Neural Processing Letters (2024) 56:9
https://doi.org/10.1007/s11063-024-11479-2

Gated Fusion Adaptive Graph Neural Network for Urban Road Traffic Flow Prediction

Liyan Xiong[1] · Xinhua Yuan[1] · Zhuyi Hu[1] · Xiaohui Huang[1] · Peng Huang[2]

Accepted: 24 November 2023 / Published online: 6 February 2024

Abstract

Accurate prediction of traffic flow plays an important role in maintaining traffic order and traffic safety, which is a key task in the application of intelligent transportation systems (ITS). However, the urban road network has complex dynamic spatial correlation and nonlinear temporal correlation, and achieving accurate traffic flow prediction is a highly challenging task. Traditional methods use sensors deployed on roads to construct the spatial structure of the road network and capture spatial information by graph convolution. However, they ignore that the spatial correlation between nodes is dynamically changing, and using a fixed adjacency matrix cannot reflect the real road spatial structure. To overcome these limitations, this paper proposes a new spatial-temporal deep learning model: gated fusion adaptive graph neural network (GFAGNN). GFAGNN first extracts long-term dependencies on raw data through stacking expansion causal convolution, Then the spatial features of the dynamics are learned by adaptive graph attention network and adaptive graph convolutional network respectively, Finally the fused information is passed through a lightweight channel attention to extract temporal features. The experimental results on two public data sets show that our model can effectively capture the spatiotemporal correlation in traffic flow prediction. Compared with GWNET-conv model on METR-LA dataset, the three indexes in the 60-minute task prediction improved by 2.27%,2.06% and 2.13%, respectively.

✉ Xinhua Yuan
yxh1013@foxmail.com

Liyan Xiong
445935939@qq.com

Zhuyi Hu
1067049918@qq.com

Xiaohui Huang
hxh016@gmail.com

Peng Huang
365702240@qq.com

[1] Department of Information Engineering, East China Jiaotong University, Nanchang 330013, Jiangxi, China

[2] Jiangxi Provincial Communications Investment Group Co., Ltd. Road Network Operation Management Company, Nanchang 330000, Jiangxi, China

Keywords Traffic flow prediction · Adjacency matrix · Adaptive graph convolutional network · Adaptive graph attention network

1 Introduction

As the number of vehicles on urban roads increases, traffic management and traffic safety [1] become more and more important. The proposed intelligent transportation system is beneficial to solve this series of problems, and traffic flow prediction [2, 3] is one of its tasks. Traffic flow prediction can predict future traffic conditions in urban road networks based on historical traffic information [4], and timely dispatch vehicles based on the predicted information to avoid traffic jams and improve the operational efficiency of traffic networks.

In recent years, deep learning models have been widely used in traffic flow prediction. The initial approaches modeled road networks as uniformly sized grid structures and then captured spatial correlations using convolutional neural networks (CNN) [5], however, they ignored the irregularity of roads and inevitably lost the topological information in the traffic network. To solve this problem, it has been proposed to construct adjacency matrices using sensors in the road network and assigned weights to the matrices by the distance between sensors, use the constructed adjacency matrices to model the spatial topology of the road network, and finally capture the non-Euclidean spatial correlation of traffic flows by graph neural networks (GNN) [6]. However, these models assumed that the spatial dependence between roads are fixed and do not consider the dynamically changing traffic states, so some models used multi-head graph attention (GAT) [7] to model the spatial dependence. Graph convolution and graph attention are highly dependent on the adjacency matrix, but sometimes the fixed adjacency matrix cannot contain the true spatial dependencies, and distant nodes may reflect similar traffic flows. For temporal dependence, many models used recurrent neural networks (RNN) [8] for temporal modeling, but its limitations were also very obvious, and its chain structure designed strictly follows temporal development, making it easy to lose long-term dependence information. The temporal attention-based model [9] provides direct access to long-term dependent information, it has the problems of slow training and easy to ignores the spatial correlation of data.

Although the above methods can solve some of the problems in traffic flow prediction, they fail to fully consider the dynamic spatial and temporal correlations [10]. As Fig. 1a shows the distribution of sensors in the traffic network. Over time, we can get the change of traffic flow correlation between sensors 1 to 3. As shown in Fig. 1b, sensor 1 exhibits different spatial-temporal correlations with other sensors at different times. Sensors 2 and 3 have a high dependency at the initial moment, but as time increases, their dependency becomes weaker and weaker, and instead the dependency with sensor 1, which is farther away increases. Therefore capturing these complex spatial-temporal dependencies is often the key to reducing prediction errors if only the adjacency matrix cannot represent their spatial correlation.

The majority of urban data is spatio-temporal, representing that it pertains not only to spatial locations but also changes over time. Firstly, we consider the geographical location of nodes, taking road intersections as nodes, road connection lines as edges, and the whole road network as spatial graph structure. Because of the periodicity of traffic information, the historical traffic flow has a time correlation with the traffic flow in the next time period. Taking these spatio-temporal features into account will lead to better learning of prediction tasks.

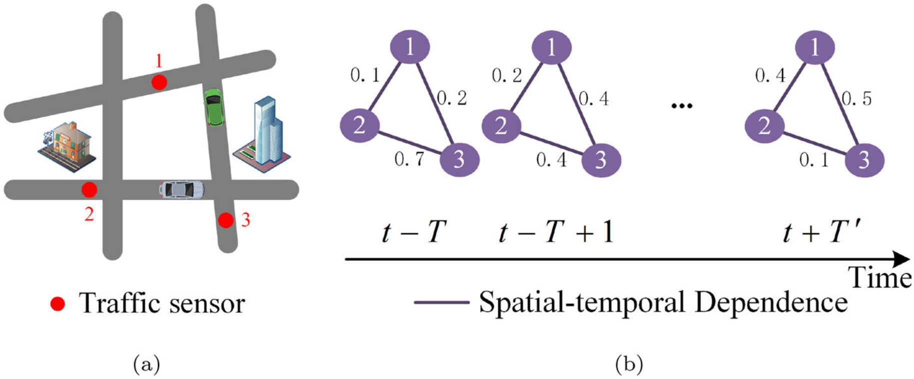


Fig. 1 Spatial-temporal correlation is dominated by the road network structure. **a** Traffic sensors distributed in the road network. **b** Dynamic spatial-temporal dependence form time $t-T$ to time $t+T'$

Based on the above spatial-temporal dependencies, we propose a new deep learning model, the gated fusion adaptive graph neural network (GFAGNN) for traffic flow prediction, which can adaptively capture the dynamic spatial-temporal dependence information of road networks and fuse the long-term and short-term spatial-temporal hidden information extracted by adaptive graph convolution and adaptive graph attention through gated units. We have evaluated GFAGNN on two public datasets (METR-LA, PEMS-BAY) and achieved satisfactory results. In summary, the main contributions of our work are as follows:

- We design a temporal framework based on gated temporal convolution and channel attention mechanism. The global dependencies are first extracted by gated temporal convolution, which consists of two parallel dilated causal convolutions, and multiple temporal convolution layers can be superimposed to process the information of each sensory field in different layers. Finally the features obtained are fused and adjusted by the channel attention mechanism.
- We design a GFA block, consisting of adaptive graph attention and adaptive graph convolution. which uses self-learning node embedding to learn potential spatial relationships instead of relying only on the adjacency matrix to model spatial dependencies. Also, a gating fusion mechanism is proposed to control the output.
- This paper compares the prediction results of the proposed model with the results of some models proposed in recent years, and the experimental results show that the performance of our proposed model is improved.

The remainder of this paper is as follows, Sect. 2 presents work related to traffic flow forecasting, Sect. 3 presents preparatory work and problem definition, Sect. 4 we detail the gated fusion adaptive graph convolution model framework, we present extensive performance comparison experiments and visualization of forecast data in Sects. 5 and 6, and perform ablation experiments to demonstrate the usefulness of each module, and finally, we conclude our work in Sect. 7.

