



An adaptive traffic flow prediction model based on spatiotemporal graph neural network

Tianbo Liu¹ · Jindong Zhang^{1,2,3}

Accepted: 3 April 2023 / Published online: 16 April 2023

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2023

Abstract

The traffic flow prediction task is essential to the urban intelligent transportation system. Due to the complex correlation of traffic flow data, insufficient use of spatiotemporal features will often lead to significant deviations in prediction results. This paper proposes an adaptive traffic flow prediction model AD-GNN based on spatiotemporal graph neural network. The gated temporal convolutional network captures the temporal dependence between layers. Moreover, the diffusion graph convolutional network simulates the spatial relationship between nodes. Then, the parameterized adjacency matrix is used to construct an adaptive convolutional network to adaptively mine the implicit global deep spatial dependence. The experimental results show that the model has good prediction performance on three real public datasets and can sufficiently meet real needs.

Keywords Adaptive · Traffic flow prediction · Spatiotemporal graph neural network · Spatial dependence

1 Introduction

With the continuous development of urbanization, traffic congestion and accidents faced by residents' travel also increase [1]. Therefore, the importance of the intelligent transportation system is self-evident.

The traffic flow prediction method refers to predicting the traffic flow at different locations by using the historical data collected and stored in the traffic system

✉ Jindong Zhang
zhangjindong_100@163.com

¹ College of Computer Science and Technology, Jilin University, Changchun 130012, China

² Key Laboratory of Symbol Computation and Knowledge Engineering of the Ministry of Education, Jilin University, Changchun 130012, China

³ State Key Laboratory of Automobile Simulation and Control, Jilin University, Changchun 130025, China

and combined with other information [2, 3]. If future traffic conditions can be accurately predicted, it can facilitate traffic planning and significantly improve the travel experience. Therefore, fully and accurately mining the information in the traffic data is particularly important for developing the intelligent transportation system.

In recent years, the number of available traffic datasets has gradually increased, providing essential data support for traffic prediction research. However, considering the traffic complexity in different scenarios, traffic flow prediction still faces many challenges. Firstly, the traffic flow of the urban road network has complex temporal and spatial correlations. The traffic conditions of a specific road section will be affected by the traffic conditions at different times in the nearby historical moment [4, 5]. The traffic state association between different road sections will be affected by various factors such as geometric distance, adjacency relationship, and functional similarity, showing complex spatial association. At the same time, most existing traffic flow prediction algorithms use long-term traffic history data collected by sensors in specific locations [6–8]. However, in actual traffic scenarios, there may be cases where the prediction task is cold-started due to the insufficient duration of historical data [9]. Furthermore, changes in the road network structure often occur in traffic scenarios [10], and this change will cause the corresponding parameters fitted before the change to no longer suit traffic prediction tasks under the new road network structure.

For the above reasons, we propose an adaptive traffic flow prediction model AD-GNN based on spatiotemporal graph neural network. We utilize the gated temporal convolutional network combined with diffusion convolution network to model spatiotemporal correlations. In order to calculate the influence weights of different time steps in the historical data on the prediction results, we introduce an attention mechanism to enhance the model's ability to describe the time series correlation. In addition, the adaptive adjacency matrix method is used to parameterize the adjacency matrix to capture the deep spatial dependencies between different locations based on a data-driven approach.

Given the complex temporal and spatial mode capture problem in traffic flow data, the main contribution of this paper is to introduce the attention mechanism into the process of temporal modeling to add the implicit state with stronger temporal dependence to every moment, to improve the performance of the model's temporal modeling. At the same time, the interaction of traffic features at different locations is analogous to the feature diffusion process on the graph data, and the adaptive adjacency matrix is used to capture the spatial correlation of other nodes in the road network. Considering the gradient disappearing problem caused by the number of model layers increase, this paper uses the residual and skip connection units to avoid this problem and complete the corresponding prediction task.

The structure of this paper is as follows. The related work is introduced in Sect. 2. The traffic flow prediction model is introduced in Sects. 3 and 4. In Sect. 5, we use different benchmark models for comparison to verify the effectiveness of the AD-GNN model on three datasets. In Sect. 6, we conclude and propose future research directions.

2 Related works

Traffic flow prediction methods are mainly divided into traditional time series methods and deep learning methods. The commonly method for the traditional time series method is the autoregressive integrated moving average model, also known as the ARIMA method [11]. The ARIMA method regards the sequence in a specific period as a general non-stationary sequence. It combines the two ideas of autoregression and moving average for simulating stationary time series. However, considering that the model initialization of the ARIMA method is too complicated, another standard prediction method is the vector autoregression method, also known as the VAR method [12]. The VAR method is widely used in traffic forecasting. However, such methods cannot characterize more complex nonlinear correlations, which limits the prediction accuracy.

Deep learning methods can simulate more complex spatiotemporal correlations and have higher prediction accuracy than traditional methods [13]. The DeepST method treats the traffic scene in a specific area as a two-dimensional grid [14]. It simulates the interaction between the inflow and outflow of adjacent sub-regions by combining the convolutional neural network with the residual unit. Some methods use the LSTM model for time series modeling and combine the CNN model to simulate spatial correlation to complete the corresponding traffic prediction [15, 16]. Based on the spatial relationship between different locations of the road network, the graph convolution method is usually used for the simulation of spatial association under the condition of the graph structure. Commonly used graph convolution methods include spatial domain graph convolution and frequency domain graph convolution that extends Fourier transform to graph data, which can be collectively referred to as graph convolutional network (GCN) [17]. The STGCN method uses a 1D gated CNN to model temporal associations and a graph convolution method to model spatial associations [18]. On this basis, the improved ASTGCN method [19], based on the one-dimensional time series convolution and graph convolution models, adds an attention module better to capture the correlation between different moments and sequences. The DCRNN method models the spatial dependence in the traffic map data as a diffusion process simulates the spatial correlation through diffusion convolution and builds a sequence encoding and decoding structure based on the LSTM model [20]. The STSGCN method fuses the graph data of the current moment [21], the last moment, and the next moment to construct a spatiotemporal graph containing both spatiotemporal features. The MSTIF-Net model integrates GCN structures, variational auto-encoders, and Seq2seq model to obtain the joint latent representation of urban ride-hailing situations that contain both Euclidean spatial features and non-Euclidean structural features and capture the spatiotemporal dynamics [22]. The ETGCN model combines the gated recurrent unit with GCN to capture spatiotemporal dependencies and their changing states to predict traffic velocity in the road network accurately [23]. The ASTGNN method uses dynamic graph convolution and embedding modules to capture the periodicity and spatial heterogeneity of traffic data [24]. The AUTO-DSTSGN method captures the short-term

and long-term spatiotemporal correlations by stacking deeper layers with dilation factors in increasing order and achieves good prediction results [25]. The DMVST-VGNN method integrates 1D CNN, Multi-Graph Attention Neural Network, and transformer network structures, which strengthen the learning capabilities of spatial dynamics and long-term temporal dependencies [26]. The AUTO-STS method uses the graph neural network-based architecture search module to capture localized spatiotemporal correlations and the convolutional neural network-based architecture search module to capture temporal dependencies with various ranges [27].

3 Problem formulation

For the prediction task, assuming that there are N traffic sensors in the predicted target road network, all traffic flow sequences in the target road network are defined as:

$$X_t = (x_t^1, x_t^2, \dots, x_t^N) \in R^{C \times N} \quad (1)$$

In the formula, x_t^i represents the corresponding observation value of the i traffic sensor in the target road network at time t . C is the number of traffic sequence features input by the prediction model.

In the AD-GNN model, the input is the observed value of traffic flow data collected by each sensor in the target road network during a time series window of a specific length, expressed as:

$$X^f = (X_1, X_2, \dots, X_T) \in R^{C \times N \times T} \quad (2)$$

In the formula, X^f is the input of the model in this section, and T is the time series window length of the input historical observation value.

