Information Sciences 607 (2022) 869–883

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Attention based spatiotemporal graph attention networks for traffic flow forecasting

Yi Wang [a], Changfeng Jing [a,*], Shishuo Xu [a], Tao Guo [b]

[a] School of Geomatics and Urban Spatial Informatics, Beijing University of Civil Engineering and Architecture, Beijing 10044, China
[b] Institute of Remote Sensing Application, Sichuan Academy of Agricultural Sciences, Chengdu 610066, China

ARTICLE INFO

Article history:
Received 14 February 2021
Received in revised form 29 May 2022
Accepted 31 May 2022
Available online 4 June 2022

Keywords:
Traffic flow forecasting
Spatiotemporal graph neural network
Network deepening
Network degradation
Dynamic spatiotemporal correlation
Intelligent transportation systems

ABSTRACT

Traffic flow forecasting is a crucial task in transportation and necessary for congestion mitigation, traffic control, and intelligent traffic management. Deep learning models can aid in high-accuracy traffic flow forecasting; however, the current research focuses only the ability of the model to capture dynamic spatiotemporal features, and studies on the effect of deeper network layers on spatiotemporal features—a critical factor affecting traffic flow forecasting accuracy—are limited. In this paper, we propose an attention-based spatiotemporal graph attention network (ASTGAT) model designed for network degradation and over-smoothing problems to investigate in-depth spatiotemporal information. Compared to other networks, ASTGAT can capture dynamic spatiotemporal correlations in data and deepen the network to improve prediction accuracy through multiple residual convolution and high-low feature concat. ASTGAT comprises three components that separately model the temporal relationships of the recent, daily, and weekly periods. Each component stacks multiple spatiotemporal blocks constructed using the attention mechanism, dilated gated convolution, and graph attention network. The graph and temporal attention layers capture spatiotemporal information dynamically, and the graph attention layer alleviates the over-smoothing phenomenon to deepen the network. The combined utilization of the attention mechanism and dilated gated convolution layer improves the medium and long temporal span prediction ability. We validated ASTGAT using two open highway datasets, and the results demonstrated that our ASTGAT model effectively extracts in-depth spatiotemporal information and the prediction results outperform those predicted by the current eight baselines. Our research is dedicated to establishing a better scientific basis for intelligent traffic management that can assist in decision making.

1. Introduction

Intelligent transportation systems (ITS) that can predict people's travel and life requirements intelligently have gained considerable interest in academic and business fields [1]. Traffic forecasting is a popular research topic in this domain, and it helps mitigate traffic congestion, prevent traffic accidents, and effectively manage intelligent traffic infrastructure [2]. However, a major challenge to traffic forecasting is the inherent nonlinearity and the complex spatiotemporal correlation of traffic flow data influenced by both temporal and spatial correlations [3]. Temporal correlations are influenced by traffic

* Corresponding author.
E-mail address: jingcf@bucea.edu.cn (C. Jing).

https://doi.org/10.1016/j.ins.2022.05.127
0020-0255/

conditions at the previous moment or even longer and spatial correlations refer to an interactive dynamic influence; that is, the traffic condition affects the flow up the road segment [3,4]. Traffic flow prediction at a certain location and time is extremely difficult and the changing traffic volume leads to a change in correlation, which further increases the difficulty in achieving accurate predictions. Thus, traffic flow prediction based on spatiotemporal correlation has become a popular research topic in ITS studies.

Technological developments have led to an increase in the number of studies being conducted on flow forecasting. Initially, researchers focused on the temporal correlation of traffic flow data using the historical average (HA) [5], autoregressive integrated moving average (ARIMA) [6], and vector autoregressive (VAR) models [7]. In these studies, traffic flow is predicted under the temporal variation pattern; however, it is difficult to achieve remarkable improvements in the model to obtain a more accurate prediction. With the rapid development of deep learning, long short-term memory network (LSTM) [8] and gated recurrent unit (GRU) [9] become the mainstream prediction models for solving problems with complex assumptions and efficiencies; however, these models focused only on temporal correlations and the results were not very satisfactory. Thereafter, spatial correlation was considered with temporal correlation. Wu and Zhang used convolutional neural networks (CNN) to extract the spatial features of traffic data; however, their results were not outstanding for non-Euclidean data [10]. With the development of graph neural networks (GNNs) [11–13], spatiotemporal GNNs became the mainstream method for traffic prediction [14]. As the network continues to improve, its models become better at capturing spatiotemporal correlations and are being used in several other fields (e.g., PM2.5 [15], crime case prediction [16], and bike-sharing prediction [17]). However, there is still room for improvement in terms of the quality of the captured spatiotemporal information.