2 Related Work

2.1 Traffic Flow Forecasting

Traffic flow forecasting has been a popular direction in deep learning, and various emerging models have been proposed to simulate traffic characteristics in recent decades with many results. The historical average (HA) and autoregressive integrated moving average model (ARIMA) [11] are representative statistical models for traffic forecasting. Kumar et al. [12] proposed a seasonal ARIMA (SARIMA) based traffic flow forecasting model that plots the autocorrelation function (ACF) and partial autocorrelation function after the model performs the necessary differentiation to stabilize the input time series. These methods consider temporal correlation and can only deal with simple linear relationships, lacking nonlinear modeling capabilities, leading to difficulties in achieving better results. To solve the above problems, a large number of machine learning methods have been applied to traffic flow prediction. Wang et al. [13] used artificial neural networks and Kalman filtering to predict short-term passenger flow in subway stations, and experiments showed that the Kalman filtering approach could effectively reduce errors. Sun et al. [14] proposed a hybrid model based on wavelet transform and support vector machine(SVM), which combined the advantages of both models to fit passenger flow information and achieved better results. Guo et al. [15] developed a feature extraction model and used the K means method to classify stations into different types, and then a hybrid model based on kernel ridge regression and Gaussian process regression was used to predict short-term passenger flow in urban transportation and validated on automatic ticketing system data. However, the above traditional machine learning methods rely heavily on manual data processing, rely only on historical temporal information, ignore dynamic spatial relationships [10], and are not suitable for application in complex road network structures.

2.2 Spatial-Temporal Prediction Based on Deep Learning

With the success of deep learning in directions such as natural language processing and image processing, more and more deep learning models are being applied in the direction of traffic flow prediction in road traffic networks. Through a large number of models and experiments, it is proved that using deep learning to capture the temporal and spatial information hidden in the road traffic network is both stable and effective.

Correlation time series prediction: Historical traffic flows play an important role in predicting future traffic flow efforts, and most such studies rely on recurrent neural networks (RNN). To solve the problems of the inability of long-term memory and gradient disappearance in backpropagation in RNN, Ma et al. [16] proposed to use long short-term memory (LSTM) neural networks to capture nonlinear dynamic temporal correlations. The gate recurrent unit (GRU) [17] and LSTM function similarly, but the GRU has fewer parameters and converges faster. While previous sequence modeling was mainly related to recurrent neural network architectures, Yu et al. [18] argue that convolutional networks achieve better results because they allow parallel computation of outputs, and their inclusion of temporal convolutional networks (TCN) in the model improves experimental efficiency, enabling very long sequences to be processed in less time. However, these studies did not explicitly consider the interdependencies between different time series, and recently transformers [19] have been used for correlated time series prediction, a type of work that usually requires training a large number of parameters and cannot be effective with insufficient training samples.

Graph neural networks: Since urban road networks present irregular network structures, traditional convolutional neural networks cannot accurately capture the spatial-temporal correlation of individual nodes. Therefore, a hybrid model based on graph neural networks (GNN) and recurrent neural networks (RNN) are proposed in the field of traffic flow prediction. GNN can directly handle more general graphs, including recurrent, directed and undirected graphs, and play an important role in dealing with spatial structure dependence. Han et al. [20] proposed a spatial-temporal graph convolutional neural network, instead of using a grid to represent regions, they converted the urban road network into an adjacency matrix and used graph convolution networks (GCN) to capture spatial-temporal correlations. T-GCN [21] combines GCN and GRU to aggregate spatial-temporal information, and AGCRN [22] is stacked multiple times and then used as an encoder to capture the spatial-temporal dependencies of road nodes. The above methods extract temporal and spatial information step by step without achieving simultaneous capture of spatial-temporal correlations, so STGCN and STSGCN [23, 24]were proposed for simultaneous capture of spatial-temporal correlations. To further propose a more suitable graph convolution for directed graphs, Li et al. [25] proposed the diffusion recurrent neural network (DCRNN), which uses bidirectional random wandering on the graph to capture spatial correlations, But these adjacency matrices are static and depend on a predefined graph structure. Graph WaveNet [26] also employs diffusion convolution in spatial modeling, but it differs from DCRNN in that it considers connected and unconnected nodes in the modeling process and uses the adaptive adjacency matrix to reconcile the information between nodes. The attention mechanism is used in various fields due to its efficiency and flexibility, it can automatically focus on important information based on historical input data, and GAT is used for traffic flow prediction to build spatial correlation models. To achieve better results, Van et al. [27] proposed a talking head mechanism by adding linear projections to the multi-headed attention mechanism. GMAN [28] is designed to learn attention scores by considering traffic features and node embeddings from the graph structure for spatial attention mechanism. However, because these models use too many attention mechanisms, they require high computational costs.

In recent years, the spatial-temporal graph neural network focuses on spatial learning methods, temporal learning methods, spatial-temporal fusion methods and other advanced technologies that can be combined. [29] Most studies are aimed at proposing new models for these problems. Huang et al. [30] Liu et al. [31] proposed a new component of spatial-temporal adaptive embedding to solve the problem of diminishing performance returns encountered in spatio-temporal traffic modes. Li et al. [32] believe that the dynamic correlation between locations in the network is crucial to the prediction task, and in addition, the fair comparison between different methods is lacking, so he designed a generative method to model the fine topology of the dynamic graph at each time step. In order to make the effectiveness of the model not overly dependent on the quality of the structure of the spatial topological graph. Lin et al. [33] captured the fine spatial-temporal topology of the traffic data by embedding a time-varying Bayesian network, and then generated a step-by-step dynamic causal graph through deep learning methods. Shao et al. [34] believed that previous work treated traffic information roughly as the result of diffusion while ignoring the inherent signal, which would have a negative impact. To address this problem, they proposed an decoupage spatio-temporal framework, which separated the inherent traffic information of diffusion in a data-driven way, and processed the separated signals separately to capture the spatial-temporal correlation. Yang et al. [35] proposed the STFAGN model to obtain incomplete spatiotemporal connection information. They first extracted spatial information by combining fusion convolution layer with the adaptive dependency matrix, then introduced gated CNN to extract time information, and finally replaced residual connection with ReZero connection to achieve faster convergence, however, the network model cannot capture the dynamic spatial relationships hidden in the traffic dataset.

The recent work described above successfully addressed certain issues, but also revealed some limitations. These models rely on the preparation work and predetermined adjacency matrix during the construction of spatial topology diagrams, making it challenging to accurately represent complex spatial information in road networks solely through static spatial matrix data. To overcome this limitation, we propose a network structure that integrates adaptive graph convolution with adaptive graph attention. By incorporating adaptive nodes into the graph structure, we can effectively capture hidden spatial structures from historical data. Furthermore, to enhance the model's long-term prediction capability, we introduce an extended causal convolution and channel attention mechanism to capture temporal correlations.

3 Preliminary

Traffic flow forecasting is the prediction of traffic information for future periods based on historical traffic information on the road. In this section, we first give some key definitions and then formally formulate the forecasting problem.

Definition 1 (traffic network graph G). As shown in Fig. 1a, in a realistic traffic road network, the closer the road nodes are, the more similar the traffic flow is, so we define a weighted graph $G = (V, E, A)$. where is a set of N road nodes (representing the sensors in the traffic road network) and E is a set of edges connecting these road nodes (representing the connection weights between nodes). The adjacency matrix $A \in R^{N \times N}$ represents the connection relationship between road nodes, where N represents the number of road nodes, A_{ij} represents the edge weights of node i and node j. For example, for any two nodes v_i and v_j, the values of A_{ij} and A_{ji} set 1 if the two nodes are connected, and the weight of the two elements is set to 0 if they are not connected.

