and promote multistep prediction accuracy.

There are two main types of forecasting methods, namely, model-driven based and data-driven based. Traditional time-series model-driven (parametric approach) models, such as autoregressive and AROMA (abstract representation of presence supporting mutual awareness) [6], are based on mathematical modeling and assumptions. However, they are insufficient to capture complex spatial-temporal correlations in raw data. These methods are also unsuitable for predicting data formed by non-Euclidean structural or other complex topologies. In comparison, the deep learning model has an advantage in handling high-dimension and nonlinear data. As shown in Figure 1, it is a typical example of deep learning work. The input data flow is processed through stacked or recurrent hidden layers and nonlinear activations and then mapped to the high-dimension feature space [7]. The network structure and feature fusion model enhance abstract features and produce final results, with learnable weights updated by back-propagation during training, and it extracts knowledge directly from raw data.

From the current study, deep learning models used for traffic flow prediction include graph convolutional networks (GCN), recurrent networks (RNN, LSTM), and transformer [8]. RNN-based methods capture all sequential states but struggle to identify local hidden features and emerging accumulated errors, a common issue with recurrent structures. Transformer-based methods excel in modeling long-term data processing and spatial correlation but require high computational and memory costs and long training times. GCN-based methods have gained popularity for their use of graph structures in road maps, enabling them to capture non-Euclidean data structures and complex spatial-temporal movement patterns. This simple architecture design is ideal for challenging problems such as long-term time series predictions.

Although studies have been conducted to predict traffic status using the primary GCN method, there still exist the following overlooked issues:

(1) Instinctively, the representation of time series data can be translated to the frequency domain. Take a one-dimensional timing signal as an example. The low-frequency part always represents the overall trend, and the high-frequency component represents fine-grained fluctuations, as shown in Figure 1. The upper figure illustrates the traffic flow variation captured by a sensor during a week in Los Angeles County, USA, and the lower figure shows the low-frequency part of the same data. We decompose it in the frequency domain by Daubechies-8 tap decomposition, assuming that the traffic flow observation period is observed in hours. In this case, the traffic flow is likely to show a relatively flat trend in the approaching period (these are the regular hours, ignoring the rush hour conditions). However, if observations are made in minutes, the traffic flow nodes will fluctuate sharply around the general trend. That is to say, different sampling windows correspond to different frequency filtering operations. Figure 1 shows that temporal-scale information exists in the frequency and time domains, and different temporal scales represent additional semantic information. Therefore, we can use the prior of temporal-scales and design a multibranch forecasting method to simplify the learning target: predicting trends and then predicting the fluctuation based on trends. We can design adaptive weights to fuse the complementary temporal-scale features to improve prediction accuracy further.

(2) By decomposing the time series into different time scales and using a multibranch structure to leverage the spatial-temporal features of the corresponding scales fully, we can obtain spatial-temporal expressions for different hierarchies. The extracted elements of each branch are then fused efficiently to generate the final prediction results. For this purpose, we designed the STFF (spatial-temporal feature fusion) module, which uses a coarse-to-fine strategy to adaptively fuse the spatial-temporal features of different scales. In this way, the temporal "trends" can be enhanced and recalibrated, and the spatial-temporal "details" can be refined.

Our main contributions in methodological and theoretical aspects are summarized as follows:

(1) We propose a multibranch prediction framework. Feature extraction branches are designed to take advantage of the complementary features at different temporal scales. This design intends to decompose the complex time series forecasting and reduce the model learning difficulty by explicitly exploiting the structural information in the temporal dimension;

(2) We propose the STFF module, which uses a coarse-to-fine strategy to fuse spatial-temporal elements of different scales and obtain the final refined prediction results;

(3) We validate this work using real-world data. There is a significant improvement in the two real-world datasets, and the prediction accuracy of the MBAF-GCN model exceeds the baseline.

The rest of this paper is as follows: in Section 2, we summarize the existing research on traffic flow prediction that is closely related to this paper. In Section 3, we define the traffic forecasting problem and present the methodology and the details of the proposed multibranch adaptive fusion graph convolutional network. In Section 4, we detail the dataset and experimental setup, analyze experimental results on real data, and conclude our work in Section 5.

2. Related Works

Graph neural network-based methods have become dominant in traffic forecasting research, due to their ability to

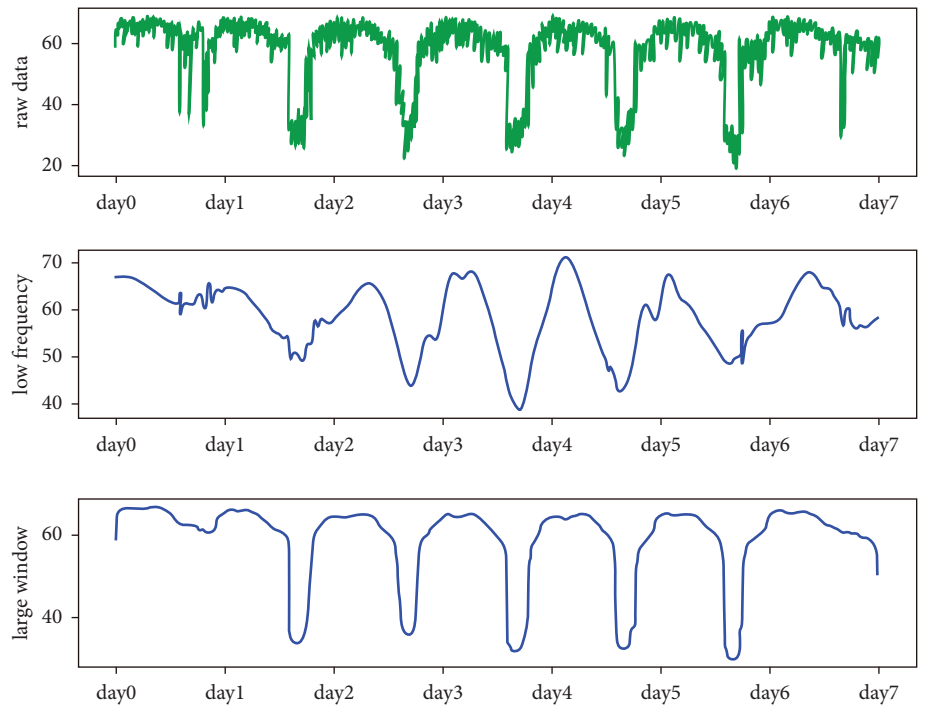


Figure 1: Low-frequency and high-frequency oscillations in time-series data.

capture spatial dependency in non-Euclidean graphs. Many methods have been proposed to leverage spatial-temporal characteristics fully. This section reviews relevant literature on GCN structure design and complementary feature exploitation.

Most GCN-based traffic prediction methods follow a straightforward single-branch network design. Tang et al. proposed a spatial-temporal graph convolutional network (STGCN) to solve the problem of traffic data chronology [5]. It overcomes the shortcomings of traditional traffic forecasting methods applied to the nonlinearity and complexity of traffic data. It integrates graphic convolution and gated temporal convolution to form the basic module of the network. Through spatial-temporal convolutional blocks, it integrates graphic convolution and gated temporal convolution. Qin et al. proposed the NDGCN model to use node information features to generate node embedding for unlabeled data. They learn a function that creates embedding by sampling and aggregating elements from the node's local neighborhood, which achieves a combined increase in performance and speed [9]. Yu et al. introduced 3D temporal graph convolutional networks (3D-TGCN), which verified the model's effectiveness in simplifying traffic data training and capturing spatial-temporal characteristics of traffic data [10]. Song et al. used Graph WaveNet for spatial-temporal graph modeling. It optimizes the shortcomings of traditional models in capturing the features of long-time series trend data by combining graph convolution with dilated traditional convolution [11]. Zhu et al. proposed the AST-GCN algorithm. They used the framework to aggregate and transform features during training, reducing training time, and memory complexity. They further optimized to the GCN, which can automatically adjust per layer in the GCN, further reducing the training time by half [12]. Chen et al. also used a semantic-interactive graph convolutional structure [13]. Wang et al. solved how to determine appropriate neighborhoods to improve the graph structure. They proposed the GraphHeat concept to enhance the smoothness of the signal on the graph structure [14]. Jepsen et al. offered relational fusion networks (RFNs) with a different adjacency matrix at different levels. The adjacency matrix can be continuously learned during training [15]. Bai et al. introduced external factors and proposed an attribute-augmented spatial-temporal graph convolutional network (A3T-GCN). We separate the dynamic and static attributes of external factors to verify that the model can effectively perceive the influence of external factors [16]. Lee and Rhee proposed the spatial node association's distance, direction, and positional relationship graph convolutional network (DDP-GCN) model. The model can automatically extract node-related features in traffic data, and the removed parts can be dynamically adjusted [17]. Lin et al. proposed spatial-temporal fusion graph neural networks. An extension of the GNN model captured the complex spatial dependencies and dynamic trends of road networks [18]. Guo et al. also extended the GNN-based model and proposed an attention-based spatial-temporal graph neural network (ASTGNN) [19].