Although many existing networks consider spatiotemporal features, it remains difficult to further deepen the network. Spatiotemporal graph neural networks (STGNNs) usually utilize temporal feature extraction (such as recurrent neural network (RNN) or CNN) and spatial feature extraction (such as graph convolutional networks (GCNs) [12]. There are two issues that need to be considered for deepening a STGNN: (1) network degradation and gradient disappearance caused by the RNN and CNN deepening process, and (2) the over-smoothing problem, i.e., features that have the same trend in GCN deepening [18]. The coupled effect of these two problems considerably limits the deepening ability of the STGNNs and the prediction capability. Therefore, an increasing number of novel frameworks have been selected to extract spatiotemporal features from fewer layers, which helps them to avoid the deepening work of networks. However, these frameworks are limited to further enhance forecasting capabilities.

An attention-based spatiotemporal graph attention network (ASTGAT) was proposed to forecast traffic flow at each location of the traffic network to solve these problems. The first "attention" in ASTGAT refers to the temporal attention layer and the second one refers to the graph attention layer. The network can work directly on graph-structured datasets and efficiently extract spatiotemporal information.

The main contributions of this study are as follows.

- A novel framework called ASTGAT that enriches spatiotemporal features by stacking spatiotemporal blocks with different levels so as to deepen the network effectively was proposed.
- A novel spatiotemporal block model and a multicomponent were designed to mitigate the rapid spatiotemporal changes. The former dynamically captures spatiotemporal correlations and the latter focuses on time patterns of different periods.
- A comparison was performed with other baseline methods in two public datasets. The results indicate that the prediction error was reduced by up to 7 % compared to the latest baseline methods. Further, the proposed model mitigates the over-smoothing problem that commonly occurs with STGNNs. The results demonstrated that the upper limit on the number of spatiotemporal blocks that can be effectively stacked in our model exceeds that of other STGNNs.

The remainder of this article is organized as follows: Section 2 presents related work on traffic prediction using the STGNN approach. In Section 3, the novel framework is described, and the design principle and detailed model are introduced. Section 4 details the experimental design, baseline experiments, and results. Section 5 provides discussions on the results. Finally, Section 6 summarizes our work.

2. Related work

2.1. Traffic flow forecasting

Traffic flow forecasting is one of the most challenging difficulties in ITS, and many academic and business studies have been conducted in this domain to solve this challenge. Traffic flow forecasting methods are divided into dynamic modeling and data mining methods. Dynamic modeling approaches use mathematical tools and physical models to simulate traffic dynamics for prediction [19]. These models require complex mathematical formulations and theoretical assumptions, and the traffic flow data cannot satisfy the theoretical distributions or hypotheses for the transportation model. Traffic data are influenced by many factors, which makes it difficult to calibrate the traffic model accurately and use data mining methods to focus on unfolding data patterns of historical data for predicting future data trends. Early data mining models such as

HA [5] are simple, fast, and require no assumptions; however, the accuracy of their prediction is low. Low-accuracy traffic prediction models cannot provide a basis for traffic management.

With the continuous development of data mining methods, a series of higher accuracy prediction models have emerged, and they are divided into parametric and nonparametric models. Parametric models are based on regression models wherein coefficients are determined by processing historical data for traffic flow forecasting. Classical parametric modeling methods include ARIMA [6,20], VAR [7], and the Kalman filter model [21]. These models have advantages of simplicity and low computational resource consumption [4]. However, they depend on stationary assumptions and cannot reflect the nonlinearity and complexity of traffic data and the resistance to scenarios such as unexpected events. The nonparametric model solves these problems by automatically learning these nonlinearities and complexities from the data. Although it requires a large amount of historical data for learning, more intensive studies are focused on these models because of their outstanding performance. Nonparametric models include support vector regression [22], k-nearest neighbor [23], Bayesian network [24], deep learning models, wherein the deep learning model is most widely used for traffic forecasting.

2.2. Euclidean convolutional network for traffic flow forecasting

With the rapid development of deep learning in various fields, an increasing number of researchers have engaged deep learning models in various fields [25,26], especially in transportation systems. The deep belief network can effectively extract high-dimensional features of traffic data to reduce a part of the predicted error [27]; however, it is difficult to extract specific spatiotemporal features from the input data. An RNN can better learn temporal dependence; however, it causes gradient disappearance and gradient explosion because the RNN accumulates errors. LSTM [8] and GRU [9] are variants and improvements of RNN obtained by removing the premature redundant information with gated mechanism. However, they have lower limitations because stacking layers results in degraded performance. A CNN perfectly solves this problem because the stable gradient and low memory requirement are special advantages; however, it requires a stack of multiple layers because of the smaller receptive field. A dilated convolution network (WaveNet [28]) can extract longer sequence information without increasing parameters, and it is more suitable for medium- and long-term traffic prediction. These models consider only temporal features, whereas traffic flow is a type of spatiotemporal data, and thus, the process of prediction cannot naturally ignore spatial dependence. Some CNN-based models that focus on both spatial and temporal correlations have been developed to better represent spatial features. ST-ResNet [10] focused on the temporal closeness, period, and trend of crowd flows. Three residual networks were designed to fuse different features separately and place external factors into the fusion dynamically to consider the prediction results collaboratively. However, CNN cannot act directly on traffic flow data, and therefore, its format needs to be changed. Further, it is difficult to capture the changing features caused by roads and ignore the connection relationship between points. Therefore, although the above method introduces a CNN to capture spatial dependencies, CNN is not suitable for traffic flow data and the captured spatial features remain incomplete.