The traffic flow data have a strong periodic characteristic and presents different rules with peak and off-peak hours. Moreover, the changing trends of traffic flow and average speed in the road network also show a substantial similarity with the change of date. Therefore, the traffic prediction model in this paper takes the position of the corresponding time in a day as the time feature and inputs it into the model together with the traffic flow feature. For the corresponding time feature at time t , the calculation method is:

$$p_t = \frac{t - t_0}{\Delta t} \quad (3)$$

In the formula, t is the current moment, and t_0 is the moment that marks the beginning of a day, that is, 0:00:00. Δt is the time span of a day, and p_t is the result of time feature calculation and marks the position of time t in all moments of a day. According to the number of observed nodes, the time feature e_t at time t is expanded to obtain $P_t \in R^N$ to match the input dimension of traffic flow features. The final time feature input is:

$$P = (P_1, P_2, \dots, P_T) \in R^{N \times T} \tag{4}$$

In the formula, P_T is the time characteristic corresponding to all nodes at time T .

In order to facilitate the description of the spatial distribution characteristics, this section defines the road network as a graph structure G , where V represents the set of nodes, that is, the set of sensors in the target road network. E is the edge set, which represents the connection state between sensors in the road network; A is an adjacency matrix, which is used to represent the connection relationship in the edge set, and its assignment method is:

$$A_{i,j} = \begin{cases} 1, & \langle v_i, v_j \rangle \in E(G) \\ 0, & \langle v_i, v_j \rangle \notin E(G) \end{cases} \tag{5}$$

In the formula, $A_{i,j}$ is the corresponding element in row i and column j in the adjacency matrix, and $\langle v_i, v_j \rangle$ represents the node pair composed of node i and node j .

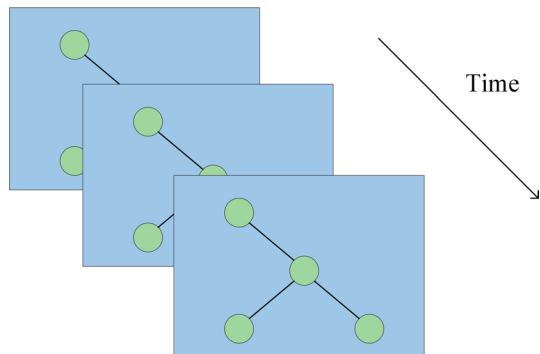
The traffic state at a specific moment corresponds to a set of graph data. Within the continuous observation period, traffic graphs at multiple moments form spatiotemporal graph data, as shown in Fig. 1.

The target road network is defined as a graph structure, and its corresponding structure is defined as G . Given a traffic road network composed of N sensor nodes, the time series window length is T . The corresponding prediction task is to use the input data to generate the predicted values of the traffic flow characteristics of all nodes in the road network at H moments in the future. Its relationship is expressed as follows:

$$[X^{in}, G] \xrightarrow{f} Y \tag{6}$$

In the formula, X^{in} is the overall input of the prediction model, obtained by concatenating the traffic flow feature input X^f and the time feature P . f is the predicted function, and Y is the predicted value.

Fig. 1 Spatiotemporal graph data



4 Adaptive traffic flow prediction model AD-GNN

The specific process of using historical observations for traffic prediction proposed in this section is as follows: Firstly, the input traffic flow features and input time features are spliced and based on the fully convolutional neural network module splicing. The final input data are subjected to feature dimension transformation. The fully convolutional network module can extract the features of the data while changing the dimension of the features, which improves the operation efficiency. Afterward, the time series correlation weights at different moments in the input time series window are calculated based on the multi-head attention mechanism. The time series weighted feature transformation is performed to capture the correlation of different historical moments dynamically. Afterward, temporal modeling is performed based on the temporal convolutional network, which encodes the temporal modality of the historical input information to extract the temporal correlation of the input data. For spatial dependence, the node adjacency relationship is used as a priori, and the spatial feature transformation process is realized through diffusion convolution. In addition, the graph convolution operation is performed by parameterizing the adjacency matrix to adaptively mine the hidden spatial correlation in the road network. The above spatiotemporal feature transformation extraction process is combined with the residual connection unit and the skip connection unit. So that the model can avoid the problem of gradient disappearance or gradient dispersion to improve the feature capture performance of the model. After that, the feature dimension is transformed through the convolution module. Finally, the traffic flow prediction values corresponding to all node sequences in the target road network are obtained at each moment in the prediction window period. The process flow of the adaptive traffic flow prediction method based on the spatiotemporal graph neural network proposed in this section is shown in Fig. 2.

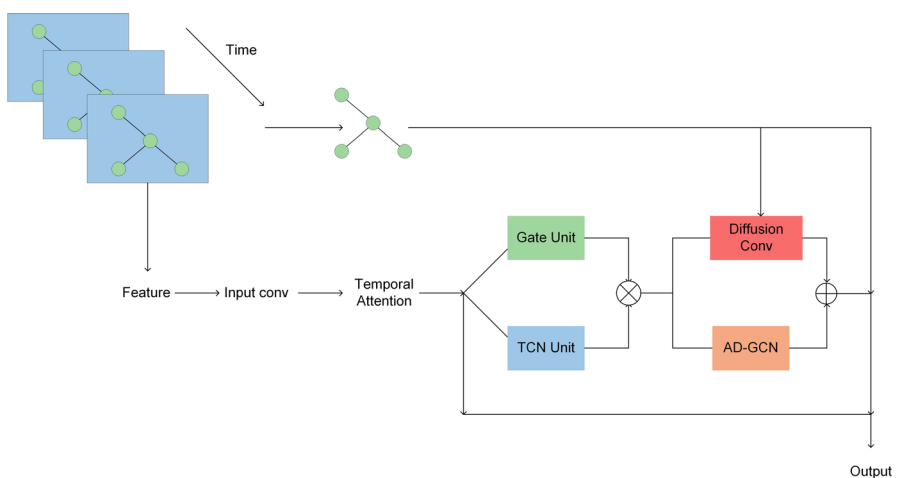


Fig. 2 The architecture of the AD-GNN model

The AD-GNN model is based on the fully convolutional neural network module for feature extraction and feature dimension transformation of model input and output data. Its calculation process is as follows:

$$X^l = \text{ReLU}(W_v^l * X^{l-1} + b_v^l) \tag{7}$$

In the formula, $*$ is the convolution operation, and ReLU is the activation function. $W_v^l \in R^{V_l \times V_{l-1} \times 1 \times 1}$ is the convolution kernel, $b_v^l \in R^{V_l}$ is the paranoid parameter, and l is the number of total convolution layers. $X^{l-1} \in R^{V_{l-1} \times N \times T}$ is the input feature of the full convolution module of this layer, $X^l \in R^{V_l \times N \times T}$ is the output feature of the full convolution operation of this layer, and V_l and V_{l-1} are the feature dimensions of the input and output features. In the fully convolutional network module proposed in this section, the convolution kernel size is $1 * 1$, and the moving step in the convolution kernel filtering process is 1. The essence of its operation is to transform the feature dimension of multi-dimensional time series data to complete the process of increasing or reducing the feature's dimension without changing the data's overall structure and generating new feature representations.

Considering the characteristics of the traffic flow prediction task, this model uses the MSE function as the loss function.

$$MSE = \frac{1}{\Omega H} \sum_{i=1}^{\Omega} \sum_{h=1}^H |\hat{y}_{T+h}^i - y_{T+h}^i|^2 \tag{8}$$

In the formula, Ω is the number of data samples, H is the length of the prediction time series window, and \hat{y}_{T+h}^i and y_{T+h}^i are the predicted value and actual value of the i th data sample at the h th time step in the future, respectively.

4.1 Temporal attention mechanism

The time series correlation of traffic flow data presents complex nonlinearity and has different time-dependent characteristics in different periods. The traffic state at a specific moment is closely related to its adjacent moments and affected by distant historical moments. Commonly used time series modeling methods such as LSTM have a solid ability to describe time series trends. However, the ability to capture the time series correlation implied at different times is weak for a specific period. The attention mechanism can solve this problem by extracting the input features and then using the obtained representation vector to calculate the weight of the features to perform feature allocation to select specific inputs and achieve the effect of optimizing feature allocation [28].