A few methods have begun to pay attention to using the multibranch GCN to exploit complementary spatial-temporal features. Guo et al. construct a two-stream graph network to consider micro and macrotraffic complementary information [19]. Micro refers to traffic sensors, and macro refers to traffic regions. The difference between our works is that they conduct the splitting in the spatial dimension, while we focus on exploiting complementary

temporal features. Ioannidis et al. also focus on using hierarchical characteristics of spatial-temporal features and simultaneously predicting the fine-grained and coarse-grained traffic conditions over a road network [20]. Although considering the coarse-grained and fine-grained structural information in spatial-temporal dimension, they explicitly build the multiscale network by topology closeness and the traffic flow similarity.

In contrast, the network scale-specific feature in our work is learned and recalibrated in an end-to-end form without manual intervention. Ke et al. introduce deformable convolution to enhance the modeling capability of spatial nonstationarity and design a multibranch network to model temporal dependency, including weekly trend, daily periodicity, and hourly closeness [21]. Like Jeon and Hong [22], they follow an artificial assumption in splitting spatial-temporal scales. All these works proved the effectiveness of conducting a multibranch network structure design to take advantage of the complementary spatial-temporal hierarchical characteristics fully.

There is also a lot of work that uses transformer-based model structures to solve traffic prediction tasks. For example, [23] uses multiple spatiotemporal attention blocks to construct the encoder and decoder of the model and applies a transform attention layer between the encoder and decoder to perform feature transformation. In [24], spatial and temporal transformers construct the base feature extraction module, which can address the existing flaws in spatio-temporal dependency. Transformer-based structure to capture spatiotemporal dependencies: a novel self-attention mechanism that is capable of utilizing the local context in the temporal dimension, and a dynamic graph convolutional module that incorporates self-attention in the spatial dimension. In [25], Zhang et al. uses self-attention to capture both short-term and long-term temporal correlations, and the proposed temporal fusion transformer has great advantages over traditional prediction models when the prediction horizon is longer than one hour. Although these transformer-based works have great advantages over long-term traffic prediction problems, the transformer structures have generally high time costs, which limit the algorithms to edge devices with insufficient computational power, and applications with high requirements for realism. This limits the scalability of the algorithms for applications with insufficient computational power and high-performance requirements.

While previous research has incorporated spatial-temporal scales, most models only use a single-branch approach and do not consider the variability of scales. Although some studies have used multibranch GCN approaches, they do not effectively decompose temporal trends. Furthermore, existing models face challenges in optimizing learning speed and capturing spatial-temporal characteristics for long-term trend data. This paper addresses these challenges by incorporating coarse-to-fine integration of different spatial and temporal scales, improving model learning effectiveness.

3. Methodology

3.1. Problem Definition. The mathematical definition of the traffic prediction problem is described then. We will introduce our framework and two key components, namely, spatial-temporal feature extraction branch and the spatial-temporal feature fusion module.

We will solve the multistep traffic forecasting problem. We use $X_{n,t}^{c} \in R^{N\times T\times C}$ to denote the series of traffic data collected by N sensors in a region during a period. $t \in (1, 2, ..., T)$ denotes the temporal sampling interval of the sensors. $c \in R^{C}$ indicates the dimension of traffic information of interest (e.g., speed, volumes, and flows), and then $\chi = \{X_{:,0}^{c}, X_{:,1}^{c}, ..., X_{:,t}^{c}, ...\}$ represents all traffic data collected in the same region during time t. As mentioned before, we predict future values of related traffic sequences based on historical observation. Therefore, we can formulate this aim as finding a function F to predict the subsequent τ steps based on the past T steps of historical data.

$$\left\{X_{:,t+1}^{c}, X_{:,t+2}^{c}, ..., X_{:,t+\tau}^{c}\right\} = F_{\theta}\left(X_{:,t}^{c}, X_{:,t-1}^{c}, ..., X_{:,t-T+1}^{c}\right), \tag{1}$$

where θ is the learnable parameter of the model; due to the spatial and temporal complex correlation between the nodes in the region, we adapt the GCN-based model and form the graph structure $G = (V, e, A)$, where V, e, and A represent the adjacency matrices of a set of vertices, a set of edges, and G, respectively. Therefore, this problem is expressed in the form of graph solving as follows:

$$\left\{X_{:,t+1}^{c}, X_{:,t+2}^{c}, ..., X_{:,t}^{c}\right\} = F_{\theta}\left(X_{:,t-T+1}^{c}, X_{:,t-T+2}^{c}, ..., X_{:,t}^{c}; G\right). \tag{2}$$

Following the spectral graph theory, the graph Laplacian matrix, eigenvalues, and eigenvectors are the theoretical basis of graph convolution. And different types of Laplacian matrices can be divided into three categories as follows:

(a) Un-normalized Laplacian, which is also called combinatorial Laplacian, formulated as $L = D - A$

(b) Normalized Laplacian, a normalized style and commonly used form in the GCN, formulated as $L = D^{-(1/2)} - LD^{-(1/2)}$

(c) Random walk normalized Laplacian, which has similarities with diffusion-convolution expressed as [5]

$$L = D^{-1}L. \tag{3}$$

Among them, $A \in R^{n\times n}$ presented as an adjacent matrix and $D \in R^{n\times n}$ presented as a diagonal degree matrix. The feature decomposition of the Laplacian matrix can obtain its eigenvector matrix U and eigenvalue matrix Λ, so the Laplacian matrix can be expressed as $L = U\Lambda U^{T}$, where $\Lambda \in R^{N\times N}$ is a diagonal matrix and $U \in R^{N\times N}$ is Fourier basis.

We formulate a graph convolutional filter $g_\theta = \mathrm{diag}(\theta)$ parameterized by $\theta = R^N$. Hence, the graph convolution of x defined in the Fourier domain is $g_\theta * Gx = Ug_\theta U^T x$. Here, $*G$ denotes a graph convolution operation and $U^T x$ is the graph Fourier transform of x.