2.3. Spatiotemporal graph neural network

The GNNs play a crucial role in capturing spatial topological structure information. A GNN with a spectral-domain convolution was first proposed in a spectral graph CNN [11], and its ability to capture graph topological information was verified on the Minist dataset. Chebyshev spectral CNN (ChebNet) [29] was proposed to approximate the variation of the graph Laplacian matrix with Chebyshev polynomials. It unfolds the convolution kernel to eliminate the computational Laplace eigenvector error while addressing the need for a spatially localized convolution kernel. The GCN [12] simplifies the previous approach by approximating the convolution kernels to run in 1st order neighborhoods around each node. The graph attention network (GAT) [13] also plays an important role in capturing structural dependencies. The GAT follows an attentive strategy to compute the hidden value of each node in the graph by aggregating its neighbors, whose presence enables the capture of hard-to-capture spatial features.

GNNs have laid the foundation for the emergence of STGNNs. STGNNs combine temporal convolution and GNNs, considering both time dependence and spatial dependence. With its rapid development, its status has become increasingly crucial in the field of human activity prediction, traffic flow forecasting, and other applications [30]. In the beginning, studies focused on the possibility of capturing data. Yu proposed STGCN [31], which is a network for traffic prediction based on gated CNN and ChebNet. The model effectively captured the overall spatiotemporal correlation to improve the prediction accuracy. In addition to the combination of CNN and GNN, there are networks that combine RNN and GNN with each other. Ling constructed the T-GCN [4] based on GRU and GCN; this model can accomplish the same achievements mentioned above. Zhu proposed AST-GCN [32] based on T-GCN to improve model accuracy precision from the perspective of data fusion, such as weather change and point of interest (POI).

After capturing complete spatiotemporal features, subsequent studies focused on capturing high-quality spatiotemporal features. Wu proposed the problem of long-time prediction, the Graph WaveNet [33], which considers more advanced temporal features. This model makes dilated convolution replace ordinary CNNs and RNNs for traffic flow prediction and improves accuracy. It is a fusion of dilated convolution and GCNs that can ensure each graph convolution layer extracts the spatial dependence information of each node using dilated causal convolution at different fine-grained levels. There are studies that consider optimizing both temporal and spatial features. Zheng proposed GMAN [34], which is designed based on the auto-encoder architecture and attention mechanism, and it can calculate the impact of spatial and temporal

factors on traffic conditions. The transform attention mechanism models the direct relationships between historical and future time steps, which helps alleviate the error propagation problem among prediction time steps. Guo proposed ASTGCN [35], which used the attention mechanism to solve the lack of dynamic spatiotemporal correlation and extract the same regular information from traffic flow data at different times in a multicomponent manner. ASTGCN uses a GCN to capture spatial correlation, and therefore, it is likely to cause Laplace smoothing during superposition. It achieves a good performance in traffic flow forecasting, but has scope for improvement in terms of accuracy. In addition, in order to reflect the dynamic changes of graph structure in traffic flow, Peng proposes a reinforcement learning approach [36] to generate graph structure for dynamic spatiotemporal feature extraction. To make the dynamic change process more interpretable, Huang proposes DSTGNN [37], which utilizes an inhomogeneous Poisson process to characterize the change process of traffic demand thus understanding the underlying mechanism of dynamic change.

3. Attention based spatiotemporal graph attention network

3.1. Preliminaries

In this study, $G(V,E,A)$ denotes the traffic network, where $V \in R^N$ represents the set of vertices, i.e., the number of sensor sites in the traffic network; $E \in R^{N\times N}$ represents the set of edges, i.e., the links between sensor sites in the traffic network; and $A \in R^{N\times N}$ represents the adjacency matrix.

The graph signal matrix is $X_G \in R^{N\times F}$, which represents the F eigenvalues detected at each node on the traffic network G at the time T. Suppose we have i historical time intervals, and each time T has a feature matrix X_t, then, historical data H can be denoted as $H = \{X_{T-i+1}, X_{T-i+2}, \cdots, X_T\}$ and predicted data P can be denoted as $P = \{X_{T+1}, X_{T+2}, \cdots, X_{T+p}\}$.

3.2. Overview of framework

Fig. 1 shows the proposed ASTGAT framework. The framework comprises three independent components with the same structure: recent, daily, and weekly periods. Each component is stacked with the same number of spatiotemporal blocks, which allow us to obtain the same level of spatiotemporal features and the validity of interactions later in the fusion process with the other two components. Finally, three components were fused using a fusion matrix to achieve the final prediction results.