The traffic flow prediction model proposed in this section uses the TCN method for temporal modeling and adds a temporal multi-head attention mechanism. In the process of feature encoding, for each moment in the history window, calculate the correlation weights between other time series information and this moment, and use the characteristics of other moments to perform feature transformation on this moment. In this way, the time series correlation can be adaptively

captured in transforming the time series features. The process of the n th group of attention calculations corresponding to the output results is:

$$Q^n = \text{ReLU}\left(x^A W_Q^n + b_Q^n\right) \tag{9}$$

$$K^n = \text{ReLU}\left(x^A W_K^n + b_K^n\right) \tag{10}$$

$$S^n = \frac{Q^n(K^n)}{\sqrt{d}} \tag{11}$$

$$E_{ij}^n = \frac{\exp\left(S_{ij}^n\right)}{\sum_{j=1}^T \exp\left(S_{ij}^n\right)} \tag{12}$$

$$V^n = \text{ReLU}\left(x^A W_V^n + b_V^n\right) \tag{13}$$

$$\text{head}^n = E^n V^n \tag{14}$$

In the formula, $X^A \in R^T$ is the input timing feature corresponding to the attention mechanism; $Q^n \in R^{T \times d}$ is the query vector in the self-attention mechanism; $K^n \in R^{T \times d}$ is the key-value vector, and $V^n \in R^{T \times d}$ is the value vector. Moreover, d is the vector feature dimension. $W_Q^n \in R^d$, $W_K^n \in R^d$, $W_V^n \in R^d$ are self-attention. The weight parameters in the attention vector calculation process, $b_Q^n \in R^d$, $b_K^n \in R^d$, $b_V^n \in R^d$ are bias parameters; $S^n \in R^{T \times T}$ is the feature score matrix. $E^n \in R^{T \times T}$ is the influence weight of features at different moments in the input data, and $\text{head}^n \in R^{T \times d}$ is the corresponding output of the n th group of attention calculations result.

For the temporal multi-head attention mechanism, the calculation results of multiple sets of attention mechanisms are spliced in the feature dimension. $x^L = [\text{head}^1, \text{head}^2, \dots, \text{head}^n]$ is the concatenated value of output results for all attention channels. The feature transformation is performed on the input features in different channels, which can extract richer time series correlations at different times to improve the network model’s overall timing correlation simulation performance. The linear dimension adjustment is performed on the feature transformation results obtained by splicing to obtain the output features of the multi-head attention mechanism module.

Figure 3 shows the computational flow of a single-group temporal attention mechanism. $f_Q(x)$, $f_K(x)$, and $f_V(x)$ are the corresponding calculation functions of the query vector, key-value vector, and value vector in the attention mechanism. Using the time series attention mechanism to solve the time series correlation degree at different moments can optimize the distribution of time series features, weight the time series information at different times to specific moments, and better model the time series dependence.

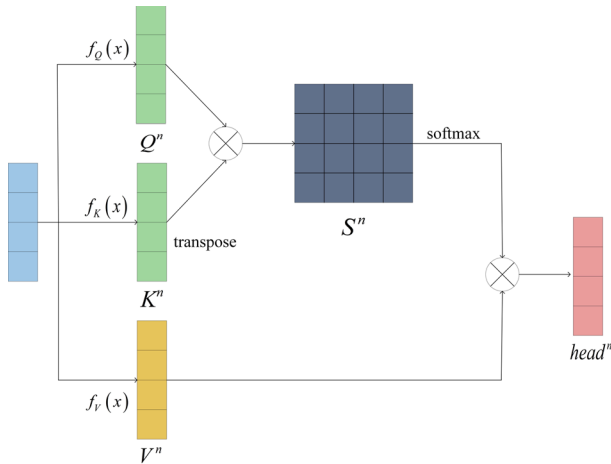


Fig. 3 Temporal self-attention mechanism

4.2 Gated temporal convolutional network

The traditional RNN structure and its variants, such as LSTM and GRU, mainly encode continuous time series feature vectors to simulate the continuous change trend of features. However, the ability of this type of network structure to process temporal features in parallel is weak, which is reflected in the fact that the encoding output is more sensitive to feature changes at adjacent moments. In the cyclic decoding process, using the output value at the last moment as the input value at the next moment will easily cause error accumulation and affect the prediction performance. Therefore, this model uses the time series convolutional network (TCN) as the core unit of time series correlation simulation and combines the causal convolution and dilated convolution structure to perform time series feature transformation. The TCN structure performs parallel convolution calculations on the input time series features to improve the perception ability of the prediction framework for information at distant historical moments [29]. The causal convolution structure ensures that in the time series feature extraction, the features of the future time are only affected by the current and historical time features, and the time series constraints are strictly followed while extracting the continuous change trend of the sequence. The hollow convolution structure helps to reduce the number of network stacking layers to improve computational efficiency. The timing convolution calculation method proposed in this section is:

$$x^L * f(t) = \sum_{s=0}^{K_t-1} f(s)x^L(t - d_f \times s) \tag{15}$$

In the formula, $x^L \in R^T$ is the input one-dimensional time series signal, f is the time series convolution kernel, and K_t is the size of the time series convolution kernel. d_f is the hole factor representing the distance between adjacent units in the convolution

kernel. When stacking multiple sequential convolutional layers, the receptive field of the convolution operation increases accordingly, which can capture multi-level complex timing correlations in the global timing mode and obtain information-rich timing encoding vectors. Furthermore, due to the existence of the cavity structure, the requirement for the depth of the convolution stack is reduced, avoiding the overfitting problem caused by the excessive complexity of the parameters.

The timing correlation simulation method proposed in this section combines the gating mechanism with the TCN module. Use the same network structure as the TCN module to transform the input time series features and use a specific activation function to scale the transformed time series features. The dimensional structure of the obtained representation vector is the same as the structure of the time series information to be processed. To regulate the amount of the information flow going through the gate control unit, the value of each element denotes the ratio that is permitted to pass in the timing information. The timing gating mechanism proposed in this section is calculated as follows:

$$X^S = \tanh(\theta_1 * X^L + b_1) \odot \sigma(\theta_2 * X^L + b_2) \quad (16)$$

In the formula, $X^L \in R^{V \times N \times T}$ is the input timing signal, and $X^S \in R^{V \times N \times (T-K+1)}$ is the output result of timing-gated convolution. $\theta_1 \in R^{V \times V \times 1 \times K}$ and $b_1 \in R^V$ are the convolution kernel and bias parameters used to capture timing correlation, and $\theta_2 \in R^{V \times V \times 1 \times K}$ and $b_2 \in R^V$ are the convolution kernel and bias parameters used to calculate the value of the gating unit. \tanh and σ are the tanh activation function and sigmoid activation function, and \odot represents the Hadamard product operation. The value range of the gating unit after the sigmoid activation function mapping is between 0 and 1. By multiplying the corresponding elements, the timing feature ratio output to the next layer of timing convolution is adjusted, which can further improve the timing-dependent simulation performance.

4.3 Diffusion graph convolutional neural network

For different locations in the road network, there are complex spatial correlations. The graph convolutional neural network extracts the spatial features of the road network by aggregating neighborhood information [30]. Among them, the spatial domain graph convolution method simulates the aggregation process of features between nodes. It uses the target node and its neighbor nodes to complete the feature update process. This method has a solid ability to capture local spatial features and has strong interpretability.

The prediction model simulates the spatial correlation of road networks by analogizing the interaction of traffic features at different locations to the process of feature diffusion on graph data. Feature diffusion on graph data can be compared to transferring from one state to another in the Markov chain state space. The following state has nothing to do with the historical state, only the current state.