Definition 2 (Traffic Flow Records X). We define $X_i^t \in R^C$ as the traffic flow at the node i at moment t, where C is the number of traffic conditions of interes(traffic speed). In this work, we aim to predict only one parameter, the speed of traffic for all vehicles(hence $C = 1$). $X^t = \left[X_1^t, X_2^t, \cdots, X_N^t\right] \in R^{N \times C}$ indicates all node information, The same $X \in R^{N \times \mathrm{C} \times T}$ represents the traffic information of all nodes at any moment.

Definition 3 (Problem Definition). The traffic flow forecasting task learns a function $f(\cdot)$, which capable of mapping historical T period observations to future T' moment traffic information using a traffic network topology graph G and historical traffic information X. In this work, we predict information about 15, 30, and 60 min into the future. The computational procedure is as follows:

$$[X^{t-T+1}, \cdots, X^t; G] \xrightarrow{f(\cdot)} \left[X^{t+1}, \cdots, X^{t+T'}\right] \in R^{N \times C \times T'}, \tag{1}$$

where $X^{(t-T+1:t)}$ represents the historical traffic information and $X^{(t+1:t+T')}$ represents the predicted traffic values at future moments.

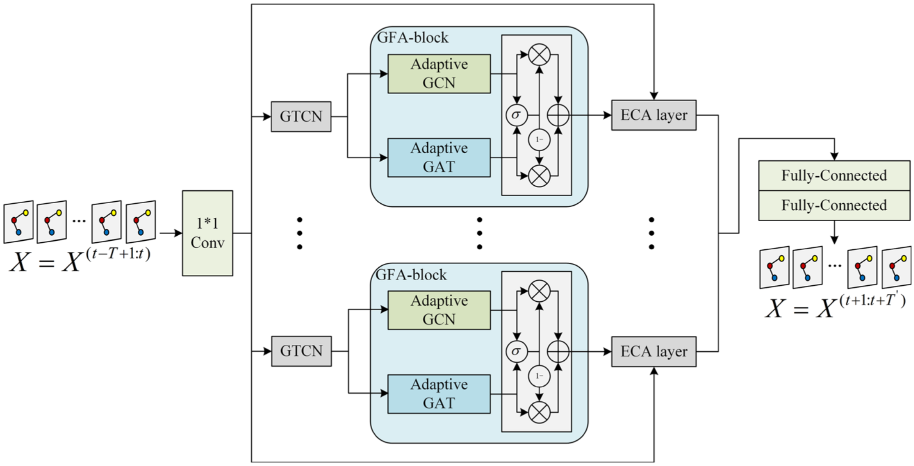


Fig. 2 The general framework of GFAGNN, which consists of an input layer, L layers GFA blocks, and an output layer

4 Framework of the GFAGNN

The general framework of the proposed GFAGNN is shown in Fig. 2, GFAGNN consists of three main components: GTCN, Adaptive GCN, and Adaptive GAT. Specifically, we stacked L layers GFA blocks, first extracted global temporal dependencies using Gate TCN, and then used Adaptive GCN and Adaptive GAT to model the spatial correlation of the traffic road network, and self-adaptive fusion of dynamic spatial–temporal correlation information by the Gate TCN. In addition, residual information is added to avoid network degradation, after lightweight channel attention to further improve the model performance. Finally, two fully connected layers is used to predict the final results.

4.1 Gated Temporal Convolution

To solve the problems such as gradient explosion and the inability of parallel computation in RNN models, we use gated temporal convolution (GTCN) to capture the dynamic temporal information in the road network. As shown in Fig. 3a, GTCN contains two convolutional operations, and each selectively retains important information through different activation functions. The convolution operation has the advantages of simple structure and stable gradient, while the dilation causal convolution can also obtain an exponential field of view as the dilation depth increases [26]. To ensure that only historical information is used to predict the traffic flow at the current moment, the temporal causal order can be maintained by padding the input sequence with zeros. Figure 3b shows an extended causal convolution with expansion factors of 1, 2, 4. Where the filter is applied to a long sequence by skipping the input value at a certain step size, we set each layer to expand the jump step size exponentially by 2, so we can express the expansion factor d for layer l^{th} as $2^{(l-1)}$, which can easily capture the dependencies of long time series as the depth increases. The gated temporal convolution equation is as follows:

$$X^{(l)} = \tanh\left(W_1 * X^{(l-1)} + c_1\right) \odot \sigma\left(W_2 * X^{(l-1)} + c_2\right), \tag{2}$$

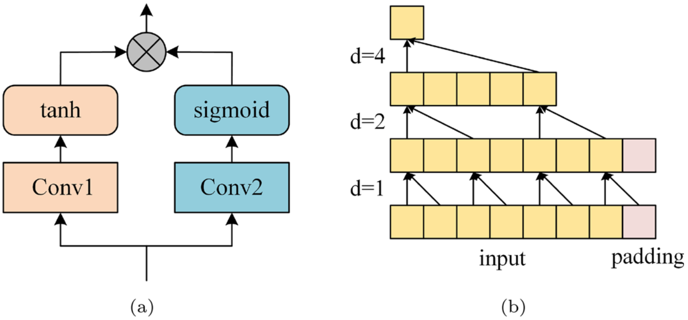


Fig. 3 GTCN and some details: **a** The framework of Gate temporal convolution. **b** Dilated causal convolution with kernel size 2

where $X^{(l-1)} \in R^{N\times C_{l-1}\times T_{l-1}}$ represents the output of $(l-1)^{th}$ as the input of l^{th}, $X^{(l)} \in R^{N\times C_l \times T_1}$ represents output of l^{th}, W_1, W_2, c_1, c_2 are learnable parameters, which are assigned by random initialization and constantly updated during model training, $tanh$ and σ are activation functions, which can determine the output of important information in the next layer, $*$ represents the convolution operation, and $\odot$ is the elements-wise product.

4.2 Adaptive Graph Convolution

For irregular topologies, graph convolution networks can act directly on the graph instead of convolutional neural networks to extract the spatial features of the topological graph. The graph convolution module aims to fuse a node's information with its neighbors' information to handle spatial dependencies in a graph There are mainly spectral methods and spatial methods to implement graph convolution [6]. The spectral domain graph convolution has problems such as large computational effort, the graph structure cannot be changed, and it is not suitable for extracting spatial features on directed graphs. Therefore, in this paper, we utilize a diffusion graph convolution base on the spatial domain. First, we simulate the diffusion process of the graphical signal with K finite steps and use the diffusion convolution [25] to capture the spatial dependence. From a space-based perspective, it is used to smooth the signals of nodes by aggregating and transforming their neighborhood information. In addition, we design an adaptive matrix to model the hidden spatial information in the road network structure. Combining predefined spatial graph information and adaptive hidden graph structure, the diffusion adaptive graph convolution is written as:

$$X^{(l)}_{\text{agcn}} = \sum_{k=0}^{K-1} A_f^k X^{(l)} W_{k1} + A_b^k X^{(l)} W_{k2} + A_{adp}^k X^{(l)} W_{k3}, \tag{3}$$

where $X^{(l)}_{agcn}, X^{(l)} \in R^{N\times C_l\times T_l}$ is the output and input of the adaptive graph convolution, K is the number of diffusion steps, W_{k1}, W_{k2} and W_{k3} are learnable parameters. A_f, A_b represent the forward and backward feature matrices, is a randomly initialized adaptive adjacency matrix. And they are constructed as follows:

$$A_f = A/rowsum(A), \tag{4}$$

$$A_b = A^T/rowsum(A^T), \tag{5}$$

$$A_{adp} = SoftMax\left(ReLU\left(e_1 e_2^T\right)\right), \tag{6}$$

where $e_1, e_2 \in R^{N \times F}$ represents the source node embedding and the target node embedding, and they are multiplied to obtain an $N \times N$ adaptive adjacency matrix, which adaptively changes the hidden spatial dependence information by stochastic gradient descent for end-to-end learning [36]. F is the embedding dimension, we set it as a hyperparameter, the details of which are shown in the experimental section.Where $rowsum$ represents summation by row, $ReLU$ and $SoftMax$ represent two different activation functions that mainly serve to eliminate weak connections and normalize.