3.3. Multicomponent structure

Let T_0, T_p, and q denote the current time, forecast window size, and the number of samples per day, respectively. We intercept three time series T_h, T_d, and T_w on the time axis as the time component inputs for the recent, daily, and weekly, respectively, where T_h, T_d, and T_w are of the same quantity as T_p. We demonstrated different time periods for the input. The details of these three periods are as follows [35]:

(1) Recent period: This period refers to the historical data near the forecast value, and it is denoted as $X_h = (X_{T_0-T_k+1}, X_{T_0-T_k+2}, \cdots, X_{T_0}) \in R^{N\times F\times T_k}$. Because sudden changes in traffic flow are precursory, the near-moment fragment is the most important for the forecast fragment.

(2) Daily period: This period refers to the historical data of one day ago at the same time as the forecast segment: it is denoted as $X_d = (X_{T_0-q+1}, X_{T_0-q+2}, \cdots, X_{T_0-q+T_p}) \in R^{N\times F\times T_d}$. It is a fragment of the same time interval as the forecast period of the previous day. Because traffic data are likely to show a part of the same pattern over some time, there are morning and evening peaks for each day of a weekday. Therefore, we select this segment as part of the common forecast, while capturing similar features of the daily period.

(3) Weekly period: This period refers to the historical data of a week ago at the same time as the forecast segment, and it is denoted as $X_w = (X_{T_0-7*q+1}, X_{T_0-7*q+2}, \cdots, X_{T_0-7*q+T_p}) \in R^{N\times F\times T_w}$. It is a fragment of the same time interval as the forecast period in the past week. The reason for this is the same as that for the daily period. For example, the flow change of Friday is very similar to that of Friday-one week ago; however, there are some differences with the flow change on the weekend. Therefore, we use it to capture similar features of the weekly period.

The recent period is a common type of historical data used by almost every forecasting framework. However, traffic flow data have different strong periodicities in the time patterns [35], and they are very helpful in improving forecasting accuracy. Daily periods can help us learn differences in traffic variability over time. However, there are still some differences in traffic between weekends and weekdays. To fill this gap, it is necessary to consider weekly changes in traffic. As the most direct and effective method, we use the same multicomponent structure as [35] as that in our framework structure for learning the temporal patterns of different periods, with a 1-day interval as the daily period and a 7-day interval as the weekly period.