Figure 4 shows the form of the state transition matrix of the graph data. T_M is the state transition matrix, and the values of the elements in it represent the characteristic transfer probability between the corresponding quantity nodes. Assuming that the

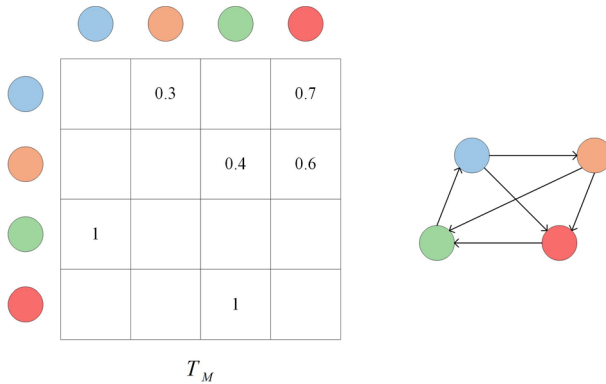


Fig. 4 State transition matrix of graph data

initial feature state distribution of the graph data is X , after a single feature transfer, its feature state distribution is $T_M X$. After multiple feature transfers, its feature state distribution is:

$$PX = \left(\sum_{k=0}^{\infty} \beta(1 - \beta)^k T_M^k \right) X \tag{17}$$

In the formula, β is the transition probability, k is the number of feature transitions, and PX is the feature state distribution after the transition. For the graph data composed of all nodes in the traffic road network, its state transition matrix can be solved by the adjacency matrix. Let the adjacency matrix of road network graph data be A , representing the adjacency relationship of nodes. The corresponding out-degree diagonal matrix is D_O , representing the number of connections between each target node and other nodes in the road network. The diagonal elements represent the number of connections, and the other position elements are all zero. The corresponding state transition matrix is $D_O^{-1}A$. When the road network graph is a directed graph, the corresponding input diagonal matrix is D_I , which represents the number of other nodes in the road network connected to each target node. The corresponding reverse state transition matrix is $D_I^{-1}A^T$. The two-way diffusion of features between nodes can enable the state transition process to have the ability to simulate the correlation between upstream and downstream nodes, which improves the flexibility of feature transfer. Through the state transition matrix, the feature state distribution after multiple feature transfers between nodes can be calculated as follows:

$$PX = \left(\sum_{k=0}^{\infty} \beta(1 - \beta)^k (D_O^{-1}A)^k \right) X + \left(\sum_{k=0}^{\infty} \beta(1 - \beta)^k (D_I^{-1}A^T)^k \right) X \tag{18}$$

For the above feature transfer process, when the number of feature transfers k tends to infinity, PX tends to be stable. For the diffusion convolution module of the AD-GNN model, the number of feature transfers is limited. Each feature transfer process

is assigned corresponding trainable convolution parameters. The calculation method of feature diffusion of road network nodes is:

$$X^Z = \sum_{k=0}^K (D_O^{-1}A)^k X\theta_{k,O} + \sum_{k=0}^K (D_I^{-1}A^T)^k X\theta_{k,I} \quad (19)$$

In the formula, $A \in R^{N \times N}$ is the adjacency matrix of the road network graph structure, $A^T \in R^{N \times N}$ is the transposition matrix of the adjacency matrix, and $D_O \in R^{N \times N}$ and $D_I \in R^{N \times N}$ are the out-degree diagonal matrix and in-degree diagonal matrix of the road network graph structure. $\theta_{k,O} \in R^{M \times F}$ and $\theta_{k,I} \in R^{M \times F}$ are the k steps features convolution parameters of feature transfer and reverse feature transfer corresponding to the diffusion process. M and F are the dimensions of the input and output features of the diffusion graph convolution operation; K is the number of feature diffusion steps; $X \in R^{N \times M}$ is the node input; and $X^Z \in R^{N \times F}$ is the output corresponding to the output features. Using the diffusion mentioned above convolution operation, it is possible to effectively simulate the spatial correlation within a limited number of steps and aggregate local traffic features. Moreover, the bidirectional structure of feature transfer enables the structure to describe spatial correlations, such as the upstream and downstream influences of roads when facing complex road networks, so it is more flexible to capture spatial dependencies.

4.4 Adaptive graph convolutional network

The spatial dependence between different nodes in the traffic road network often presents complex characteristics. Taking the adjacency relationship or straight-line distance as the basis for the node characteristics of the aggregation graph can only capture the local spatial correlation inside the road network. However, for nodes with long adjacency relationships or straight-line distances in the road network, the traditional graph convolution structure cannot capture their spatial dependence [31], then resulting in the lack of spatial correlation information, which reduces the performance of structure feature simulation.

The traffic flow prediction framework uses the adaptive adjacency matrix to capture the hidden deep spatial dependence in the road network structure and adds discrimination to the feature transfer process between the same node and corresponding different adjacent nodes. In addition, the adaptive adjacency matrix establishes the implicit connection relationship between nodes with distant or nonexistent adjacency in the road network. It describes its spatial dependence state, effectively supplementing the spatial features lost by the localized diffusion convolution operation. Constructing the adaptive adjacency matrix and the adaptive convolution module does not require knowledge such as node adjacency or straight-line distance as a priori, making the model structure more concise. The adaptive adjacency matrix calculation process proposed in this section is:

$$M_1 = \tanh(\alpha E_1 \theta_1) \quad (20)$$

$$M_2 = \tanh(\alpha E_2 \theta_2) \tag{21}$$

$$A_{adp} = \text{ReLU}(\tanh(\alpha(M_1 M_2^T - M_2 M_1^T))) \tag{22}$$

In the formula, α is the hyperparameter that controls the saturation rate of the activation function, $E_1 \in R^{N \times U}$ and $E_2 \in R^{N \times U}$ are the embedding representation vectors of the road network nodes, and $\theta_1 \in R^{U \times U}$ and $\theta_2 \in R^{U \times U}$ are the matrix parameters of the embedding representation vector transformation. Moreover, U is the hidden state dimension in constructing the adaptive matrix; $M_1 \in R^{N \times U}$ and $M_2 \in R^{N \times U}$ generate the required variable matrix for the adaptive matrix. The calculation result A_{adp} is the adaptive adjacency matrix. The adaptive adjacency matrix obtained by the above calculation method has a lower triangular matrix structure with zero values at the upper right of the diagonal. So it is more suitable for capturing the space in the process of one-way feature transfer between nodes. In addition, the number of nodes that are spatially associated with nodes is usually limited. If all nodes are considered to be associated with other nodes, it will increase the computational overhead of the spatial dependency capture process. Therefore, based on the adaptive adjacency matrix A_{adp} , for each node in the road network, keep the k nodes closest to its adjacency relationship as spatially associated nodes, and keep the corresponding values in the adjacency matrix. Furthermore, for other nodes, the corresponding value in the adjacency matrix is zero, and the adaptive adjacency matrix A_{adp} is obtained after processing. The spatial node correlation implied by A_{adp} is more in line with the actual situation in the traffic scene, so the ability to capture the spatial dependence of the road network is substantial, and the matrix structure is more sparse, which significantly reduces the computational overhead. The calculation process of graph convolution operation using an adaptive adjacency matrix is as follows:

$$X^A = \sum_{k=0}^K \tilde{A}_{adp}^k X \theta_{k,OA} + \sum_{k=0}^K \tilde{A}_{adp}^T X \theta_{k,IA} \tag{23}$$

In the formula, $\tilde{A}_{adp} \in R^{N \times N}$ is the adaptive adjacency matrix, and $\tilde{A}_{adp}^T \in R^{N \times N}$ is the transpose matrix of the adaptive adjacency matrix. $\theta_{k,OA} \in R^{M \times F}$ and $\theta_{k,IA} \in R^{M \times F}$ are the feature transfer and reverse feature transfer volumes corresponding to the k step feature diffusion process in the adaptive graph convolutional neural network product parameters. M and F are the dimensions of the input and output features of the adaptive graph convolution operation; K is the number of steps for feature transfer; and $X \in R^{N \times M}$ and $X^A \in R^{N \times F}$ are the node input and output features corresponding to the diffusion graph convolution operation. Through the graph convolution operation, the implicit spatial correlation between road network nodes can be mined from the data features to break through the limitations of the adjacency relationship and straight-line distance between nodes and capture the deep spatial dependence in the road network. For the two transfer directions of features between nodes, the adaptive graph convolution operation is performed separately, which conforms to the asymmetric nature of the spatial influence of the road

network structure and can adaptively extract the correlation between nodes from the data.