However, the computational complexity of such an operation is too large, so the Chebyshev approximation is generally used (note: the form of the Chebyshev polynomialis $T_0(x) = 1, T_1(x) = x, T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x)$).

$$g_{\theta'} \approx \sum_{k=0}^{K} \theta_k T_k(\tilde{\Lambda}). \tag{4}$$

Then, the graph convolution can be expressed as the Chebyshev approximation filter as follows:

$$g_{\theta'} * x \approx \sum_{k=0}^{K} \theta_k T_k(\tilde{L})x. \tag{5}$$

In general, we only take $k = 1$ to avoid further complexity. That is to say, the first-order approximation of spectral graph convolution is used. The formula changes to

$$\begin{aligned} g_{\theta'} * x &\approx \sum_{k=0}^{1} \theta_k T_k(\tilde{L})x \\ &= \theta_0 x + \theta_1 (L - I)x \\ &= \theta_0 x - \theta_1 D^{-1/2} A D^{-1/2} x. \end{aligned} \tag{6}$$

Since θ is the filter's parameter, so $\theta = \theta_0' = -\theta_1'$. Then, the formula changes to

$$g_{\theta'} * x \approx \theta\left(I + D^{-1/2} A D^{-1/2}\right)x. \tag{7}$$

3.2. Model. As shown in Figure 2, the framework of our model mainly contains two parts, namely, multibranch spatial-temporal feature extraction and spatial-temporal feature fusion module. Each branch is a response to a specific temporal scale. We achieved this by carefully setting the temporal receptive field and designing a global attention module to guide a branch's GCN layers. Specifically, GCN layers in each branch are guided by different high-level attention maps. We construct the global graph attention module to generate different adjacency matrices for each branch, making them specific for feature extraction and prediction at a particular scale.

We classify the feature branches into three types, namely, tendency branch, coarse branch, and fine branch, based on the temporal scales setting from coarse to fine. To balance the trade-off of prediction accuracy and efficiency, we first use a self-attention structure to extract the global correlation from the input data and provide global guidance for the subsequent graph convolutional module. Then, each branch uses CNN-based blocks to extract local spatial-temporal features, ultimately achieving effective prediction.

Each branch's original input is extracted as scale-specific features and then fused by the adaptive spatial-temporal fusion module. As Figure 2 shows, we follow the coarse-to-fine fusion strategy, which is to fuse tendency branch features and coarse branch features first, and then the enhanced features are fused with the fine branch subsequently. Through the spatial-temporal fusion layer, the initial prediction information will be recalibrated and refined. We will describe the abovementioned modules in detail.

3.2.1. Spatial-Temporal Feature Extraction Branch. The entire spatial-temporal feature extraction branch consists of several stacked feature extraction blocks and one global attention guidance module. We use the gated TCN without a dilation ratio setting as the temporal feature extraction module, unlike Graph WaveNet [26]. We disagree with the use of dilation ratios in convolutions to expand the receptive field. Instead, we suggest designing a global spatial-temporal module to provide global guidance, as the receptive field is critical for modeling complex correlation patterns between nodes. Our approach involves self-attention, allowing any node in the sequence to access any other nodes based on the long-term correlation matrix. Therefore, there is no need to use a dilation ratio to expand the receptive field. The feature extraction block's formal expression is as follows:

We got the raw historical traffic data $\chi \in R^{N \times D \times S}$, we extract temporal features as the form

$$h = g(\Theta_1 * \chi + b) \odot \sigma(\Theta_2 * \chi + c), \tag{8}$$

where Θ_1, Θ_2, b, and c are learnable weights and biases of normal temporal convolutional layers. $\odot$ is the element operates. $g(\cdot)$ is the tanh activation function, and $\sigma(\cdot)$ is the sigmoid activation function that acts as a temporal gate, which determines the ratio of information that passes to the next layer.

Exploiting spatial correlation, the GCN module guided by global attention is used in the spatial feature extraction part. We follow the default setting as Graph WaveNet, except using the global attention matrix as an additional adjacency matrix. Its formal expression is as follows:

$$Z = AXW. \tag{9}$$

Z is the spatial feature results, A is the adjacent matrix, and W is the learnable weights.

We have designed three branches that are specific to different spatial-temporal scales, namely, the tendency branch, coarse branch, and fine branch. The tendency branch utilizes a temporal convolutional kernel size equal to the length of the entire historical sequence (12 in our work) to extract global information for tendency feature extraction. In contrast, smaller kernel sizes are used in the coarse and fine branches to extract more detailed information from the local neighborhood. Each branch also receives intermediate supervision specific to its scale. To achieve this, we remove high-frequency components of labels with different degrees.

3.2.2. Global Graph Attention Module. To provide scale-aware attention guidance in each branch, as shown in Figure3, we embed a global graph attention module, which undergoes convolutional filtering of different kernel sizes in

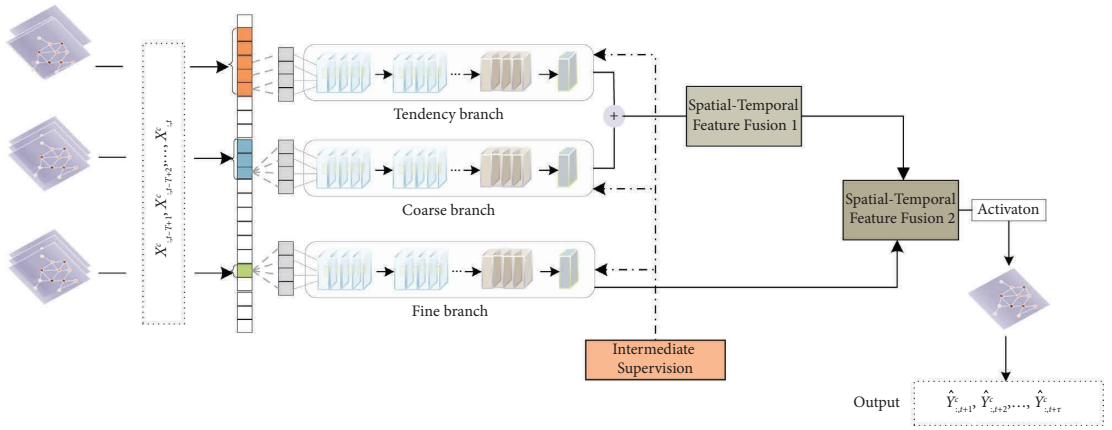


Figure 2: Multibranch spatial-temporal graph convolutional network.

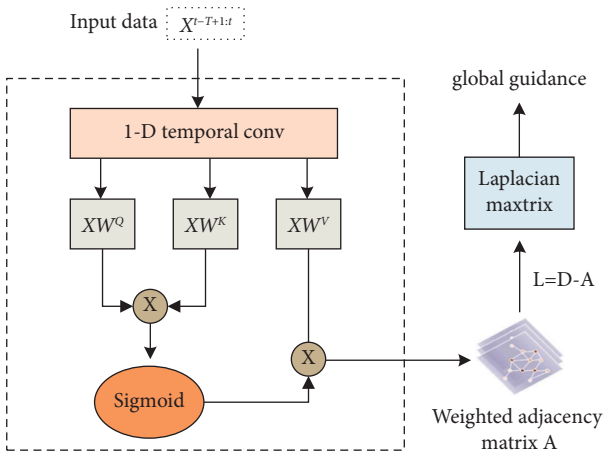


Figure 3: Get Laplacian matrix by the global graph attention module.

the temporal dimension. To improve performance, we use multihead attention to establish dependencies among every element and express the information of different subspaces. This stabilizes the learning process and ensures the suitability of the graph convolutional network for our purposes, as the adjacent matrix can significantly affect performance.

Given the input $X^{t-T+1:t} = [X^{t-T+1:t}, ..., X^t] \in R^{T*N*P}$, we simplify the notation as X. We use a 1-D temporal convolutional block to convert the input features into higher-dimension features on each node in the initial step. Notably, different branches have different kernel sizes to achieve scale-specific. As demonstrated in Figure 4, the tendency branch has the largest temporal kernel size setting, and the fine branch has the smallest temporal kernel size setting.

$$\overrightarrow{X}^{t-T+1:t} = W_k \otimes X, \tag{10}$$

where $\otimes$ refers to the convolutional operation, W refers to the learnable parameters of convolutional layers, and k refers to kernel size.