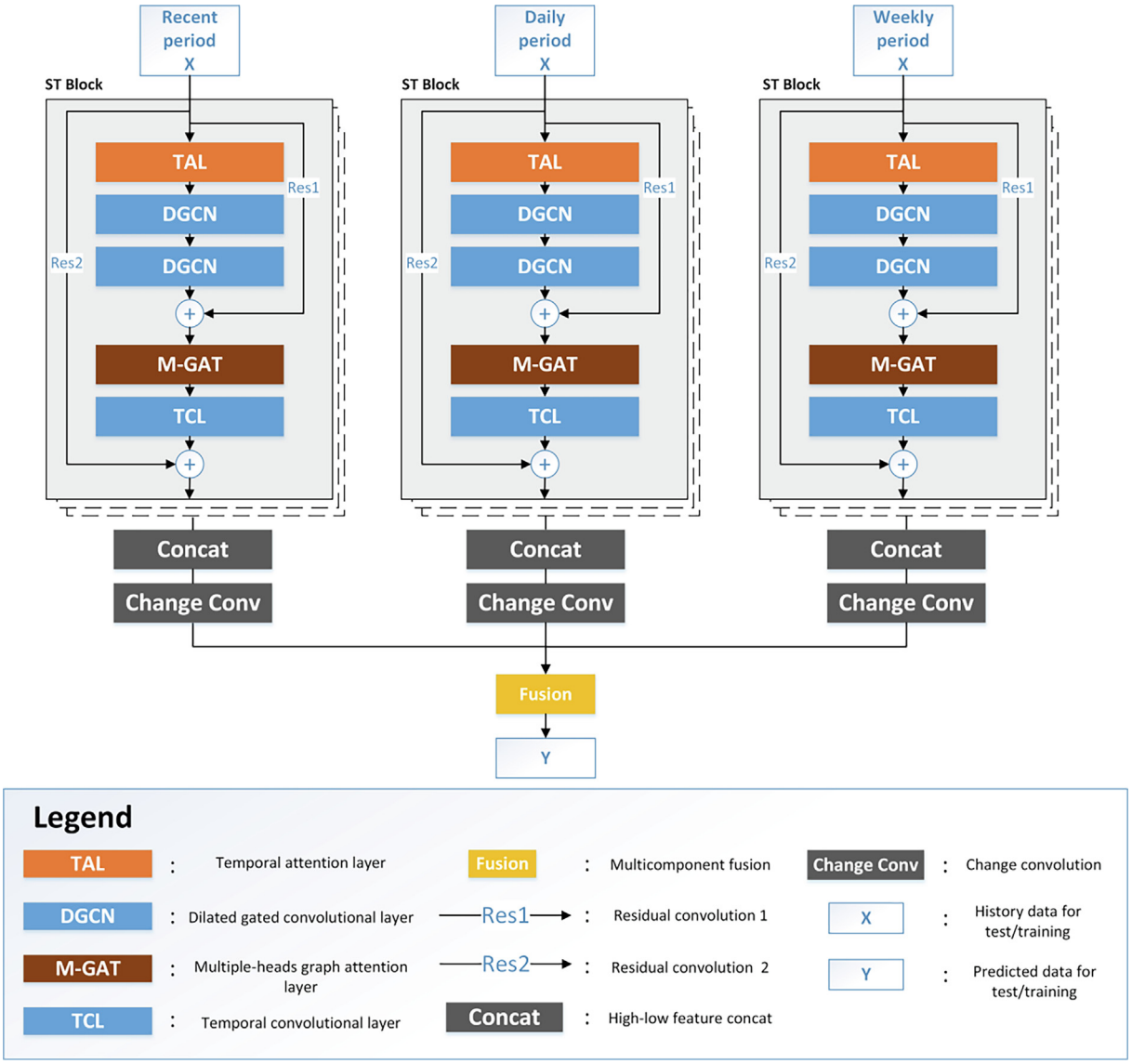

Fig. 1. Framework of attention-based spatiotemporal graph attention networks. The ST Block refers to the spatiotemporal block.

3.4. Spatiotemporal block

(1) Connection approach.

There is irreversible information loss in the process of deepening the network because of the simple stacking method, which causes network degradation and gradient disappearance. Further, GCN can suffer from over-smoothing problems because of single features. Residual structures are commonly used in other STGNNs to solve the first two problems. As shown in the STGCN [31] paper and the ASTGCN [35] public code, the default optimal number of spatiotemporal block stacks is 2. They both employ the residual structure, and the residuals do not allow their networks to continue to deepen. This is largely caused by the lack of richness in the features derived from the residual structure. For solving these problems, we need a stacking approach to make our network both robust against deepening information loss and feature enrichment. The connection of each spatiotemporal block is designed inspired by the skip connection of U-Net [38]; it is called a high-low feature concat. The inner operation process is shown in Fig. 2. The high-low-feature concat allows us to fully enrich the captured spatiotemporal features by integrating different level features. Then, we apply a Change Conv with the parameter $W \in R^{4F \times F\prime}$ (F represents the input feature dimension and F' represents the output feature dimension) to adjust the proportion of high- and low-level features adaptively. This adaptive adjustment allows us to properly deflate and integrate the features of the different blocks in the most direct and simple manner to further ensure the effectiveness of network deepening.

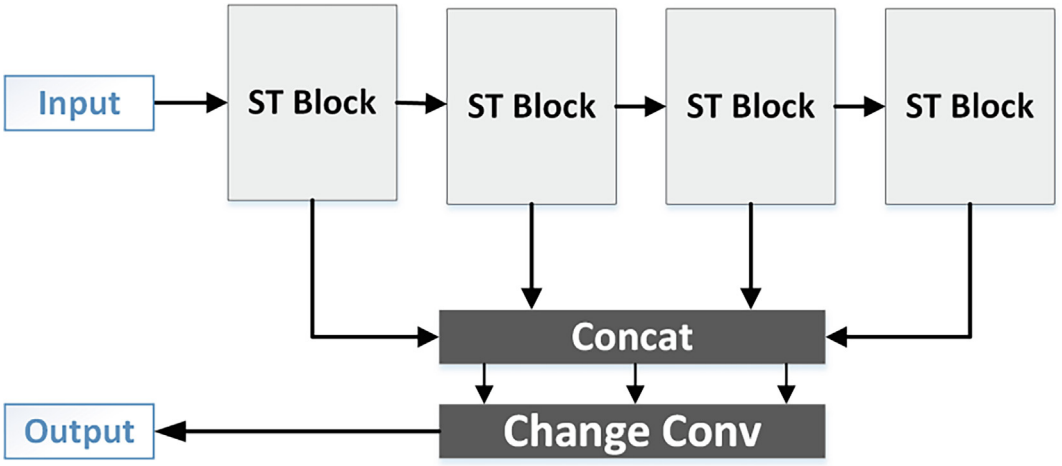

Fig. 2. Connection between each spatiotemporal block (High-low feature concat).

We considered the connection approach between layers in each spatiotemporal block. A spatiotemporal block was designed to capture the features in the time and space of time-series data with a graph structure. Each block comprises of a temporal attention layer, two dilated gated convolution layers (DGCN, where the dilated rates are 2 and 1), a multi-headed graph attention layer, and a temporal convolution layer stacked sequentially. There are two residual convolutions because spatiotemporal block within our framework contains five layers of content that are more complex [39]. The first residual is connected to the output of the second DGCN, and the other residual is connected to the output of the entire spatiotemporal block.