4.3 Adaptive Graph Attention

Neighboring roads have similar traffic flow conditions, but different nodes have different influences on each other. To address the inability of graph convolution to allow assigning different weights to different nodes in the neighborhood, adaptive graph attention [37, 38]is used in the graph structure to model dynamic spatial correlation. The advantage of graph attention is that each node can be assigned different weights to neighboring nodes based on their characteristics. The adaptive graph attention structure is shown in Fig. 4, And so that the model can better learn the hidden traffic states, we connect the adaptive nodes to the hidden states and use the scaled dot product method to calculate the attention. The input is a node feature matrix $X^t \in R^{N \times C_l}$ (where N nodes in the graph and each node has C_l features), and the node embedding $e \in R^{N \times F}$ (F is the embedding dimension) is randomly initialized and trained step by step. The attention coefficients are calculated as follows:

$$s_{ij} = \frac{ReLU\left\langle W_q\left(X_i^t \| e_i\right), W_k\left(X_j^t \| e_j\right)\right\rangle}{\sqrt{d}}, \tag{7}$$

$$\alpha_{ij} = SoftMax_j\left(s_{ij}\right) = \frac{\exp\left(s_{ij}\right)}{\sum_{k \in N_i} \exp\left(s_{ik}\right)}, \tag{8}$$

In the formula, $\|$ represents the concatenation operation, $\langle \cdot, \cdot \rangle$ represents the inner product operation, s_{ij} is the similarity score between node i and node j, W_q and W_k is the query and key learnable parameter matrix, they are initialized randomly and then updated during trainingand, d is the dimension of the key and value. After calculating the attention score, the s_{ij} is normalized using the $softmax$ function, representing all neighbors of the node i.

$$X_i^{(l)} = \sigma\left(\sum_{j \in N_i} \alpha_{ij} X_j^{(l)}\right), \tag{9}$$

The key idea of attention is to dynamically assign different weights to different nodes. Where $X_i^{(l)} \in R^{C_l \times T_l}$ is the weight information representation of the node i. The same $X_{agat}^{(l)} \in R^{N \times C_l \times T_l}$ represents the output of all nodes. In order to stabilize the learning process of self-attention, the residual connections are added to each layer of attention. And the non-linear factors are added through the σ activation function to improve the expressiveness of the model.

4.4 Gated Fusion Module

In order to extract nonlinear dynamic spatial features on road traffic networks, we design two ways to aggregate the information of proximity neighbors, namely adaptive graph convolu-

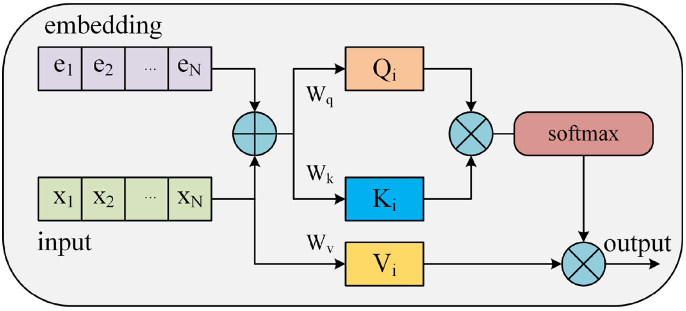


Fig. 4 Adaptive graph attention convolution network

tion and adaptive graph attention, directly splicing these two features will lead to unstable performance, so this paper combines a gated fusion mechanism [39] to construct learning gates for selective learning. With $X_{\text{agcn}}^{(l)}$, $X_{\text{agat}}^{(l)} \in R^{N\times C_l\times T_l}$ representing the output of adaptive graph convolution and adaptive graph attention convolution of l^{th} layer, the gate fusion formula can be expressed as:

$$Z^{(l)} = \sigma(X_{agcn}^{l} W_{Z1} + X_{agat}^{l} W_{Z2} + c), \tag{10}$$

$$X_Z^{(l)} = Z^{(L)} \odot X_{agcn}^{(l)} + (1 - Z^{(L)}) \odot X_{agat}^{(l)} + X^{(l)}, \tag{11}$$

where $W_{Z1}, W_{Z2} \in R^{C_l\times C_l}$ and $c \in R^{C_l}$ are learnable parameters, they are initialized randomly and then updated during trainingand, $\odot$ represents the element-wise product. $Z^{(l)}$ represents the gate and $X_Z^{(l)} \in R^{N\times C_l\times X_l}$ is the output incorporating spatial-temporal correlation from the adaptive graph neural network, which can satisfy both long-term and short-term prediction tasks. In addition, to avoid the problem of network performance degradation as the network depth increases, we add a residual structure that both maintains local states and explores deep neighborhood information.

4.5 ECA Layer

As attention mechanisms are introduced into traffic flow prediction tasks and show great potential for performance improvement, the computational effort increases with higher model accuracy and complexity. Therefore, we introduce a lightweight Efficient Channel Attention Module (ECA) [40], a local cross-channel interaction strategy without dimensionality reduction. The input feature map is first compressed with spatial features, and then the compressed feature map is subjected to channel feature learning, and the learned scores are multiplied with the input features channel by channel to finally output a feature map with channel attention. which can significantly improve the model performance although it involves only a few parameters. Given the output $X_i^t = X_Z^{(l)}[C_l : i : t] \in R^{C_l}$ of the gated fusion module, as the input to the ECA layer. The weights of each channel in ECA are calculated as follows:

$$y = g\left(X_Z^{(l)}\right) = \frac{1}{NT_l}\sum_{i=1}^{N}\sum_{t=1}^{T_l} X_i^t, \tag{12}$$

$$\omega_i = \sigma\left(\sum_{j=1}^{k} W^j y_i^j\right), y_i^j \in \Omega_i^k, \tag{13}$$

Table 1 Details of datasets

Dataset	Sensors	Loads	Samples	Unit	Input length	Output length
METR-LA	207	1515	34272	5min	12	12
PEMS-BAY	325	2369	52116	5min	12	12

$$X_{\text{eca}}^{(l)} = \omega X_Z^{(l)} = \omega \left(X_1, X_2, \cdots, X_{C_l} \right) \in R^{N \times C_l \times T_l}, \tag{14}$$

where $g\left(X_Z^{(l)}\right) \in R^{C_l}$ represents the global average pool (aggregated feature), y_i^j represents the k^{th} neighbor of the i^{th} channel of the aggregate feature. W^j indicates that all channels share the same weight, ω is the set of channel attention weights, where Ω_i^k indicates the set of k adjacent channels of y_i.

$$k = \psi(C_l) = \frac{\log_2(C_l)}{\gamma} + \frac{b}{\gamma}, \tag{15}$$

The size of the convolution kernel in ECA can be adaptively determined based on the ratio between the number of channels C_l and the kernel size $\psi(C_l)$. When the number of channels is large, the required convolution kernel will increase. In order to facilitate subsequent convolution operations, γ and b are set to 2 and 1 respectively. This alteration adjusts the ratio between channel count C_l and convolution kernel size k, enabling effective interaction among each channel.

5 Experiment

In this section, we conduct experiments on two large real-world datasets to demonstrate the effectiveness of GFAGNN in traffic flow prediction. We first introduce the experimental datasets, parameter settings, and evaluation metrics, and then list some traffic prediction models in recent years as a baseline against which the results of GFAGNN are compared in the experiments. In addition, we design some ablation experiments to evaluate the impact of basic structural components and training strategies on the experiments.

5.1 Datasets

To evaluate the performance of GFAGNN, we conducted comparative experiments on two real road traffic datasets (METR-LA and PEMS-BAY) published by Li et al. [25] The raw traffic data were summarized into a 5-minute interval, including two characteristics of vehicle speed and number of vehicles, and only one feature of traffic speed was considered in this study. We divide the dataset into training, validation and test sets in the ratio of 7:1:2 in chronological order, and then process the above segmented data through a sliding window of length $T = 12$ to predict the traffic speed at the next $T' = 12$ time step. Besides, the spatial adjacency graph of each dataset is constructed based on the actual road network. Table 1 shows statistical information about the dataset.