4.5 Residual connection unit and skip connection unit

The traffic flow prediction model stacks multilayer temporal convolutional networks and spatial graph convolutional networks to describe the spatiotemporal patterns in traffic flow information. However, as the number of network layers increases, the gradient will disappear due to the multiplication effect, and the convergence speed during network training will also be significantly reduced. The AD-GNN model applies the residual connection and skip connection to the multilayer spatiotemporal feature extraction process. After the residual connection unit maps the input features of each layer of the spatiotemporal extraction network, the output features corresponding to this layer are summed. The skip connection unit maps the shallow spatiotemporal features and sums them with the final calculation results of spatiotemporal feature extraction. The feature mapping process of the residual connection unit and the skip unit is also completed through the fully convolutional network. The feature mapping calculation process is as follows:

$$X_{res}^j = \text{ReLU}(W_{res}^j * X^j + b_{res}^j) \quad (24)$$

$$X_{skip}^j = \text{ReLU}(W_{skip}^j * X^j + b_{skip}^j) \quad (25)$$

In the formula, $*$ is the convolution operation, $W_{res}^j \in R^{F \times F \times 1 \times 1}$ and $W_{skip}^j \in R^{F \times F \times 1 \times 1}$ are the convolution kernels, $b_{res}^j \in R^F$ and $b_{skip}^j \in R^F$ are the paranoid parameters, and F is the feature dimension of the hidden state in the spatiotemporal feature extraction process. j is the number of layers corresponding to the current feature extraction module, and $X^j \in R^{F \times N \times T}$ is the input data of the spatiotemporal feature extraction network module of the j th layer. The method is based on the fully convolutional network to complete the corresponding feature mapping of the residual connection and skip the connection process.

For each layer of spatiotemporal extraction network module, the calculation method of combined residual connection unit is:

$$X_{st}^{j+1} = f_{st}^j(X_{st}^j) + f_{res}^j(X_{st}^j) \quad (26)$$

In the formula, f_{st}^j is the feature mapping function of spatiotemporal feature extraction, and f_{res}^j is the feature mapping function of the residual connection unit. Moreover, $X_{st}^j \in R^{F \times N \times T}$ and $X_{st}^{j+1} \in R^{F \times N \times T}$ are the input data of the j th layer and $j + 1$ th layer spatiotemporal feature extraction network modules, respectively. The identity mapping structure is established through the residual connection unit, which solves the problem of gradient disappearance caused by the increase in network layers to a certain extent.

For the overall output representation of the spatiotemporal feature extraction network, the calculation method combined with skip connection units is:

$$\bar{X}_{st} = X_{st} + f_{skip}^1(X_{st}^1) + f_{skip}^2(X_{st}^2) + \dots + f_{skip}^J(X_{st}^J) \quad (27)$$

In the formula, J is the number of layers of the spatiotemporal feature extraction network, and f_{skip} is the feature mapping function of the skip connection unit. $X_{st} \in R^{F \times N \times T}$ is the overall output representation after the spatiotemporal feature extraction network operates the J layer, and $X_{st}^j \in R^{F \times N \times T}$ is the spatiotemporal prediction output combined with the skip connection features of each layer characterization. The skip connection unit introduces the shallow network information into the output result so that the output representation obtained by the deep feature extraction module contains different levels of mode information, and the description of the implicit spatiotemporal correlation of the data is complete.

The process of the AD-GNN model is shown as follows.

Algorithm 1 The adaptive traffic flow prediction model AD-GNN

- 1: Concatenate the input traffic flow features with the input time features
 - 2: Transform the feature dimension of the input data
 - 3: Calculate the timing correlation weights at different moments
 - 4: **while** start time! = end time **do**
 - 5: Encoding temporal modalities to extract temporal associations
 - 6: Using diffusion convolution to realize spatial feature transformation
 - 7: Adaptively mining hidden spatial associations by parameterized matrix
 - 8: Using residual connected unit to improve the capture performance
 - 9: **for** each moment in the time window **do**
 - 10: **if** the current time equals end time **then**
 - 11: Get the output of the model as the predicted flow value
 - 12: **else**
 - 13: Cycling the spatio-temporal feature transformation process
 - 14: **end if**
 - 15: **end for**
 - 16: **end while**
-

5 Experimental simulations

5.1 Environment introduction

In order to measure the predictive performance of our model, we conducted a series of related experiments. The hardware environment and software environment are shown in Tables 1 and 2.

This paper uses three real public datasets for performance evaluation experiments, namely PeMSD4 dataset, PeMSD8 dataset and 2020-CCF spatiotemporal training dataset. Among them, the PeMS series of datasets are collected by the California Department of Transportation's Performance Measurement System

Table 1 Hardware environment

Options	Parameters
CPU	Intel(R) Core(TM) i5-10400F CPU @ 2.90GHz
RAM	16.0GB
GPU	NVIDIA GeForce RTX 1650
Graphics memory	8.0GB

Table 2 Software environment

Options	Parameters
Operating system	Ubuntu 18.04
Python	3.8
CUDA	10.2
PyTorch	1.6

Table 3 Experimental parameter settings

Parameter	Value
Batch size	64
Learning rate	0.001
Dropout	0.3
Hidden size	32
Weight decay	0.0001
The number of attention heads	2
The number of layers	3
Optimizer	Adam

(Performance Measurement System, PeMS for short). The system deploys traffic sensors in significant areas of California's highway network with data collection intervals of 30 s. The 2020-CCF spatiotemporal training dataset was provided by Didi for the 2020-CCF Big Data and Computational Intelligence Competition.

Among them, the specific content of each dataset is as follows.

PeMSD4 dataset: The period is from January 1, 2018, to February 28, 2018, which contains traffic flow data from 307 San Francisco Bay Area traffic sensors. It contains three sequences of traffic flow, lane occupancy, and average speed.

PeMSD8 dataset: The period is from July 1, 2016, to August 31, 2016, which contains traffic flow data from 170 traffic sensors in San Bernardino County. It contains three sequences of traffic flow, lane occupancy, and average speed.

2020-CCF spatiotemporal training dataset: The period is from July 1, 2019, to July 31, 2019, which contains topological information on different roads in Xi'an city in July 2019 and road conditions at other times.

5.2 Baseline models and evaluation metrics

In order to verify the effectiveness of the model, this paper uses multiple public datasets to conduct performance evaluation experiments and uses the classical time series prediction model and the spatiotemporal traffic flow prediction model as the benchmark model to compare the prediction performance with the AD-GNN model. The baseline models used in the experiment are as follows:

(1) **Autoregressive Integrated Moving Average Model (ARIMA)** [11]: Transform non-stationary series by differential method, and combine autoregressive and moving average methods for time series prediction.

(2) **Vector autoregression (VAR)** [12]: The univariate autoregression method is extended to multivariate series, which can simulate the linear relationship between variables.

(3) **Spatiotemporal Graph Convolutional Network (STGCN)** [18]: Use temporal and graph convolutional neural networks to construct spatiotemporal blocks and stack multiple convolutional blocks for traffic flow prediction.

(4) **Diffusion Convolutional Recurrent Neural Network (DCRNN)** [20]: Capture spatial dependencies through diffusion graph convolution and temporal dependencies through recurrent neural networks. Then, introduce a planned sampling mechanism to build an encoding-decoding structure to improve long-range time series prediction performance.

(5) **Adaptive Graph Convolutional Recurrent Network (AGCRN)** [32]: An adaptive parameter learning module is proposed to capture specific node patterns. Furthermore, a data generation module is proposed to derive the dependencies between traffic sequences and capture the time sequence through loop structure dependencies.