There are two reasons for the spatiotemporal block design. We want to ensure that our spatiotemporal blocks can not only capture dynamic spatiotemporal features similar to other emerging networks, but also deepen them. The GAT and temporal attention layers provided dynamic capture capabilities. We use a double residual convolution to effectively prevent network degradation and gradient disappearance as the network becomes deeper for obtaining spatiotemporal blocks that can be deepened. Further, GAT can alleviate the effect of over-smoothing in the deepening process of the network, and it cannot be alleviated by the GCN. We want to improve our medium- and long-term prediction abilities. The concerted use of attention mechanisms and DGCNs can extract effective spatiotemporal features in a large receptive field. The attention mechanism filters redundant information from different data, thereby allowing the DGCN to expand the receptive field with confidence and use gating to highlight important information once again.

(2) Temporal attention layer.

There is a correlation between changes in traffic flow over a time horizon. Ordinary convolution considers a small number of moment-to-moment correlations and lacks temporal correlation over time. The attention model we apply was used in the field of natural language processing (NLP) to express the degree of association between two words [40]; it is used here to express the importance of any moment over time relative to another moment [35]. Further, the attention model adaptively assigns different weights to the data depending on the data, and finally obtains the dynamic temporal correlation. While obtaining a dynamic temporal correlation over time, it also reduces the portion of noise. We input the data of different categories (total traffic flow, average speed, and average occupancy) together into the attention layer to perform data fusion to extract more reliable attention weights. The attention weight is applied to traffic flow using.

$$\mathrm{E} = V_e \cdot \sigma\left(\left(\left(X_h^r\right)^T U_1\right) U_2 \left(U_3 X_h^r\right) + b_e\right) \tag{1}$$

$$E'_{i,j} = softmax\left(E_{i,j}\right) = \frac{\exp\left(E_{i,j}\right)}{\sum_{j=1}^{T_r} \exp\left(E_{i,j}\right)} \tag{2}$$

where $V_e, b_e \in R^{T_r \times T_r}$, $U_1 \in R^N$, $U_2 \in R^{C_r \times N}$, and $U_3 \in R^{C_r}$ are the parameters to be learned. $X_h^r = \left(X_1, X_2, \cdots, X_{T_{r-1}}\right) \in R^{N \times C_r \times T_r}$ is the output of the r^{th} spatiotemporal block and the input of the r^{th+1} spatiotemporal block. E denotes the number of attention scores at each moment, which vary according to the input data. The value $E_{i,j}$ expresses the correlation between time i and time j and it is normalized in the same way as the normalization of the GATs obtained using the *SoftMax* function to ensure that the weights at each point add to 1. We will apply the temporal attention matrix directly to the input data to dynamically adjust the input information, i.e., that is $\widehat{X}_h^r = \left(\widehat{X}_1, \widehat{X}_2, \cdots, \widehat{X}_{T_r}\right) = \left(X_1, X_2, \cdots, X_{T_r}\right) E' \in R^{N \times C \times T}$.

(3) Dilated-gated convolutional layer.

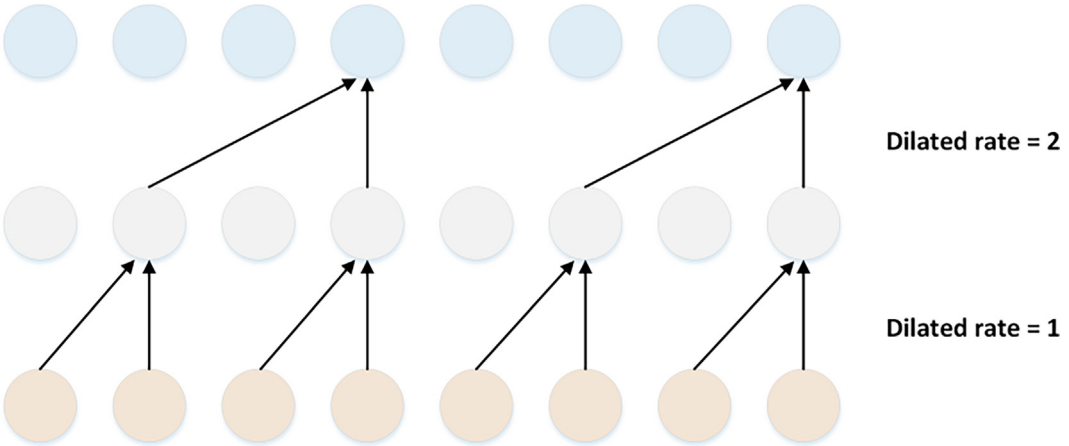

Fig. 3. The process of dilated convolution.