Then, three subspaces are obtained, namely, query subspace spanned by $Q \in R^{N*d_q}$, key subspace $K \in R^{N*d_k}$, and value subspace $V \in R^{N*d_v}$. The latent subspace learning process can be formulated as

$$Q = \overrightarrow{X}W^Q, K = \overrightarrow{X}W^K, V = \overrightarrow{X}W^V, \tag{11}$$

where $W_q{}^s \in R^{dG \times d_A^s}, W_k{}^s \in R^{dG \times d_A^s}$ and $W_v{}^s \in R^{dG \times dG}$ are the weight matrices for Q^s, K^s, and V^s, respectively.

Scaled dot-product attention is used to compute the global attention matrix as follows:

$$\text{Attention}(Q, K, V) = \text{soft max}\left(\frac{QK^T}{\sqrt{d_K}}\right)V. \tag{12}$$

After we use multihead design to get richer latent information, we concatenate all attentions together and project again to get the final values:

$$\begin{aligned} \text{MultiHead}(Q, K, V) &= \text{Concat}(\text{head}_1, ..., \text{head}_h)W^O, \\ \text{where head}_i &= \text{Attention}(Q_i, K_i, V_i). \end{aligned} \tag{13}$$

The attention map generated by the global attention modules is used for global guidance for spatial-temporal extraction blocks followed up.

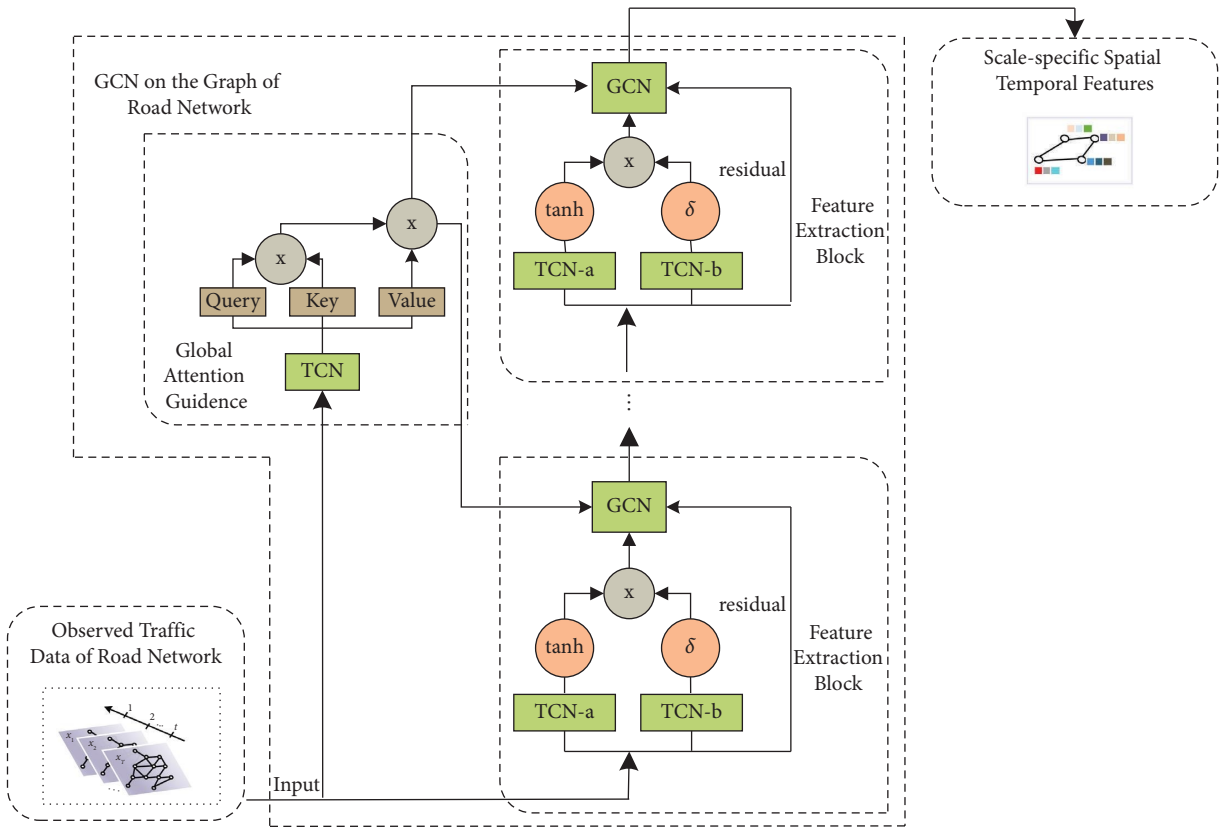


Figure 4: Spatial-temporal feature extraction branch.

3.2.3. Spatial-Temporal Feature Fusion Module. The fusion between the tendency branch and the coarse branch is similar.

We designed the multibranch spatial-temporal fusion module to achieve coarse-to-fine feature fusion. The coarse-to-fine strategy exploits the temporal scale prior to refining prediction results.

As previously mentioned, we obtain scale-aware features from the multibranch network. These features correspond to different temporal semantic meanings based on the sampling windows. We first fuse the features extracted from the tendency branch with those extracted from the coarse branch. Then, we fuse the resulting features with features similarly extracted from the fine branch. The complementary features from different branches are adaptively merged and enhanced through this fusion process.

The specific fusion module design is shown in Figure 5. We illustrate our fusion method by fusing the coarse and fine branches. We use the temporal attention matrix extracted from the coarse branch to enhance the relevant parts of the features in the fine branch and avoid any loss of detailed information with residual links.

The formal expression of the fusion model is as follows:

Given the coarse branch feature map $F \in R^{C\times H\times W}$ as input, fusion sequentially infers a 1-D temporal attention map $M_c \in R^{C\times 1\times 1}$, as illustrated in Figure 5. The overall attention process can be summarized as

$$F' = M_c(F) \otimes F, \tag{14}$$

where $\otimes$ denotes element-wise multiplication. The following describes the details of each attention module:

$$\begin{aligned} M_c(F) &= \sigma(\mathrm{MLP}(\mathrm{AvgPool}(F)) + \mathrm{MLP}(\mathrm{MaxPool}(F))) \\ &= \sigma\left(W_1\left(W_0\left(F^c_{\mathrm{avg}}\right)\right) + W_1\left(W_0\left(F^c_{\mathrm{max}}\right)\right)\right), \end{aligned} \tag{15}$$

where σ denotes the sigmoid function, $W_0 \in R^{C/r\times C}$ and $W_1 \in R^{C\times C/r}$ are the learnable parameters. Note that the MLP weights, W_0 and W_1 are shared for both inputs, and the ReLU activation function is followed by W_0. After the final fusion, supervision between prediction and ground truth labels are processed in the training phase.

4. Experiments

4.1. Dataset Description. In this section, we evaluate the performance of the MBAF-GCN model on two real-world traffic datasets, namely, METR-LA (ranging from Mar 1st, 2012, to Jun 30th, 2012) and PEMS-BAY (ranging from Jan 1st, 2017, to May 31st, 2017). Each dataset contains key attributes of the transportation network and time-stamped geographic information. METR-LA provides traffic speed and volume data collected by 207 loop detectors on the Los Angeles County highway network, and PEMS-BAY provides Bay Area data collected by 325 sensors. METR-LA and PeMS-BAY are commonly used datasets in the field of traffic flow prediction, providing rich data and scenarios for evaluating and comparing the performance and effectiveness of different algorithms. These datasets have large amounts of

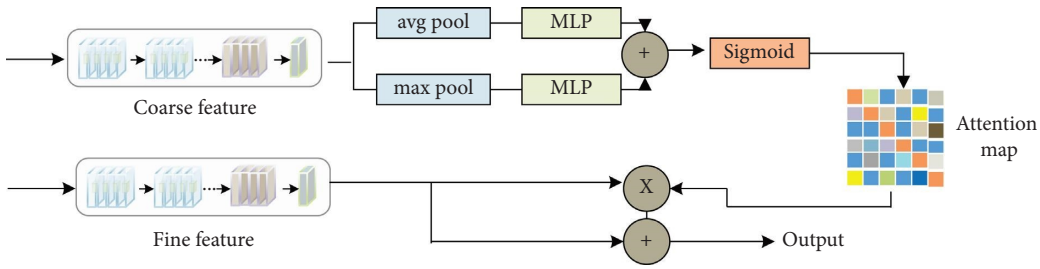


Figure 5: The spatial-temporal feature fusion module between coarse and fine branches.

real-time traffic data, enabling researchers to use machine learning and deep learning techniques to predict and analyze traffic flow. We extracted traffic speeds from both datasets and aggregated them into 5-minute intervals, and applied Z-score normalization. The sensor distribution of the datasets is visualized in Figure 6.