(6) **Evolution Temporal Graph Convolutional Network (ETGCN)** [23]: The ETGCN model first fuses multiple graph structures and utilizes graph convolutional network (GCN) to model spatial correlation. Then, the gated recurrent unit is combined with GCN to capture spatial-temporal correlations and their changing status, simultaneously.

(7) **Attention based Spatial–Temporal Graph Neural Network (ASTGNN)** [24]: The ASTGNN method uses dynamic graph convolution and embedding modules to capture the periodicity and spatial heterogeneity of traffic data.

(8) **Automated Dilated spatiotemporal Synchronous Graph Network (AUTO-DSTSGN)** [25]: The Auto-DSTSGN method captures the short-term and long-term spatiotemporal correlations by stacking deeper layers with dilation factors in increasing order and achieves good prediction results.

(9) **Automated Spatiotemporal Synchronous Model (AUTO-STS)** [27]: The AUTO-STS method uses the graph neural network-based architecture search module to capture localized spatiotemporal correlations and the convolutional neural network-based architecture search module to capture temporal dependencies with various ranges.

To compare the prediction performance, this paper uses the following indicators to evaluate the model's prediction performance.

MAE: The mean value of the absolute value of the error between the predicted value and the actual value, which can intuitively reflect the size of the prediction error.

$$MAE = \frac{1}{\Omega H} \sum_{i=1}^{\Omega} \sum_{h=1}^H |\hat{y}_{T+h}^i - y_{T+h}^i| \quad (28)$$

RMSE: The square root of the error between the predicted value and the actual value, which is more sensitive to errors with high dispersion.

$$RMSE = \sqrt{\frac{1}{\Omega H} \sum_{i=1}^{\Omega} \sum_{h=1}^H |\hat{y}_{T+h}^i - y_{T+h}^i|^2} \quad (29)$$

MAPE: The ratio of the absolute value of the error between the predicted value and the true value to the true value, reflecting the relative size of the error.

$$MAPE = \frac{1}{\Omega H} \sum_{i=1}^{\Omega} \sum_{h=1}^H \left| \frac{\hat{y}_{T+h}^i - y_{T+h}^i}{y_{T+h}^i} \right| \quad (30)$$

In the formula, Ω is the number of data samples, H is the length of the prediction time series window, and \hat{y}_{T+h}^i and y_{T+h}^i are the predicted value and actual value of the i th data sample at the h th time step in the future, respectively.

5.3 Analysis of results

The relevant parameters are set in Table 3.

For the three real traffic datasets mentioned above, we predict the traffic flow in the next hour ($H=12$). The specific prediction results are as follows:

Tables 4, 5 and 6 show the prediction evaluation of the AD-GNN model and other prediction models on the three datasets. The smaller the evaluation index value, that

Table 4 Comparison of predictive performance between AD-GNN and other models on PeMSD4 dataset

	MAE	RMSE	MAPE(%)
ARIMA	29.24/33.83/43.67	50.12/56.57/68.69	19.78/25.71/29.84
VAR	22.36/24.34/27.51	35.81/38.35/42.32	14.63/16.37/18.79
STGCN	20.27/22.72/27.08	32.29/35.68/41.34	13.03/14.83/17.85
DCRNN	19.72/21.73/25.43	31.30/34.40/39.82	12.96/14.28/16.94
AGCRN	19.02/20.14/22.65	30.42/32.10/35.71	12.93/13.79/15.41
ETGCN	18.83/19.78/22.31	30.27/31.88/35.19	12.71/13.24/14.93
ASTGNN	18.56/19.44/21.92	29.85/31.29/34.61	12.48/12.82/14.19
AUTO-DSTSGN	<u>18.07/18.95/20.45</u>	<u>28.97/30.45/33.54</u>	<u>11.32/11.98/13.02</u>
AUTO-STG	18.21/19.16/21.47	29.41/30.91/34.07	11.96/12.21/13.65
AD-GNN	17.82/18.70/19.89	28.68/30.10/32.89	10.80/11.65/12.77

Table 5 Comparison of predictive performance between AD-GNN and other models on PeMSD8 dataset

	MAE	RMSE	MAPE(%)
ARIMA	24.24/30.78/38.67	42.69/48.26/56.10	17.54/21.26/24.93
VAR	16.87/19.82/24.22	25.19/29.31/35.13	11.63/13.37/14.92
STGCN	16.14/17.89/21.24	24.33/27.93/32.94	10.33/11.18/12.86
DCRNN	15.02/16.56/19.39	23.36/25.88/29.88	9.65/10.62/12.45
AGCRN	14.75/15.79/17.92	23.25/25.12/28.54	9.36/10.62/11.19
ETGCN	14.53/15.38/17.41	22.98/24.68/27.91	9.19/10.24/11.03
ASTGNN	14.16/15.14/17.13	22.31/24.19/27.23	9.01/9.89/10.84
AUTO-DSTSGN	<u>13.62/14.49/16.09</u>	<u>21.65/23.35/25.84</u>	<u>8.70/9.65/10.52</u>
AUTO-STS	13.96/14.87/16.63	21.89/23.67/26.67	8.83/9.71/10.65
AD-GNN	13.42/14.24/15.50	21.44/23.05/25.19	8.59/9.30/10.41

Table 6 Comparison of predictive performance between AD-GNN and other models on 2020-CCF spatiotemporal training dataset

	MAE	RMSE	MAPE(%)
ARIMA	6.46/7.19/8.34	8.75/10.24/11.01	15.52/19.87/22.37
VAR	6.42/7.08/7.77	8.54/9.69/10.12	15.37/19.61/21.98
STGCN	6.19/6.85/7.15	8.37/9.21/9.79	15.21/19.26/20.84
DCRNN	6.13/6.66/7.04	8.26/8.96/9.64	15.08/18.75/20.36
AGCRN	6.09/6.62/6.98	7.18/7.77/8.53	14.89/18.54/19.77
ETGCN	5.78/6.31/6.65	6.86/7.49/8.05	14.68/17.98/18.96
ASTGNN	5.59/6.12/6.40	6.61/7.12/7.51	14.03/17.36/18.14
AUTO-DSTSGN	<u>5.05/5.49/5.82</u>	<u>6.12/6.64/7.01</u>	<u>13.07/15.65/16.92</u>
AUTO-STS	5.32/5.81/6.09	6.37/6.89/7.27	13.41/16.41/17.45
AD-GNN	4.94/5.31/5.67	5.88/6.39/6.78	12.86/15.09/16.56

is, the higher the prediction accuracy. Among them, the predictive performance of the model in this paper is marked in bold, and the predictive performance of the model with the second predictive performance is underlined. As shown in Tables 4, 5 and 6, the AD-GNN model proposed in this paper outperforms other baseline models in predicting traffic flow in the next 15 min, 30 min, and 60 min.

As the prediction period increases, the traffic state becomes more complex, resulting in degraded prediction performance. Compared with other baseline models, the AD-GNN model has a smaller prediction performance decrease as the time span increases and has better time series stability.