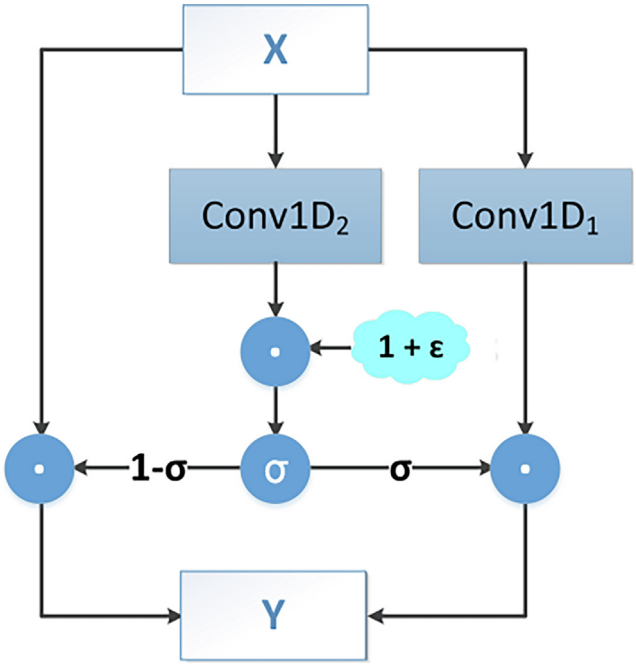

Fig. 4. Structure of dilated gated convolutional layer. ⊙ represent the Hadamard product and σ denotes the sigmoid function that determines the ratio of information passed to the next layer.

All input data contain dynamic spatiotemporal correlations after passing through the temporal attention layer. We used the DGCN to continue to extract deep temporal features with longer receptive fields to better capture the complex temporal dependencies (Fig. 4).

The DGCN is a temporal convolutional layer based on the dilated convolution and gated mechanisms. The DGCN is similar to the GLU function or GTCN of the STGAT [41]. One key component of DGCN is the gated mechanism. Here, each layer contains two convolutional layers with the same setting, and the sigmoid function on one side is applied to other, thereby acting as a gating control to control the change in information. Other important information is highlighted, and the redundant information is reduced by this gate. Further, this model uses a residual structure to prevent gradient disappearance during network deepening while also allowing more information to be transmitted in multiple features. The other key of DGCN is dilated convolution, whose internal operation process is shown in Fig. 3. As the dilated rate increases, the receptive field also increases, and it is useful for long forecasting. However, dilated convolution with same dilated rate is not good practice because it can lead to a "grid effect" and information loss when stacking convolutional layers. The dilated rates of the two DGCNs within a spatiotemporal block were set to 2 and 1 to ensure that each value can participate in the convolution

process and obtain complete temporal information (here, 2 represents the dilated rate of the first DGCN and 1 denotes the second DGCN in each spatiotemporal block).

A regularization term similar to the DropPath mechanism [42] is also used in this study. A uniformly distributed random tensor ε that perturbs the "gate" of the DGCN is defined as $\varepsilon \in [-0.1, 0.1]$. The training results are more stochastic and not over-fit because of the deepened network. This new mechanism makes the model more flexible. Given $\widehat{X}_h^r \in R^{N\times C_r\times T_r}$ as the input of this layer, $\widehat{X}_h^r$ represents the output from the attention layer after the r^{th} layer. The dilated gated convolution can be defined as.

(4) Graph attention layer.

In this study, traffic flow data are essentially a graph structure, and therefore, the features of each node can be considered as signals on the graph. GAT is a very effective method for dealing with graph structure information [13]. Unlike GCN [12] (or ChebNet [29]), GAT is a spatial-domain convolutional network and does not need to be converted to a spectral domain for convolution. Further, it does not rely significantly on graph structures like GCN, but on the magnitude of the graph signal value at each point and the distance between the points to complete the convolution process. Therefore, it can alleviate part of the over-smoothing problem caused by GCN deepening, which can help contribute to the accuracy of predicting peaks and valleys [4]. On the one hand, GAT introduces the possibility of further deepening our model. On the other hand, its attention weights change in real time based on the data changes, which directly adds dynamic spatial correlation to our network without using other approaches.

We sliced the traffic flow at each time point and placed it into the GAT separately. After passing through the graph attention layer, we obtain the information of each node mixed with the information of the neighboring points. We update the information of each time point using a common convolution to merge the information of the neighboring time points. The formula is as given as.

$$e_{ij} = LeakyReLU\left(\overrightarrow{a_i^T}\left(W\overrightarrow{h_i}\|W\overrightarrow{h_j}\right)\right) \tag{5}$$

Our GAT implementation was implemented in the same way as in the original study [13]. The input required in this layer is a set of point features, $\mathrm{h} = \left\{\overrightarrow{h_1}, \overrightarrow{h_2}, \cdots, \overrightarrow{h_N}\right\}, \overrightarrow{h_i} \in R^F$, where N represents the number of nodes in the graph, and F represents the number of features for each node.

In Equation (5), $\|$ denotes the operation in series and $\overrightarrow{a_j} \in R^{2F'}$ is the weight vector to be learned.

$$a_{ij} = softmax(e_{ij}) = \frac{\exp(e_{ij})}{\sum_{k\in N_{(i)}}\exp(e_{ik})} \tag{6}$$

$$h_i' = \sigma\left(\sum_{j\in N_{(i)}} a_{ij}W\overrightarrow{h_i}\right) \tag{7}$$

In Equations (6) and (7), $N_{(i)}$, σ, and h_i' represent the neighboring nodes of node i, nonlinear activation function (in our model, we use the ELU activation function), and features of node i in the next layer.

$$h_i' = \|_{k=1}^K \sigma\left(\sum_{j\in N(i)} a_{ij}^k A^k\overrightarrow{h_j}\right) \tag{8}$$

$$h_i' = \sigma\left(\frac{1}{K}\sum_{k=1}^K\sum_{j\in N(i)} a_{ij}^k A^k\overrightarrow{h_j}\right) \tag{9}$$

where k represents the number off attention heads. We apply multiple attention assignments (i.e., multiple attention heads) to avoid the chance of assigning attention and to ensure the correctness of the weights. Each individual attention head has different parameters, and there are two ways to combine their parameter results: concatenation and averaging. In our study, all GAT layers were aggregated using the average method.