5.2 Experimental Details

The model was implemented by Pytorch 1.10.0 and all experiments were performed on an Nvidia GeForce RTX 3080Ti GPU, in addition, we used the same hyperparameters for METR-LA and PEMS-BAY. To cover the input sequence length, the number of GFA layers is set to 8, where the sequence of expansion factors for each layer in the gated time convolution is set to 1, 2, 1, 2, 1, 2, and the diffusion step $k = 2$. The dimension of the adaptive node embedding F=16. We set the maximum number of iterations to 100, the batch size to 64, and the initial learning rate to 0.001, and use the Adam optimizer optimization is performed, and Dropout of $p = 0.3$ is applied to the output of adaptive graph convolution and adaptive graph attention. To test the prediction performance of the model, we evaluated the true value y and the predicted value y' using the following three metrics.

- Mean Absolute Error (MAE):

$$MAE = \frac{1}{T}\sum_{i=1}^{T}|y_i - y_i'| \tag{16}$$

- Root Mean Squared Error (RMSE):

$$RMSE = \sqrt{\frac{1}{T}\sum_{i=1}^{T}\left(y_i - y_i'\right)^2} \tag{17}$$

- Mean Absolute Percentage Error (MAPE):

$$MAPE = \frac{1}{T}\sum_{i}^{T}|\frac{y_i - y_i'}{y_i}| \tag{18}$$

where T denotes the total number of observed samples, and y_i and y_i' denote the actual and prediction values of the ith sample. MAE is the average absolute error loss, which can reflect the actual situation of the predicted value of traffic flow, a higher MAE indicates lower average prediction accuracy. RMSE is the root mean square error that measures the deviation between the predicted value and the actual traffic. MAPE, which stands for mean Absolute percentage error, is a relative error measure that does not change with the global scaling of the forecast and can be applied to problems with large forecast gaps. The smaller the value of these three metrics, the better the prediction model performance.

5.3 Baselines

We compared GFAGNN with a number of advanced traffic forecasting models in recent years, the baseline models of which are described below.

- DCRNN [25]: Diffusion Convolutional Recurrent Neural Network, which modelled traffic network temporal information with bidirectional GCN with GRU.
- STGCN [23]: Spatial-temporal Graph Convolutional Network, this model used graph convolution to extract spatial correlation and one-dimensional convolution to extract temporal correlation.
- GMAN [28]: GMAN designed an encoder-decoder architecture with spatial, temporal, and transformer attention to capture the spatial-temporal information of traffic flows.

- Graph WaveNet [26]: The model created an adaptive correlation matrix to capture the hidden spatial correlations in the data and combined diffusion map convolution with one-dimensional extended convolution.
- FC-GAGA [41]: Fully connected gated graph architecture, a hard graph gating mechanism for traffic flow prediction is proposed.
- MTGNN [42]: A graph learning module is proposed to construct spatial information, and then the self-learning graph architecture is used for multivariate time series prediction.
- STAWnet [37]: The model captured spatial-temporal correlation by combining temporal convolution with an attention network.
- GWNET-conv [43]: A new loss function (covariance loss) is introduced and applied to Graph WaveNet.

5.4 Experiment Results and Analysis

As shown in Tables 2 and 3, we conducted a prediction comparison experiment over 60 min using GFAGNN and the baseline model from recent years. Notably, GFAGNN achieved advanced performance on all three evaluation metrics in both datasets, in both the long and short term. Among these compared methods, GFAGNN outperforms the spatial-temporal methods (including DCRNN, STGCN), explained by our inclusion of adaptive node embedding in the graph model, which can learn hidden spatial correlations from historical traffic data. The GMAN model is better in long-term prediction due to the enhanced ability to capture long-term information using a large amount of attention but at the cost of costing a long time to train the model and poor short-term prediction. We fuse adaptive graph convolution and adaptive graph attention by gating to improve long-term prediction without degrading short-term prediction. Compared with the best performance in Graph WaveNet, MTGNN, and GWNET-conv, GFAGNN reduces MAE by about 2.27%, RMSE by 2.06%, and MAPE by 2.13% in a 60-minute prediction task on the METR-LA dataset. FC-GAGA and STAWnet rely only on self-learning spatial relationships to predict future traffic flow, and ignoring the information of the neighboring graphs make it difficult for the model to capture local sequence correlations, which reduces the performance of short-term prediction.

We also use the T-test to test the significance of GFAGNN in 60-minute ahead predictions compared to GWNET. The p-value is equal to 1.255e$-$06 and less than 0.05, which demonstrates that GFAGNN statistically outperforms GWNET. In order to demonstrate the trade-off between improved performance and computational complexity of the proposed model, we counted the running time of the model and our model runs faster than DCRNN and GMAN, which is due to the time-consuming sequence learning and attention mechanism in recurrent networks, and STGCN runs the fastest but has poorer predictive performance. It is worth noting that our model has similar speed and better predictive performance than Graph WaveNet and STAWnet.

In summary, the gating unit of GFAGNN fuses both adaptive graph convolution and adaptive graph attention to compensate for their shortcomings, capturing both long- and short-term dynamic spatial and temporal correlations, and demonstrating through data that our proposed model is more effective than these baselines.

5.5 Convergence Analysis

To explore the convergence of the model, we show the decreasing trend of the training loss and validation loss over 100 epochs for both datasets. Figure 5 show the loss convergence curves

Table 2 Experimental results on the METR-LA dataset

Method	15 min			30 min			60 min		
	MAE	RMSE	MAPE(%)	MAE	RMSE	MAPE(%)	MAE	RMSE	MAPE(%)
DCRNN	2.77	5.38	7.30	3.15	6.45	8.80	3.60	7.60	10.50
STGCN	2.88	5.75	7.62	3.47	7.24	9.57	4.59	9.40	12.70
GMAN	2.77	5.48	7.25	3.07	6.34	8.35	**3.40**	7.21	9.72
WaveNet	2.69	5.15	6.90	3.07	6.22	8.37	3.53	7.37	10.01
FC-GAGA	2.75	5.34	7.25	3.10	6.30	8.57	3.51	7.31	10.14
MTGNN	2.69	5.18	6.86	3.05	6.17	8.19	3.49	7.23	9.87
STAWnet	2.70	5.22	6.98	3.04	6.14	8.22	3.44	7.16	9.82
GWNET	2.69	**5.14**	6.83	3.07	6.13	8.26	3.53	7.27	9.85
GFAGNN	**2.69**	5.15	**6.81**	**3.04**	**6.10**	**8.12**	3.45	**7.12**	**9.64**

Table 3 Experimental results on the PEMS-BAY dataset

Method	15 min			30 min			60 min		
	MAE	RMSE	MAPE(%)	MAE	RMSE	MAPE(%)	MAE	RMSE	MAPE(%)
DCRNN	1.38	2.95	2.90	1.74	3.97	3.90	2.07	4.74	4.90
STGCN	1.36	2.96	2.90	1.81	4.27	4.17	2.49	5.69	5.79
GMAN	1.34	2.82	2.81	1.62	3.72	3.63	**1.86**	**4.32**	**4.31**
WaveNet	1.30	2.74	2.73	1.63	3.70	3.67	1.95	4.52	4.63
FC-GAGA	1.36	2.86	2.87	1.68	3.80	3.80	1.97	4.52	4.67
MTGNN	1.32	2.79	2.77	1.65	3.74	3.69	1.94	4.49	4.53
STAWnet	1.31	2.78	2.76	1.62	3.70	3.67	1.89	4.36	4.47
GWNET	1.30	2.73	2.69	1.62	3.67	3.59	1.91	4.40	4.47
GFAGNN	**1.29**	**2.71**	**2.69**	**1.60**	**3.60**	**3.55**	1.89	4.35	4.41