5.4 Parameter sensitivity experiment

To further study the influence of the setting of some parameters on the predictive performance of the model, sensitivity analysis of the critical parameters in the model

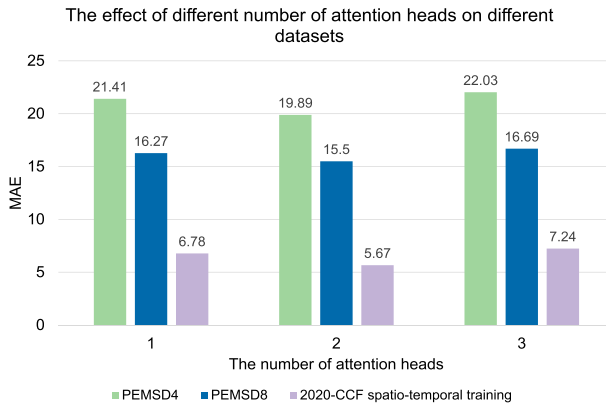


Fig. 5 The effect of different number of attention heads on different datasets

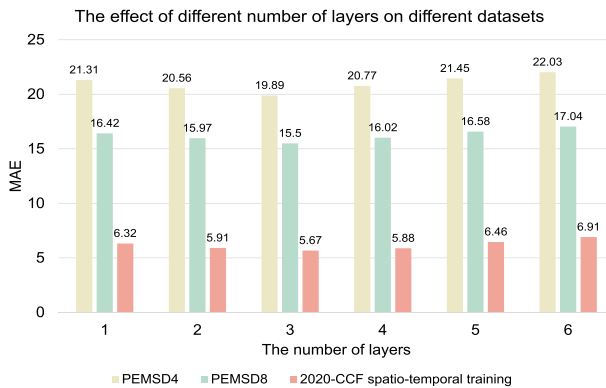


Fig. 6 The effect of different number of layers on different datasets

was carried out in this paper to compare the predictive performance of the model under different conditions. The parameters involved include the number of heads of the attention mechanism and the number of layers of the model, and the corresponding predictive performance is as follows:

Figures 5 and 6 demonstrate the impact of the above parameters on the model's predictive performance. As shown in the figures above, as the number of attention heads increases, the timing information that the attention mechanism can simulate is more abundant, but the module space complexity also increases. The AD-GNN model proposed in this paper achieves the best prediction performance when the number of heads of the temporal attention mechanism is 2. Similarly, when the number of layers is 3, the model's prediction performance reaches the best. The sensitivity analysis of the above parameters proves that the setting of critical parameters in the prediction model is reasonable.

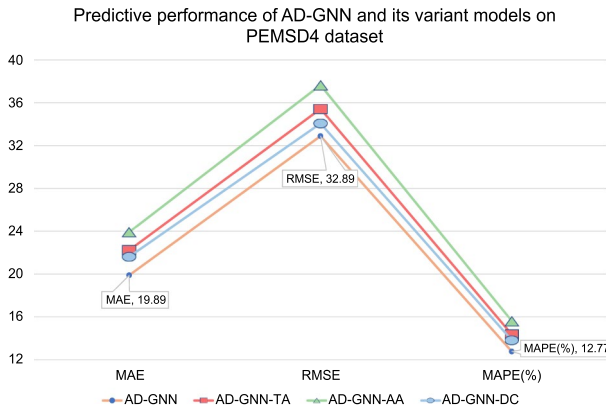


Fig. 7 Predictive performance of AD-GNN and its variant models on PeMSD4 dataset

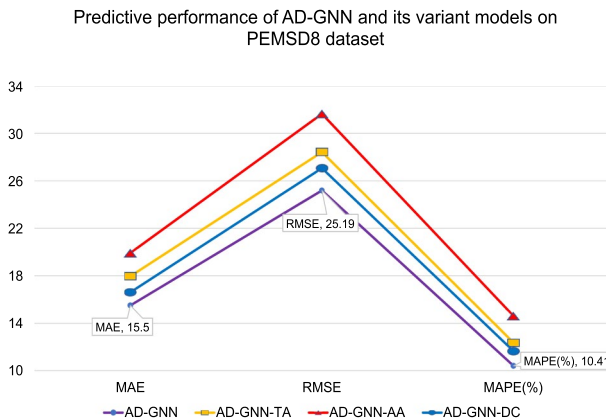


Fig. 8 Predictive performance of AD-GNN and its variant models on PeMSD8 dataset

5.5 Ablation experiment

In order to verify the effectiveness of the different modules in the AD-GNN model, this section removes each part of the network modules in the model through ablation analysis, forms a variant model of the AD-GNN model, and evaluates the prediction performance separately. Verify the necessity of the corresponding module. Ablation experiments include the following variant models:

- (1) AD-GNN-TA: Based on the AD-GNN model, the temporal attention module is removed.
- (2) AD-GNN-AA: Based on the AD-GNN model, the adaptive graph convolution module composed of the parameterized adaptive adjacency matrix is removed.
- (3) AD-GNN-DC: Based on the AD-GNN model, the diffusion convolution module is removed.

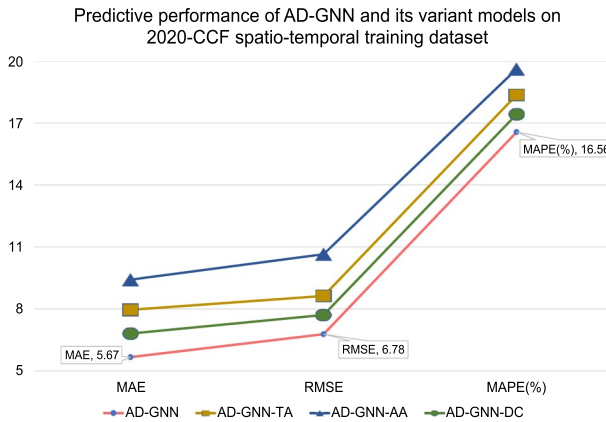


Fig. 9 Predictive performance of AD-GNN and its variant models on 2020-CCF spatiotemporal training dataset

Figures 7, 8 and 9 show the prediction evaluation index values of each variant model proposed in this section on the three datasets. As shown in the figure, the adaptive graph convolution module in the AD-GNN model significantly influences the prediction performance. Based on the data-driven method, this module captures the road network's complex and deep spatial correlations, so the model has a better prediction performance. The temporal attention mechanism can dynamically describe the temporal dependence of each input moment and improve the model's ability to capture temporal dependence. The diffusion graph convolution module introduces road network structure information to supplement the spatial correlation captured by adaptive graph convolution, thus improving the prediction performance.

6 Conclusion and future works

Given the complexity and uncertainty of traffic flow data, this paper proposes an adaptive traffic flow prediction model AD-GNN. It uses the gated time series convolutional network to extract the multi-level time series correlation in the traffic flow data. The graph convolutional networks is also used to capture the spatial associations of road network nodes. The experimental results show that the AD-GNN model has better performance for traffic flow prediction in different periods, and the accuracy is higher than other baseline models; at the same time, with the increase in period, the prediction performance is stable. However, some concerns need to be addressed in the next research work. This paper does not consider the influence of external factors in the traffic scene, such as weather factors and sudden accidents. Furthermore, the prediction model itself also has a certain complexity. In the following work, we will use external information to supplement the traffic flow prediction results, taking into account the relevance of the road network to a greater extent so that the prediction model has better universality.

Appendix A: List of variables and abbreviations

See Table 7.

Table 7 Parameter settings for the baseline models

Parameter	Value
The graph convolution kernel size of STGCN	3
The temporal convolution kernel size of STGCN	3
The recurrent layers of DCRNN	2
The maximum steps of random walks in DCRNN	3
The embedding dimension of AGCRN	10
The number of heads in ASTGNN	8
The max hop in Auto-DSTSG block	2
The layers of AUTO-STG	3
The depths of graph convolution in AUTO-STG	3
The number of vertices in AUTO-STG	2

Author Contributions Tianbo Liu performed the experiment; JZ contributed significantly to analysis and manuscript preparation; TL performed the data analyses and wrote the manuscript; and TL helped perform the analysis with constructive discussions.

Funding This study was supported by the National Key Research and Development Program of China (2017YFB0102500), the National Natural Science Foundation of China (61872158,62172186), the Science and Technology Development Plan Project of Jilin Province (20190701019GH), the Korea Foundation for Advanced Studies' International Scholar Exchange Fellowship for the academic year of 2017-2018, the Fundamental Research Funds for the Chongqing Research Institute, and Jilin University (2021DQ0009).

Data Availability The data and materials used or analyzed during the current study are available from the corresponding author on reasonable request.

Code Availability The codes used or analyzed during the current study are available from the corresponding author on reasonable request.

Declarations

Conflict of interest The authors declare no competing interests

Ethics approval Not applicable.