4.2. Experiment Settings. The experiments were conducted on a Linux operating system running on an Intel(R) Xeon(R) CPU E5-2640 v4 @ 2.40 GHz and an NVIDIA GeForce GTX 1080 GPU. The entire work was built using the open-source Python machine-learning library, Pytorch. To obtain the best parameters and avoid overfitting, we selected some data augmentation strategies, such as injecting small noise/outliers into time series using random noise perturbations to improve the model's robustness. We also performed a grid search to locate the optimal parameters.

During the training phase, we used the Adam optimizer and mean square error as the loss function. The initial learning rate was set to $10e-4$, with a decay rate of 0.7 after every 15 epochs. The model's input was the historical traffic speed data, and the output was the predicted traffic speed values for a certain time interval (15 min/30 min/1 hour) in the future.

We evaluated the performance of the MBAF-GCN model on two real-world traffic datasets, namely, METR-LA and PEMS-BAY. The datasets contain key attributes of the transportation network and time-stamped geographic information. The sensor distribution of the datasets is shown in Figure 6. We aggregated the traffic speed data into 5-minute intervals and applied Z-score normalization. The datasets were split chronologically into 70% for training, 10% for validation, and 20% for testing.

4.2.1. Evaluation Metric. To evaluate the performance of different methods, this work employs mean absolute error (MAE), mean absolute percentage error (MAPE), and root mean square error (RMSE) metrics, defined as

$$\begin{aligned} \text{MAE} &= \frac{1}{n}\sum_{t=1}^{n}\left|v_t - \tilde{v}_t\right|, \\ \text{MAPE} &= \frac{1}{n}\sum_{t=1}^{n}\left|\frac{v_t - \tilde{v}_t}{v_t}\right| \times 100, \\ \text{RMSE} &= \left[\frac{1}{n}\sum_{t=1}^{n}\left(v_t - \tilde{v}_t\right)^2\right]^{1/2}, \end{aligned} \tag{16}$$

where v_t is the speed of the detected vehicle and $\tilde{v}_t$ is the predicted vehicle speed.

4.2.2. Baselines. We compare our framework MBAF-GCN with the following baselines: (1) HA: historical average method, which models historical traffic as a seasonal process, and then uses the weighted average of historical seasons as the forecast value [27]; (2) ARIMA: with Kalman filter [28]; (3) SVR: based on historical data, SVR uses linear support vector machines to train models, establish input-output relationships, and then make predictions [29]; (4) FC-LSTM: recurrent neural networks with fully connected LSTM hidden units [30]; (5) WaveNet: its main component is causal convolution, which is a convolutional network architecture for sequence data [31]; (6) DCRNN: a diffusion convolutional recurrent neural network that captures temporal dependencies with graph convolutions formalized by a diffusion process and spatial dependencies with an encoder-decoder framework [32]; (7) GGRU: graph-gated recurrent unit network [30]; (8) STGCN: a spatial-temporal graph convolutional model based on a fixed Laplacian matrix to capture spatial-temporal features [33]; (9) Graph WaveNet: integrating diffusion convolution and 1-D dilation convolution to capture spatial-temporal correlations [34].

Figures 7 and 8 visually depict the traffic flow prediction for a randomly selected set of road sections in METR-LA over one day. The green and yellow segments of the figure represent the local visualizations of the prediction results during the hourly periods of smooth and drastic traffic

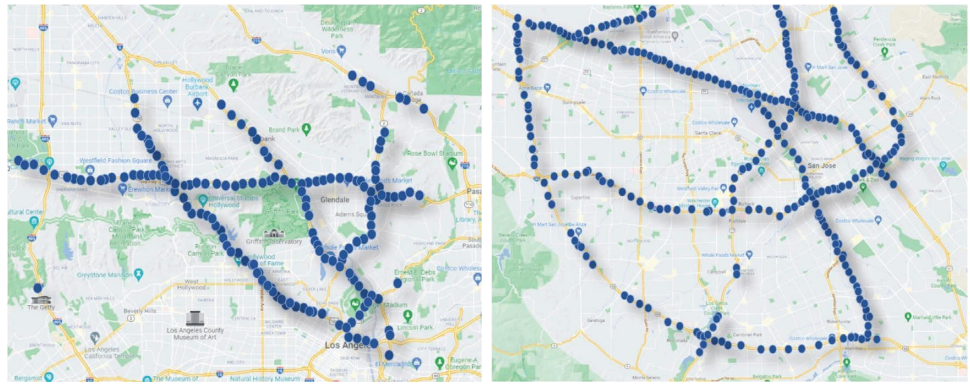


Figure 6: Sensor distribution of the METR-LA and PEMS-BAY dataset.

changes, respectively. The prediction curves generated by our proposed method and GraphWaveNet demonstrate proximity to the ground truth during the period of smooth traffic changes [34]. However, during the period of severe traffic changes, our method outperforms GraphWaveNet in accurately fitting the ground truth curves. This outcome can be attributed to the effective extraction of complementary and discriminative features through the implementation of the temporal multiscale structure in our method.

4.2.3. Experimental Results Analysis. We validated our model and nine baselines on the datasets METR-LA and PEMS-BAY for 15 minute, 30 minute, and 60 minute ahead predictions and present the results in Table 1.

Table 1 shows that traditional time-series prediction techniques (HA, VAR, and SVM) have the lowest accuracy and are inadequate for modeling nonlinear and complex spatial-temporal relationships. CNN-LSTM, an early deep learning technique, can significantly enhance prediction accuracy by avoiding artificial assumptions and learning valuable features from data but is still incapable of modeling intricate spatial-temporal correlations. GCN-based schemes (STGCN and MSTGCN) offer higher accuracy due to their superiority in modeling complex nonlinear non-Euclidean distance structures and simultaneously modeling spatial and temporal dependencies. Recent studies, such as Graph WaveNet, have limitations in modeling complex spatial-temporal movement patterns. In contrast, MBAF-GCN, which captures complementary temporal dependencies and utilizes the coarse-to-fine fusion design, outperforms these baseline schemes in terms of prediction accuracy. The model significantly outperforms Graph WaveNet on the 30 and 60 minute-ahead predicted values and is on par with it on the 15 minute horizon. To better illustrate the predictive power of different models, we visualized the MAE, MSE, and MAPE prediction errors of different models on the METR-LA dataset and PEMS-BAY. It can be visualized that the prediction accuracy of Graph WaveNet and MBAF-GCN is significantly lower than that of other control methods. At the same time, MBAF-GCN is more advantageous at different time intervals, especially at 30 min and 60 min, and reaches the lowest prediction accuracy.

4.2.4. Ablation Experiments. To verify the effectiveness of the proposed module in our scheme, we conducted two sets of ablative experiments as follows:

(1) Effect of the Multibranch Structure. To verify the efficacy of the proposed multibranch structure, we established a control group using a global graph attention module and spatial-temporal layers in a single branch form. The objective was to examine whether the accuracy enhancement resulted from the complementary temporal-scale features. In the multibranch settings, we created a single-branch model with nearly identical parameters (notably, we adjusted the convolution parameters in each module to ensure consistency in the output dimensionality with the multibranch), while keeping the other parameters constant. For the experimental group, which featured the MBAF-GCN prototype structure, we eliminated the fusion module from the experimental setup and replaced it with an additional operation to prevent any interference with the experimental outcomes due to the introduction of additional covariates in the fusion module. We utilized the addition operation to merge features from multibranches. We conducted controlled experiments on the two datasets, and the experimental results are shown in the following form (Table 2).