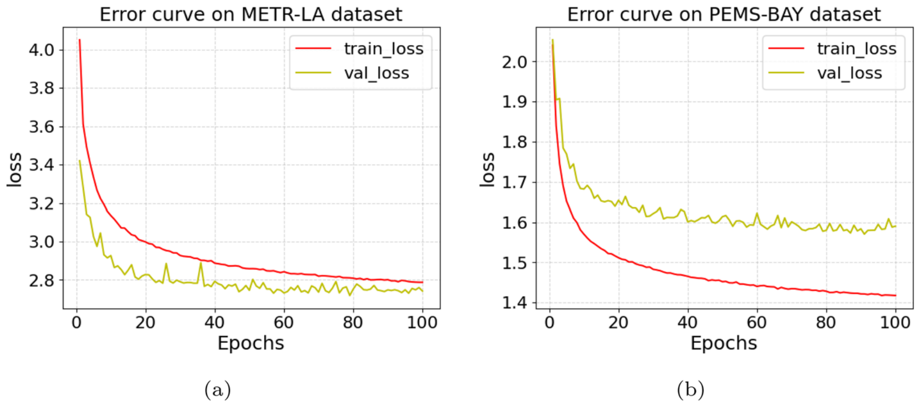


Fig. 5 Training and validation error convergence curves on two datasets. **a** Training and validation loss convergence curves on METR-LA dataset. **b** Training and validation loss convergence curves on PEMS-BAY dataset

of the GFAGNN model on the METR-LA dataset and the PEMS-BAY dataset, respectively, where the x-axis represents the number of training times and the y-axis shows the training and validation loss values. In Fig. 5, the verification loss is always lower than the training loss value. We guess this may be due to the small training sample of the dataset and the dropout operation during the training process. However, the verification loss finally converges at the 80th epoch, while the training loss is still decreasing. It can be seen that the overall trend of validation loss is similar for both models, with the loss curves first decreasing and then stabilizing during the training process with less volatility, indicating that our models have good stability.

5.6 Ablation Study

To analyze the effectiveness of our model components, we designed five variants of the GFAGNN model and conducted ablation experiments on the METR-LA and PEMS-BAY datasets.

1. w/o GCN: indicates that the adaptive graph convolution and gated fusion modules are removed and only the adaptive graph attention is retained retaining the adaptive graph attention to extract spatial-temporal features.
2. w/o ECA: indicates that the lightweight channel attention module is removed.
3. w/o GAT: indicates that the adaptive graph attention and gated fusion modules are removed, and this module does not need the adjacency matrix spatial information to extract the spatial-temporal features directly from the historical traffic flow data.
4. w/o GCN+ECA: indicates that the adaptive graph convolution and channel attention modules are removed and only the adaptive graph attention is retained.
5. w/o GAT+ECA: indicates that the adaptive graph attention and channel attention modules are removed and only the adaptive graph convolution is retained.

Table 4 shows our MAE, RMSE, and MAPE metrics on the variants. This finding proves that several important components of GFAGNN are effective. The adaptive graph convolution module has the greatest impact, with MAE metrics decreasing by 3.35%, 5.26%, and 3.19% at 15 min, 30 min, and 60 min, respectively, on the METR-LA dataset. This proves that the

Table 4 Results of the ablation experiments on the METR-LA and PEMS-BAY datasets

Method	**METR-LA**								
	15 min			30 min			60 min		
	MAE	RMSE	MAPE (%)	MAE	RMSE	MAPE (%)	MAE	RMSE	MAPE (%)
w/o GCN	2.78	5.33	7.26	3.20	6.39	8.91	3.53	7.38	10.60
w/o ECA	2.70	5.17	6.96	3.07	6.18	8.38	3.50	7.26	10.09
w/o GAT	2.74	5.27	7.40	3.13	6.35	9.03	3.55	7.43	10.69
w/o GCN+ECA	2.79	5.41	7.38	3.18	6.42	8.79	3.61	7.33	10.02
w/o GAT+ECA	2.73	5.27	7.13	3.07	6.20	8.48	3.50	7.19	10.20
GFAGNN	**2.69**	**5.15**	**6.81**	**3.04**	**6.10**	**8.12**	**3.45**	**7.12**	**9.64**
Method	**PEMS-BAY**								
	15 min			30 min			60 min		
	MAE	RMSE	MAPE (%)	MAE	RMSE	MAPE (%)	MAE	RMSE	MAPE (%)
w/o GCN	1.34	2.79	2.80	1.70	3.77	3.89	2.00	4.51	4.72
w/o ECA	1.31	2.77	2.85	1.63	3.69	3.81	1.91	4.37	4.63
w/o GAT	1.32	2.79	2.71	1.66	3.73	3.63	1.97	4.49	4.63
w/o GCN+ECA	1.33	2.81	2.74	1.65	3.71	3.64	1.96	4.47	4.61
w/o GAT+ECA	1.32	2.80	2.74	1.65	3.75	3.64	1.95	4.51	4.50
GFAGNN	**1.29**	**2.71**	**2.69**	**1.60**	**3.60**	**3.55**	**1.89**	**4.35**	**4.41**

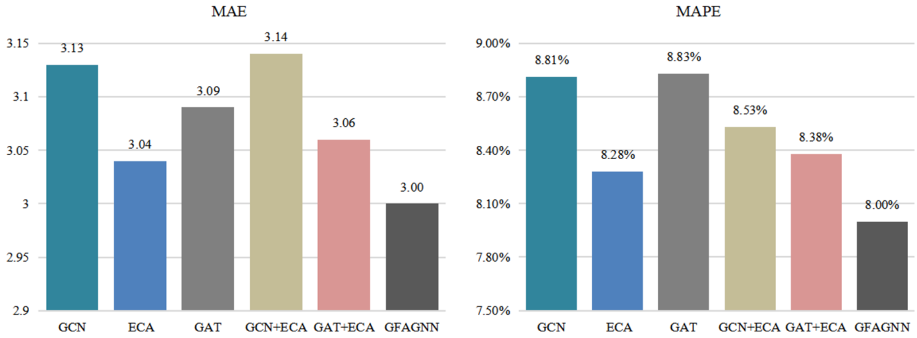


Fig. 6 Experimental results of GFANN and different variants on the METR-LA dataset

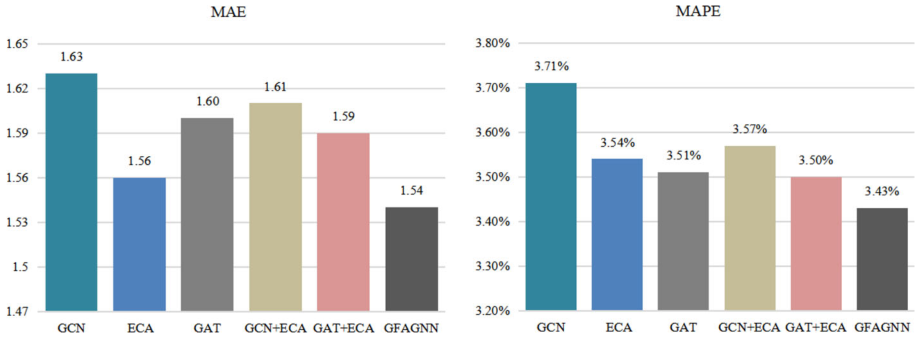
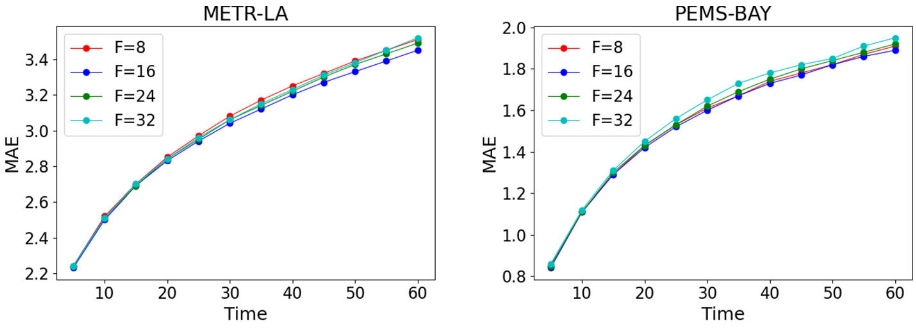