Consent to participate Informed consent was obtained from all individual participants included in the study.

Consent for publication The participant has consented to the submission of the case report to the journal.

References

1. Tedjopurnomo DA, Bao Z, Zheng B et al (2022) A survey on modern deep neural network for traffic prediction: trends, methods and challenges. *IEEE Trans Knowl Data Eng* 34(4):1544–1561. <https://doi.org/10.1109/TKDE.2020.3001195>
2. Hu C, Fan W, Zeng E et al (2022) Digital twin-assisted real-time traffic data prediction method for 5g-enabled internet of vehicles. *IEEE Trans Industr Inf* 18(4):2811–2819. <https://doi.org/10.1109/TII.2021.3083596>
3. Gong Y, Li Z, Zhang J (2022) Online spatio-temporal crowd flow distribution prediction for complex metro system. *IEEE Trans Knowl Data Eng* 34(2):865–880. <https://doi.org/10.1109/TKDE.2020.2985952>
4. Kang Y, Mao S, Zhang Y (2022) Fractional time-varying grey traffic flow model based on viscoelastic fluid and its application. *Transp Res Part B Methodol* 157(2):149–174. <https://doi.org/10.1016/j.trb.2022.01.007>
5. Li Z, Xiong G, Tian Y (2022) A multi-stream feature fusion approach for traffic prediction. *IEEE Trans Intell Transp Syst* 23(2):1456–1466. <https://doi.org/10.1109/TITS.2020.3026836>
6. Liu D, Baldi S, Yu W et al (2022) On training traffic predictors via broad learning structures: a benchmark study. *IEEE Trans Syst Man Cyber Syst* 52(2):749–758. <https://doi.org/10.1109/TSMC.2020.3006124>
7. Yuan Z, He K, Yang Y (2022) A roadway safety sustainable approach: modeling for real-time traffic crash with limited data and its reliability verification. *J Adv Transp*. <https://doi.org/10.1155/2022/1570521>
8. Chen C, Liu L, Wan S et al (2021) Data dissemination for industry 4.0 applications in internet of vehicles based on short-term traffic prediction. *ACM Trans Internet Technol*. <https://doi.org/10.1145/3430505>
9. Wang J, Chen Q (2020) A traffic prediction model based on multiple factors. *J Supercomput* 77(3):2928–2960. <https://doi.org/10.1007/s11227-020-03373-0>
10. Yang Y, He K, Wang Yp (2022) Identification of dynamic traffic crash risk for cross-area free-ways based on statistical and machine learning methods. *Phys A Statist Mech Appl*. <https://doi.org/10.1016/j.physa.2022.127083>
11. Hajirahimi Z, Khashei M (2022) A novel parallel hybrid model based on series hybrid models of Arima and ANN models. *Neural Process Lett* 54(3):2319–2337. <https://doi.org/10.1007/s11063-021-10732-2>
12. Ermagun A, Ermagun A (2018) Spatiotemporal traffic forecasting: review and proposed directions. *Transp Rev* 38(6):786–814. <https://doi.org/10.1080/01441647.2018.1442887>
13. Javed AR, Rehman SU, Khan MU et al (2021) Canintelliids: detecting in-vehicle intrusion attacks on a controller area network using CNN and attention-based GRU. *IEEE Trans Network Sci Eng* 8(2):1456–1466. <https://doi.org/10.1109/TNSE.2021.3059881>
14. Xu C, Jin X, Wei S et al (2022) Deepst: identifying spatial domains in spatial transcriptomics by deep learning. *Nucleic Acids Res*. <https://doi.org/10.1093/nar/gkac901>
15. Mehdi MZ, Kammoun HM, Benayed NG (2022) Entropy-based traffic flow labeling for CNN-based traffic congestion prediction from meta-parameters. *IEEE Access* 10:16123–16133. <https://doi.org/10.1109/ACCESS.2022.3149059>
16. Saha S, Bovolo F (2022) Change detection in image time-series using unsupervised LSTM. *IEEE Geosci Remote Sens Lett*. <https://doi.org/10.1109/LGRS.2020.3043822>
17. Zhao L, Song Y, Zhang C et al (2021) T-GCN: A temporal graph convolutional network for traffic prediction. *IEEE Trans Intell Transp Syst* 21(9):3848–3858. <https://doi.org/10.1109/TITS.2019.2935152>
18. Pan C, Zhu J, Kong Z et al (2021) Dc-STGCN: Dual-channel based graph convolutional networks for network traffic forecasting. *Electronics*. <https://doi.org/10.3390/electronics10091014>
19. Guo S, Lin Y, Feng N, et al (2019) Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In: *Thirty-Third AAAI Conference on Artificial Intelligence* pp 922–929
20. Mallick T, Balaprakash P, Rask E (2020) Graph-partitioning-based diffusion convolutional recurrent neural network for large-scale traffic forecasting. *Transp Res Rec* 2674(9):473–488. <https://doi.org/10.1177/0361198120930010>

21. Song C, Lin Y, Guo S et al (2020) Spatial-temporal synchronous graph convolutional networks: A new framework for spatial-temporal network data forecasting. In: Thirty-Third AAAI Conference on Artificial Intelligence vol 34, pp 914–921
22. Jin G, Yan C, Liang Z et al (2020) Urban ride-hailing demand prediction with multiple spatio-temporal information fusion network. *Transp Res Part C Emerging Technol.* <https://doi.org/10.1016/j.trc.2020.102665>
23. Zhang Z, Li Y, Song H et al (2021) Multiple dynamic graph based traffic speed prediction method. *Neurocomputing* 461:109–117. <https://doi.org/10.1016/j.neucom.2021.07.052>
24. Guo S, Lin Y, Wang H et al (2022) Learning dynamics and heterogeneity of spatial-temporal graph data for traffic forecasting. *IEEE Trans Knowl Data Eng* 34(11):5415–5428. <https://doi.org/10.1109/TKDE.2021.3056502>
25. Jin G, Li F, Zhang J et al (2022) Automated dilated spatio-temporal synchronous graph modeling for traffic prediction. *IEEE Trans Actions on Intell Trans Syst.* <https://doi.org/10.1109/TITS.2022.3195232>
26. Jin G, Xi Z, Feng Y et al (2022) Deep multi-view graph-based network for citywide ride-hailing demand prediction. *Neurocomputing* 510:79–94. <https://doi.org/10.1016/j.neucom.2022.09.010>
27. Li F, Yan H, Jin G, et al (2022) Automated spatio-temporal synchronous modeling with multiple graphs for traffic prediction. In: *Proceedings of the 31st ACM International Conference on Information and Knowledge Management* pp 1084–1093. <https://doi.org/10.1145/3511808.3557243>
28. Zhang A, Liu Q, Zhang T (2021) Spatial-temporal attention fusion for traffic speed prediction. *Soft Comput* 26(2):695–707. <https://doi.org/10.1007/s00500-021-06521-7>
29. Lv M, Hong Z, Chen L et al (2021) Temporal multi-graph convolutional network for traffic flow prediction. *IEEE Trans Intell Transp Syst* 22(6):3337–3348. <https://doi.org/10.1109/TITS.2020.2983763>
30. Huang Y, Weng Y, Yu S, et al (2019) Diffusion convolutional recurrent neural network with rank influence learning for traffic forecasting. *2019 18th IEEE International Conference on Trust, Security and Privacy in Computing and Communications* pp 678–685. <https://doi.org/10.1109/TrustCom/BigDataSE.2019.00096>
31. Cui Z, Henrickson K, Ke R et al (2020) Traffic graph convolutional recurrent neural network: a deep learning framework for network-scale traffic learning and forecasting. *IEEE Trans Intell Transp Syst* 21(11):4883–4894. <https://doi.org/10.1109/TITS.2019.2950416>
32. Bai L, Yao L, Li C, et al (2020) Adaptive graph convolutional recurrent network for traffic forecasting. In: *34th Conference on Neural Information Processing Systems* 33

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

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.