The experimental findings suggest that the prediction accuracy is enhanced by the explicit exploitation of structural information of spatial-temporal features using a multibranch design with a parallel multiscale structure, which outperforms the sequence homogeneous structure network with comparable parameters. In addition, the results highlight the effectiveness of utilizing the temporal scale as prior knowledge.

In this study, a global graph attention module is integrated into each branch to provide global spatial-temporal attention guidance and scale-aware attention. To investigate the impact of spatial-temporal scaling-specific modeling, the heatmap of the adjacent weighted matrix in each branch is visualized. Specifically, Figure 9 depicts the heatmap of the global graph attention module in the tendency branch, coarse branch, and fine branch, which are learned from the METR-LA dataset under the same input settings.

The results demonstrate that the attention module in the tendency branch (Figure 9(a)) exhibits high values on distant

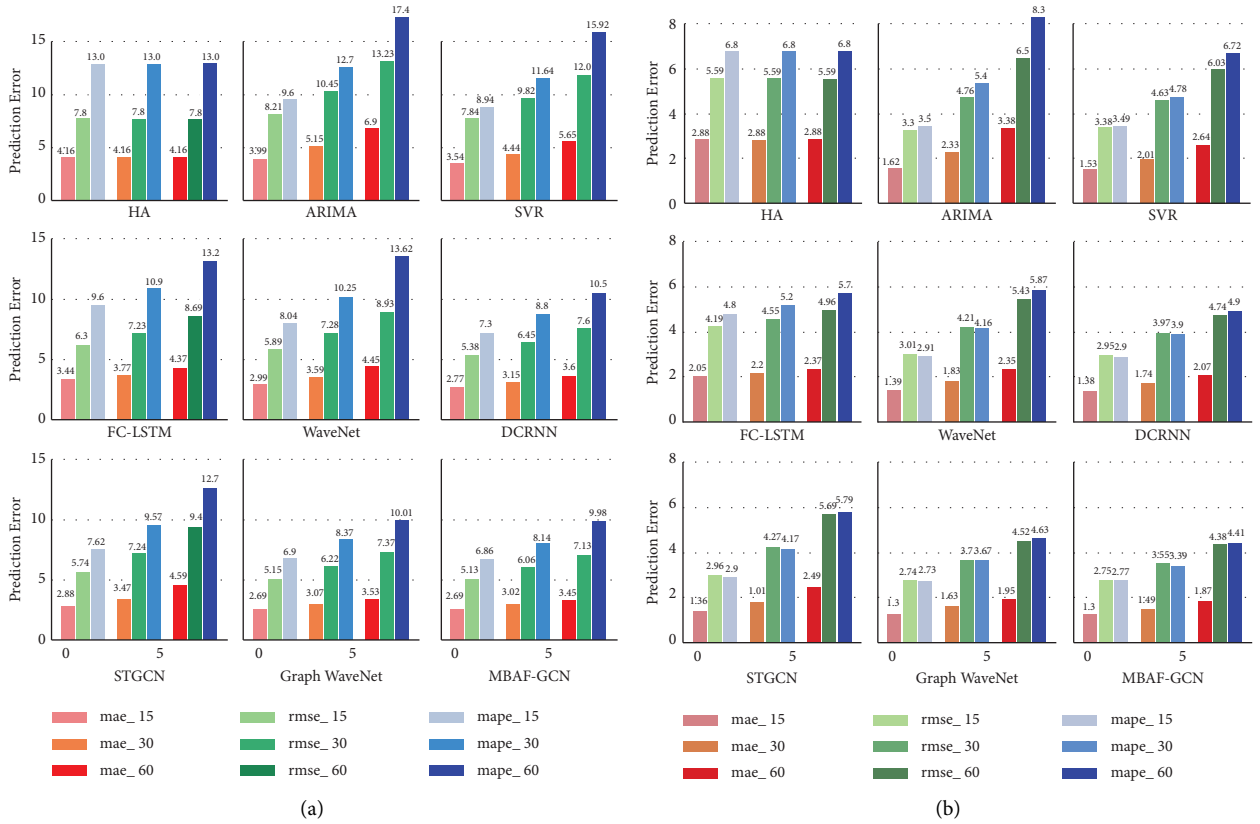


Figure 7: Visualize traffic time series forecast results at 15 min, 30 min, and 60 min on (a) METR-LA and (b) PEMS-BAY.

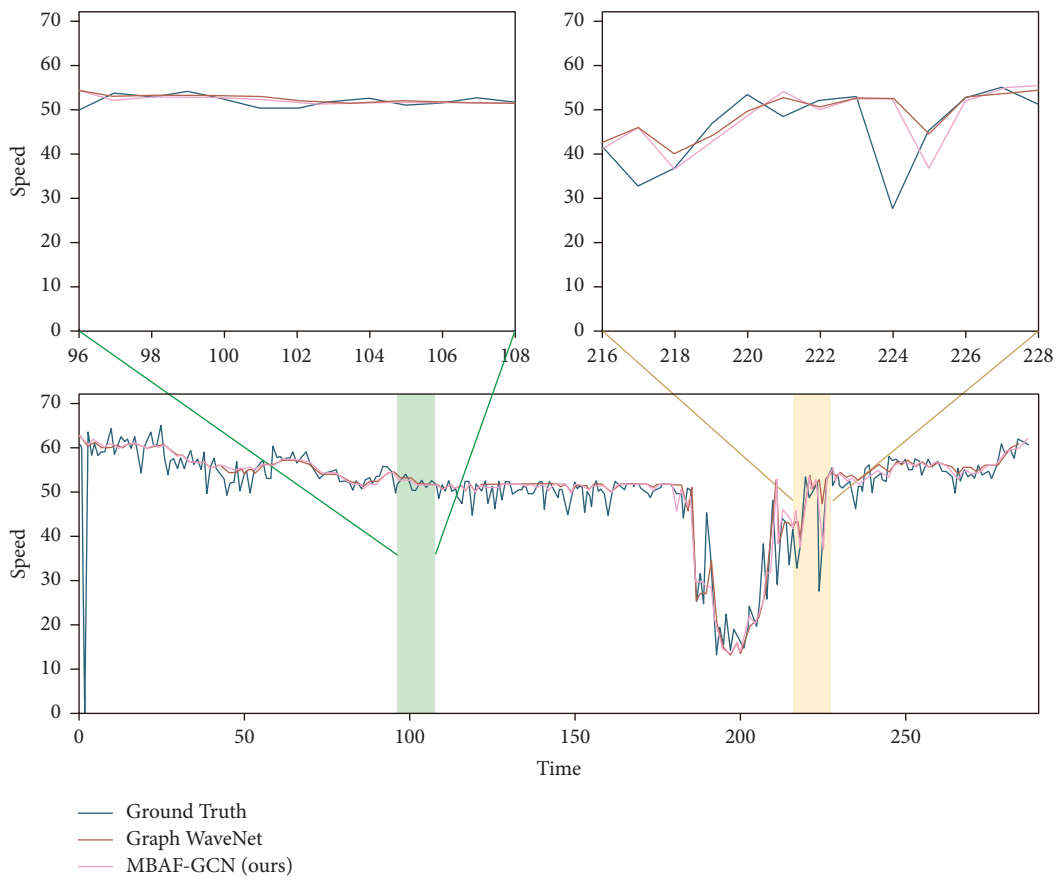


Figure 8: Traffic flow prediction in randomly sampled road sections in METR-LA.