Fig. 7 Experimental results of GFANN and different variants on the PEMS-BAY dataset

hidden spatial features can be effectively mined using the adjacency matrix and adaptive learning node information. For the adaptive graph attention module, the long-term prediction has been the advantage of the attention mechanism, which can reduce the error of the long-time prediction results. ECA is lightweight channel attention, which can adjust the obtained features during the training process to improve the model performance. We also explored other combinations of these three modules, such as ignoring the adaptive graph convolution module and the lightweight channel attention module, or ignoring the adaptive graph attention module and the lightweight channel attention module. The experimental results show that these methods are not feasible, and the three modules are helpful to improve the performance of the model. In addition, to observe and compare the importance of each module more visually, we show the average values of the two metrics predicted in one hour by histograms in Figs. 6 and 7. In conclusion, the three modules used in this paper can help the model to better mine different spatial-temporal information and further improve the prediction accuracy of the model.

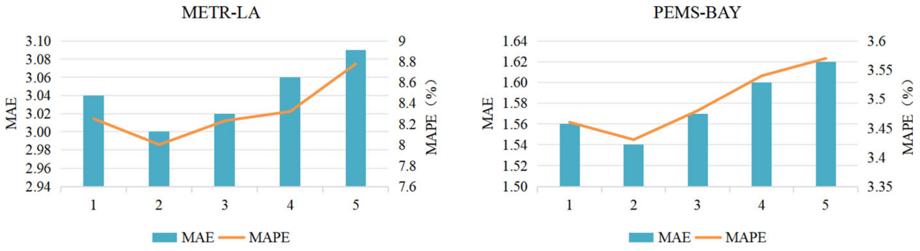
5.7 Hyperparametric Studies

To further verify the effectiveness of hyperparameter F in adaptive node embedding, we use different values in two datasets, such as F=8, F=16, F=24, and F=32. The GFAGNN is evaluated with the above variables, and the optimal value is selected by a 60-minute comparison experiment to achieve the best prediction accuracy of the model. Using MAE as



(a) (b)

Fig. 8 Variation of error for different F on two datasets: **a** the experimental errors corresponding to different F values on the METR-LA dataset; **b** the experimental errors corresponding to different F values on the PEMS-BAY dataset



(a) (b)

Fig. 9 Effects of different K values on two datasets: **a** results of GFAGNN with different values of K on METR-LA; **b** results of GFAGNN with different values of K on PEMS-BAY

the evaluation metric, the experimental results are shown in Fig. 8a shows the experimental results on the METR-LA dataset and Fig. 8b shows the experimental results on the PEMS-BAY dataset, we observe that the best performance is achieved when F=16. The possible reason for this result is that graph attention and graph convolution learning are strongest when F=16. If the embedding dimension is reduced, the model cannot fully extract spatial-temporal features, and when the embedding dimension is increased, the model may suffer from overfitting due to too many learning parameters. The above experiments show that increasing the node embedding with appropriate dimensionality can effectively improve the model prediction performance.

To illustrate the effect of different diffusion step K values on the accuracy, Fig. 9 plots the MAE and MAPE values for different k in the range of 1 to 5 on both datasets. It can be seen that for both the METR-LA and PEMS-BAY datasets, MAE and MAPE usually start at a high value before minimizing at $K = 2$ and finally increasing again with increasing k. The results are shown in Fig. 9. The general trend shown in Figure 9 proves that properly establishing spatial dependencies between nodes other than neighboring nodes has a positive impact on the model's effectiveness, and that too low or too high a value of K can have a negative effect.

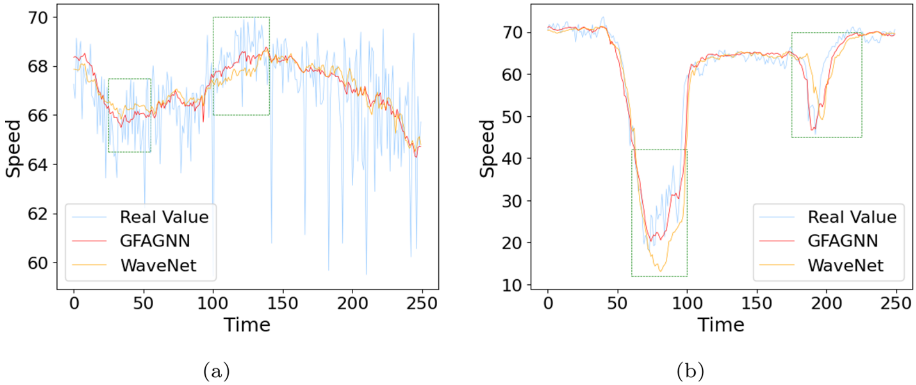


Fig. 10 Comparison of prediction curves between GFAGNN and Graph WaveNet for 60 min ahead prediction on a snapshot ofthe test data of METR-LA: **a** prediction curves on METR-LA dataset; **b** prediction curves on PEMS-BAY dataset

6 Case Study

To better demonstrate the traffic speed prediction on the road network, we randomly select a road node (sensor) to compare its detected real speed with the speed predicted by GFAGNN 60 min ago and plot the graph with the horizontal coordinate representing the time and the vertical coordinate representing the speed, in addition, we also put the Graph WaveNet model predictions into the same graph for comparison.

Figure 9a shows a node speed selected on the METR-LA dataset, and we find that the real traffic speed on this road changes more frequently. From the highlighted part of the figure (shown in the dashed box), we can see that our model has a more stable prediction performance in the face of complex traffic situations compared to Graph WaveNet. Figure 9b shows the traffic situation at a node on the PEMS-BAY dataset. The traffic speed on this road varies more drastically, and from the highlighted part of the figure, we can see that GFAGNN fits the real traffic speed better when facing the drastically changing traffic flow.

7 Conclusion

In this paper, we propose a new spatial-temporal network framework for predicting traffic flow data, namely GFAGNN. We combine extended causal convolution with adaptive spatial learning networks to capture dynamic spatial-temporal correlations effectively. Firstly, the adaptive adjacency matrix is added to the graph convolution to learn the hidden spatial association, and the self-learning node is embedded in the graph attention network to learn the dynamic spatial association. Finally, the two modules are fused through the gating mechanism to obtain the long-term and short-term spatial-temporal features. We conducted comparative experiments with other baselines on two real traffic data sets to verify the validity of the model. In addition, ablation experiments show that the design combining adaptive graph convolution and adaptive graph attention is reasonable and effective.

Our proposed model can learn spatial-temporal relationships from historical traffic data without relying on a predetermined adjacency matrix, which reduces the reliance on a priori information about the road network. We may face limitations on the quantity and quality of datasets in future work, so we will focus on how to utilize the limited amount of data for small-sample learning and improve model prediction performance.

Acknowledgements This research was supported by the National Natural Science Foundation of China (Nos.62067002 and 62062033), in part by the Natural Science Foundation of JiangXi Province (No.20212BAB202008), and in part by the Science and Technology Project of Transportation Department of Jiangxi Province, China (No.2022X0040).

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Zhang M, Zhang D, Fu S, Kujala P, Hirdaris S (2022) A predictive analytics method for maritime traffic flow complexity estimation in inland waterways. Reliab Eng Syst Saf 220:108317
2. Lv Y, Duan Y, Kang W, Li Z, Wang F-Y (2014) Traffic flow prediction with big data: a deep learning approach. IEEE Trans Intell Transp Syst 16:865–873
3. Yuan H, Li G (2021) A survey of traffic prediction: from spatio-temporal data to intelligent transportation. Data Sci Eng 6:63–85
4. Ermagun A, Levinson D (2018) Spatiotemporal traffic forecasting: review and proposed directions. Transp Rev 38:786–814
5. Ajit A, Acharya K, Samanta A (2020) A review of convolutional neural networks. In: 2020 International conference on emerging trends in information technology and engineering (ic-ETITE), pp. 1–5. IEEE
6. Wu Z, Pan S, Chen F, Long G, Zhang C, Philip SY (2020) A comprehensive survey on graph neural networks. IEEE Trans Neural Netw Learn Syst 32:4–24
7. Fang W, Zhuo W, Yan J, Song Y, Jiang D, Zhou T (2022) Attention meets long short-term memory: a deep learning network for traffic flow forecasting. Physica A 587:126485
8. Sha S, Li J, Zhang K, Yang Z, Wei Z, Li X, Zhu X (2020) Rnn-based subway passenger flow rolling prediction. IEEE Access 8:15232–15240
9. Zhang T, Guo G (2022) Graph attention LSTM: a spatiotemporal approach for traffic flow forecasting. IEEE Intell Transp Syst Magaz 14(2):190
10. Ali A, Zhu Y, Zakarya M (2022) Exploiting dynamic spatio-temporal graph convolutional neural networks for citywide traffic flows prediction. Neural Netw 145:233–247
11. Rubio L, Alba K (2022) Forecasting selected colombian shares using a hybrid arima-svr model. Mathematics 10(13):2181
12. Kumar SV, Vanajakshi L (2015) Short-term traffic flow prediction using seasonal arima model with limited input data. Eur Transp Res Rev 7:1–9
13. Wang Y, Zheng D, Luo SM, Zhan DM, Nie P (2013) The research of railway passenger flow prediction model based on BP neural network. Adv Mater Res 605:2366–2369
14. Sun Y, Leng B, Guan W (2015) A novel wavelet-SVM short-time passenger flow prediction in Beijing subway system. Neurocomputing 166:109–121
15. Li W, Sui L, Zhou M, Dong H (2021) Short-term passenger flow forecast for urban rail transit based on multi-source data. EURASIP J Wirel Commun Netw 2021:1–13
16. Ma X, Tao Z, Wang Y, Yu H, Wang Y (2015) Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. Transp Res Part C Emerg Technol 54:187–197

17. Agarap AFM (2018) A neural network architecture combining gated recurrent unit (GRU) and support vector machine (svm) for intrusion detection in network traffic data. In: Proceedings of the 2018 10th international conference on machine learning and computing, pp. 26–30
18. Wang X, Lv R, Zhao Y, Yang T, Ruan Q (2020) Multi-scale context aggregation network with attention-guided for crowd counting. In: 2020 15th IEEE International Conference on Signal Processing (ICSP), vol. 1, pp. 240–245. IEEE
19. Xu M, Dai W, Liu C, Gao X, Lin W, Qi G-J, Xiong H (2020) Spatial-temporal transformer networks for traffic flow forecasting. arXiv preprint arXiv:2001.02908
20. Han Y, Wang S, Ren Y, Wang C, Gao P, Chen G (2019) Predicting station-level short-term passenger flow in a citywide metro network using spatiotemporal graph convolutional neural networks. ISPRS Int J Geo Inf 8:243
21. Zhao L, Song Y, Zhang C, Liu Y, Wang P, Lin T, Deng M, Li H (2019) T-gcn: a temporal graph convolutional network for traffic prediction. IEEE Trans Intell Transp Syst 21:3848–3858
22. Bai L, Yao L, Li C, Wang X, Wang C (2020) Adaptive graph convolutional recurrent network for traffic forecasting. Adv Neural Inf Process Syst 33:17804–17815
23. Yu B, Yin H, Zhu Z (2017) Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting. arXiv preprint arXiv:1709.04875
24. Song C, Lin Y, Guo S, Wan H (2020) 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
25. Li Y, Yu R, Shahabi C, Li Y (2018) Diffusion convolutional recurrent neural network: data-driven traffic forecasting. In: 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30–May 3, 2018, Conference Track Proceedings, pp. 1–16
26. Wu Z, Pan S, Long G, Jiang J, Zhang C (2019) Graph wavenet for deep spatial-temporal graph modeling. In: Proceedings of the twenty-eighth international joint conference on artificial intelligence, IJCAI 2019, Macao, China, August 10–16, 2019, pp. 1907–1913
27. Van der Wel RP, Welsh T, Böckler A (2018) Talking heads or talking eyes? effects of head orientation and sudden onset gaze cues on attention capture. Attent Percept Psychophys 80:1–6
28. Zheng C, Fan X, Wang C, Qi J (2020) Gman: a graph multi-attention network for traffic prediction. Proc AAAI Conf Artif Intell 34:1234–1241
29. Jin G, Liang Y, Fang Y, Huang J, Zhang J, Zheng Y (2023) Spatio-temporal graph neural networks for predictive learning in urban computing: a survey. arXiv preprint arXiv:2303.14483
30. Huang X, Ye Y, Ding W, Yang X, Xiong L (2022) Multi-mode dynamic residual graph convolution network for traffic flow prediction. Inf Sci 609:548–564
31. Liu H, Dong Z, Jiang R, Deng J, Deng J, Chen Q, Song X (2023) Spatio-temporal adaptive embedding makes vanilla transformer sota for traffic forecasting. arXiv preprint arXiv:2308.10425
32. Li F, Feng J, Yan H, Jin G, Yang F, Sun F, Jin D, Li Y (2023) Dynamic graph convolutional recurrent network for traffic prediction: benchmark and solution. ACM Trans Knowl Discov Data 17(1):1–21
33. Lin J, Li Z, Li Z, Bai L, Zhao R, Zhang C (2023) Dynamic causal graph convolutional network for traffic prediction. arXiv preprint arXiv:2306.07019
34. Shao Z, Zhang Z, Wei W, Wang F, Xu Y, Cao X, Jensen CS (2022) Decoupled dynamic spatial-temporal graph neural network for traffic forecasting. arXiv preprint arXiv:2206.09112
35. Yang S, Li H, Luo Y, Li J, Song Y, Zhou T (2022) Spatiotemporal adaptive fusion graph network for short-term traffic flow forecasting. Mathematics 10(9):1594
36. Ni Q, Zhang M (2022) STGMN: a gated multi-graph convolutional network framework for traffic flow prediction. Appl Intell 52(13):15026–15039
37. Tian C, Chan WK (2021) Spatial-temporal attention wavenet: a deep learning framework for traffic prediction considering spatial-temporal dependencies. IET Intel Transport Syst 15:549–561
38. Kong X, Zhang J, Wei X, Xing W, Lu W (2022) Adaptive spatial-temporal graph attention networks for traffic flow forecasting. Appl Intell 52:4300–4316
39. Dai G, Ma C, Xu X (2019) Short-term traffic flow prediction method for urban road sections based on space-time analysis and gru. IEEE Access 7:143025–143035
40. Wang Q, Wu B, Zhu P, Li P, Zuo W, Hu Q (2020) Supplementary material for ‘eca-net: efficient channel attention for deep convolutional neural networks. In: Proceedings of the 2020 IEEE/CVF conference on computer vision and pattern recognition, IEEE, Seattle, WA, USA, pp. 13–19
41. Oreshkin BN, Amini A, Coyle L, Coates M (2021) Fc-gaga: Fully connected gated graph architecture for spatio-temporal traffic forecasting. In: Proceedings of the AAAI conference on artificial intelligence, vol. 35, pp. 9233–9241

42. Wu Z, Pan S, Long G, Jiang J, Chang X, Zhang C (2020) Connecting the dots: Multivariate time series forecasting with graph neural networks. In: Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining, pp. 753–763
43. Yoo B, Lee J, Ju J, Chung S, Kim S, Choi J (2021) Conditional temporal neural processes with covariance loss. In: International conference on machine learning, pp. 12051–12061. PMLR

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.