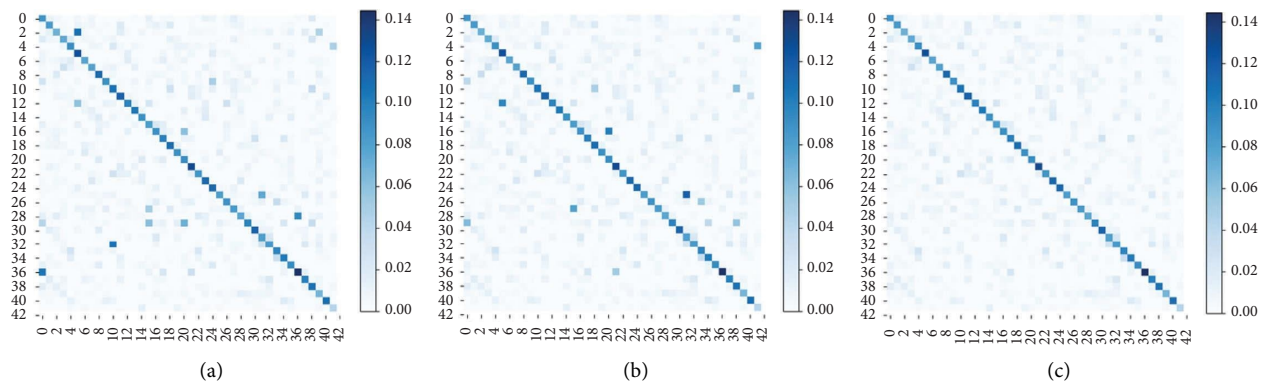
Table 1: Compares the performance of MBAF with other baseline models at 15 min, 30 min, and 60 min.

Data	Models	15 min			30 min			60 min		
		MAE	RMSE	MAPE (%)	MAE	RMSE	MAPE (%)	MAE	RMSE	MAPE (%)
METR-LA	HA	4.16	7.80	13.0	4.16	7.80	13.0	4.16	7.80	13.0
	ARIMA	3.99	8.21	9.60	5.15	10.45	12.70	6.90	13.23	17.40
	SVR	3.54	7.84	8.94	4.44	9.82	11.64	5.65	12.07	15.92
	FC-LSTM	3.44	6.30	9.60	3.77	7.23	10.90	4.37	8.69	13.20
	WaveNet	2.99	5.89	8.04	3.59	7.28	10.25	4.45	8.93	13.62
	DCRNN	2.77	5.38	7.30	3.15	6.45	8.80	3.60	7.60	10.50
	GGRU	2.71	5.24	6.99	3.12	6.36	8.56	3.64	7.65	10.62
	STGCN	2.88	5.74	7.62	3.47	7.24	9.57	4.59	9.40	12.70
	Graph WaveNet	2.69	5.15	6.90	3.07	6.22	8.37	3.53	7.37	10.01
	MBAF-GCN	**2.69**	**5.13**	**6.86**	**3.02**	**6.06**	**8.14**	**3.45**	**7.13**	**9.98**
PEMS-BAY	HA	2.88	5.59	6.80	2.88	5.59	6.80	2.88	5.59	6.80
	ARIMA	1.62	3.30	3.50	2.33	4.76	5.40	3.38	6.50	8.30
	SVR	1.53	3.38	3.49	2.01	4.63	4.78	2.64	6.03	6.72
	FC-LSTM	2.05	4.19	4.80	2.20	4.55	5.20	2.37	4.96	5.70
	WaveNet	1.39	3.01	2.91	1.83	4.21	4.16	2.35	5.43	5.87
	DCRNN	1.38	2.95	2.90	1.74	3.97	3.90	2.07	4.74	4.90
	GGRU	—	—	—	—	—	—	—	—	—
	STGCN	1.36	2.96	2.90	1.81	4.27	4.17	2.49	5.69	5.79
	Graph WaveNet	1.30	**2.74**	**2.73**	1.63	3.70	3.67	1.95	4.52	4.63
	MBAF-GCN	**1.30**	2.75	2.77	**1.49**	**3.55**	**3.39**	**1.87**	**4.38**	**4.41**

The MBAF-GCN model is superior to the comparison model.

Table 2: The parallel multiscale structure results.

Dataset	Model name	Mean MAE	Mean RMSE	Mean MAPE (%)
METR-LR	Single-branch (similar parameters)	3.58	7.18	10.21
	Multibranch (addition fusion)	3.13	6.26	8.65
PEMS-BAY	Single-branch (similar parameters)	1.80	4.05	4.18
	Multibranch (addition fusion)	1.62	3.61	3.72



(a) (b) (c)

Figure 9: The visualization of the global graph attention module in each branch: ((a) tendency branch attention matrix visualization, (b) coarse branch attention matrix visualization, and (c) fine branch attention matrix visualization).

nodes, suggesting its ability to model long-term spatial-temporal dependency. This is critical as predicting the tendency of time series requires the merging of all information, even those located far away. The heatmaps in the coarse branch and fine branch (Figures 9(b) and 9(c)) show more correlations on the diagonal and are sparse in long-term connections due to the smaller receptive field of these branches. As a result, the learned attentions are more focused on local correlations. Notably, although these attention modules are in different branches, they provide scale-specific and complementary features that can be leveraged in subsequent steps.

(2) Effective of the Fusion Module. To validate the effectiveness of our proposed fusion module in fusing spatio-temporal features of different scales and to verify the coarse-to-fine fusion strategy, we conducted a series of ablative experiments. The control group used concatenation and addition to fuse features from different scale branches, without distinguishing features of different scales or using

Table 3: Comparison of experimental results.

Dataset	Model name	Mean MAE	Mean RMSE	Mean MAPE (%)
METR-LR	Concatenation fusion	3.58	7.18	10.21
	Addition fusion	3.13	6.26	8.65
	Coarse-to-fine fusion	3.10	6.21	8.68
PEMS-BAY	Concatenation fusion	1.80	4.05	4.18
	Addition fusion	1.62	3.61	3.72
	Coarse-to-fine fusion	1.61	3.63	3.59

any attention mechanism to adaptively adjust the fusion weights [34]. By comparing the results of these experiments, we can demonstrate the effectiveness of the proposed coarse-to-fine fusion strategy. The experiments were conducted on two datasets.

The experiments show that the multiscale fusion strategy is significantly better than the addition and concatenation feature combination approach, as shown in Table 3. The concatenation or addition fusion strategy merges all branch features with the same weights and ignores the intrinsic correlation of temporal features at different scales. While the coarse-to-fine mechanism can enhance and recalibrate the temporal trend and refine the detailed prediction of the time series.

5. Conclusion

The study proposes the MBAF-GCN model as a novel approach for traffic forecasting. This model employs a multibranch structure with a coarse-to-fine fusion design, offering advantages over comparative models. Firstly, unlike traditional single-branch network designs, the proposed model leverages prior knowledge of spatial-temporal characteristics across different temporal scales to estimate traffic conditions in real-time, capturing both temporal patterns and complex spatial dependencies. Secondly, each branch in the multibranch framework has its loss supervision, which facilitates the learning process and enhances prediction accuracy.

Our study conducted extensive comparison experiments on two real data sets to evaluate the performance of the MBAF-GCN model. Our results demonstrate the following:

(1) The MBAF-GCN model outperforms the traditional single-branch prediction structure in terms of accuracy. In particular, it shows significant improvements over the Graph WaveNet model in predicting values 30 and 60 minutes ahead.

(2) Our study provides novel insights into the use of multibranch complementary temporal features in graph convolutional networks and the fusion of spatial-temporal features from coarse to fine. The MBAF-GCN model achieves competitive results compared to other models based on actual data and is capable of continuously correcting the predicted traffic trends.

In conclusion, while the MBAF-GCN model has demonstrated high prediction accuracy and validity, it is still subject to the influence of external factors that affect traffic conditions in the real world, such as weather changes, social events, and air conditions. In future research, we plan to investigate how to incorporate these external factors into the model in a reasonable way to improve the realistic prediction accuracy of the multibranch network design.

Data Availability

The data used to support the findings of this study are available at https://gitee.com/zhouchena1/MTGNN/.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors acknowledge the funding support from project 04SBS000097C120 at Nanyang Technological University, Singapore.

References

[1] A. Hofleitner, R. Herring, and A. Bayen, "Arterial travel time forecast with streaming data: a hybrid approach of flow modeling and machine learning," *Transportation Research Part B: Methodological*, vol. 46, no. 9, pp. 1097–1122, 2012.

[2] Y. Yuan, Z. Zhang, X. T. Yang, and S. Zhe, "Macroscopic traffic flow modeling with physics regularized Gaussian process: a new insight into machine learning applications in transportation," *Transportation Research Part B: Methodological*, vol. 146, pp. 88–110, 2021.

[3] J. C. Herrera and A. M. Bayen, "Incorporation of Lagrangian measurements in freeway traffic state estimation," *Transportation Research Part B: Methodological*, vol. 44, no. 4, pp. 460–481, 2010.

[4] Y. Yang, Z. Feng, M. Song, and X. Wang, "Factorizable graph convolutional networks," *Advances in Neural Information Processing Systems*, vol. 33, pp. 20286–20296, 2020.

[5] J. Tang, J. Liang, F. Liu, J. Hao, and Y. Wang, "Multi-community passenger demand prediction at region level based on spatio-temporal graph convolutional network," *Transportation Research Part C: Emerging Technologies*, vol. 124, Article ID 102951, 2021.

[6] S. Wan, C. Gong, P. Zhong, B. Du, L. Zhang, and J. Yang, "Multiscale dynamic graph convolutional network for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 58, no. 5, pp. 3162–3177, 2020.

[7] Y. Liu, Y. Liu, and C. Yang, "Modulation recognition with graph convolutional network," *IEEE Wireless Communications Letters*, vol. 9, no. 5, pp. 624–627, 2020.

[8] M. Lv, Z. Hong, L. Chen, T. Chen, T. Zhu, and S. Ji, "Temporal multi-graph convolutional network for traffic flow prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 6, pp. 3337–3348, 2021.

[9] Y. Qin, H. Luo, F. Zhao, C. Wang, and Y. Fang, "NDGCN: network in network, dilate convolution and graph convolutional networks based transportation mode recognition," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 3, pp. 2138–2152, 2021.

[10] B. Yu, Y. Lee, and K. Sohn, "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*, vol. 114, pp. 189–204, 2020.

[11] X. Song, J. Li, Y. Tang, T. Zhao, Y. Chen, and Z. Guan, "Jkt: a joint graph convolutional network based deep knowledge tracing," *Information Sciences*, vol. 580, pp. 510–523, 2021.

[12] J. Zhu, Q. Wang, C. Tao, H. Deng, L. Zhao, and H. Li, "AST-GCN: attribute-augmented spatiotemporal graph convolutional network for traffic forecasting," *IEEE Access*, vol. 9, pp. 35973–35983, 2021.

[13] B. Chen, Z. Zhang, Y. Lu, F. Chen, G. Lu, and D. Zhang, "Semantic-interactive graph convolutional network for multilabel image recognition," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 52, no. 8, pp. 4887–4899, 2022.

[14] H. W. Wang, Z. R. Peng, D. Wang et al., "Evaluation and prediction of transportation resilience under extreme weather events: a diffusion graph convolutional approach," *Transportation Research Part C: Emerging Technologies*, vol. 115, Article ID 102619, 2020.

[15] T. S. Jepsen, C. S. Jensen, and T. D. Nielsen, "Relational fusion networks: graph convolutional networks for road networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 1, pp. 418–429, 2022.

[16] J. Bai, J. Zhu, Y. Song et al., "A3T-GCN: attention temporal graph convolutional network for traffic forecasting," *ISPRS International Journal of Geo-Information*, vol. 10, no. 7, p. 485, 2021.

[17] K. Lee and W. Rhee, "DDP-GCN: multi-graph convolutional network for spatiotemporal traffic forecasting," *Transportation Research Part C: Emerging Technologies*, vol. 134, Article ID 103466, 2022.

[18] L. Lin, Z. He, and S. Peeta, "Predicting station-level hourly demand in a large-scale bike-sharing network: a graph convolutional neural network approach," *Transportation Research Part C: Emerging Technologies*, vol. 97, pp. 258–276, 2018.

[19] S. Guo, Y. Lin, H. Wan, X. Li, and G. Cong, "Learning dynamics and heterogeneity of spatial-temporal graph data for traffic forecasting," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 11, pp. 5415–5428, 2022.

[20] V. N. Ioannidis, A. G. Marques, and G. B. Giannakis, "Tensor graph convolutional networks for multi-relational and robust learning," *IEEE Transactions on Signal Processing*, vol. 68, pp. 6535–6546, 2020.

[21] J. Ke, X. Qin, H. Yang, Z. Zheng, Z. Zhu, and J. Ye, "Predicting origin-destination ride-sourcing demand with a spatio-temporal encoder-decoder residual multi-graph convolutional network," *Transportation Research Part C: Emerging Technologies*, vol. 122, Article ID 102858, 2021.

[22] S. Jeon and B. Hong, "Monte Carlo simulation-based traffic speed forecasting using historical big data," *Future Generation Computer Systems*, vol. 65, pp. 182–195, 2016.

[23] C. Zheng, X. Fan, C. Wang, and J. Qi, "GMAN: a graph multi-attention network for traffic prediction," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 01, pp. 1234–1241, 2020.

[24] L. Huang, F. Mao, K. Zhang, and Z. Li, "Spatial-temporal convolutional transformer network for multivariate time series forecasting," *Sensors*, vol. 22, no. 3, p. 841, 2022.

[25] H. Zhang, Y. Zou, X. Yang, and H. Yang, "A temporal fusion transformer for short-term freeway traffic speed multistep prediction," *Neurocomputing*, vol. 500, pp. 329–340, 2022.

[26] J. Zhang, F. Chen, Y. Guo, and X. Li, "Multi-graph convolutional network for short-term passenger flow forecasting in urban rail transit," *IET Intelligent Transport Systems*, vol. 14, no. 10, pp. 1210–1217, 2020.

[27] R. K. C. Chan, J. M. Y. Lim, and R. Parthiban, "A neural network approach for traffic prediction and routing with missing data imputation for intelligent transportation system," *Expert Systems with Applications*, vol. 171, Article ID 114573, 2021.

[28] Y. Xie, P. Zhang, and Y. Chen, "A fuzzy ARIMA correction model for transport volume forecast," *Mathematical Problems in Engineering*, vol. 2021, Article ID 6655102, 10 pages, 2021.

[29] Z. Lv, Y. Li, H. Feng, and H. Lv, "Deep learning for security in digital twins of cooperative intelligent transportation systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 9, pp. 16666–16675, 2022.

[30] W. Chen, Z. Li, C. Liu, and Y. Ai, "A deep learning model with conv-LSTM networks for subway passenger congestion delay prediction," *Journal of Advanced Transportation*, 2021.

[31] C. Tian and W. K. Chan, "Spatial-temporal attention wavenet: a deep learning framework for traffic prediction considering spatial-temporal dependencies," *IET Intelligent Transport Systems*, vol. 15, no. 4, pp. 549–561, 2021.

[32] F. Hou, Y. Zhang, X. Fu, L. Jiao, and W. Zheng, "The Prediction of Multistep Traffic Flow Based on AST-GCN-LSTM," *Journal of Advanced Transportation*, vol. 2021, Article ID 9513170, 10 pages, 2021.

[33] Y. Li, P. Wang, and C. Y. Chan, "RESTEP into the future: relational spatio-temporal learning for multi-person action forecasting," *IEEE Transactions on Multimedia*, vol. 1, 2021.

[34] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, "Graph Wavenet for Deep Spatial-Temporal Graph Modeling," 2019, https://arxiv.org/abs/1906.00121.