(5) Temporal convolution layer.

After passing through the graph attention layer, we obtain the information of each node mixed with the information of the neighboring points. We update the information of each time point using a common convolution to merge the information of the neighboring time points. This is expressed as.

$$X_h^{r+1} = \sigma\left(WX_h^r + b\right) \tag{10}$$

where $W \in R^{F\times F'}$ and $b \in R^{F'}$ represent the parameters to be learned, X_h^r denotes the information output on the r^{th} layer, and σ represents the activation function (GELU [43] is chosen in this study, but it can also be ReLU).

3.5. Multicomponent fusion

Some locations focus on different time patterns, and we want to enlarge the effect that matches the current location pattern component. For example, the time pattern for 8:00 a.m. on Sunday of each week is very similar to that of this Saturday and the previous Sunday at this moment; the traffic forecast for this moment needs consider more daily and weekly period outputs. However, there are some time points where there is no obvious period pattern, and therefore, more values of the proximity period need to be considered. Thus, these three components have different degrees of influence on the fusion process, and therefore, we use a tensor to define the fusion matrix $\widetilde{W}$ that can autonomously learn the component weights from the data adaptively to achieve more accurate forecasting. We defined the fusion process as follows:

$$\widetilde{W} = (W_h, W_d, W_w) \tag{11}$$

$$\hat{Y} = \widetilde{W}\left(\hat{Y}_h, \hat{Y}_d, \hat{Y}_w\right)^T = W_h \odot \hat{Y}_h + W_d \odot \hat{Y}_d + W_w \odot \hat{Y}_w \tag{12}$$

where $\odot$ denotes the Hadamard product; $\widetilde{W} \in R^{3N\times T}$ denotes the parameter fusion matrix which consists of three parts; $W_h, W_d, W_w \in R^{N\times T}$, where N represents the number of nodes and T represents the number of predicted time steps. These parameters can be learned autonomously in the matrix, and they influence the weights of different locations at different moments in the three components.

4. Experiment design

Two open traffic datasets were selected for traffic flow forecasting for evaluating the performance of the proposed model.

4.1. Dataset

We applied two highway traffic datasets obtained from California (PeMSD4 and PeMSD8) to validate our model. The data were collected every 30 s in real time by the Caltrans performance measurement system (PeMS) [44]. Traffic data were aggregated every 5 min from the raw data, thereby obtaining a total of 288 data points per day.

PeMSD4 is the traffic data for the San Francisco Bay area from January to February 2018. PeMSD8 is the traffic data of San Bernardino from July to August 2016. Both datasets contain three traffic metrics: total traffic, average speed, and average occupancy. The detailed information is summarized in Table 1.

We ensure that the distance between each adjacent detector is longer than 3.5 miles to avoid unnecessary calculations, and therefore, discarded some redundant detectors. We performed linear interpolation of the missing values of the data to fill them. The data were normalized $X' = X - mean(X)$ so that all data values were between [0,1], and the mean value of the data set was 0.

4.2. Experimental setting

In our model, all graph attention layers and temporal convolution layers use 64 channels. The dilated rate of DGCN in each temporal block was 2, 1, and two DGCN layers were superimposed. The size of the prediction window, T_p = 12, indicates that we predict the traffic flow within one hour. The longer the historical data, the smaller periods it contains, and the better is the traffic forecasting. However, we keep the historical span and forecast span in the same way as in the other models to ensure the fairness of the experiment and verify the forecasting ability gap seen between these models. For each of the three spatiotemporal blocks, we considered 12 historical data: T_h = 12, T_d = 12, and T_w = 12. During the training process, we used the Adam optimizer to train the model with a learning rate of 0.0001. Attention dropout is applied using 0.6, and the number of attention heads is 8 in the graph attention layer. In addition, the loss function of the model uses L2Loss, and the initialization method is XavierNorm. The evaluation metrics follow ASTGCN and include the mean absolute error (MAE) and root mean square error (RMSE).

Table 1
Detailed information on PeMSD4 and PeMSD8.

Dataset	PeMSD4	PeMSD8
Data type	Traffic flow	Traffic flow
Nodes(Sensors)	307	170
Edges	341	295
Time steps	16,992	17,856
Features	3	3
Data frequency	5 min	5 min

$$\mathrm{MAE} = \frac{1}{m}\sum_{i}\|x_i - \hat{x}_i\| \tag{13}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{m}\sum_{i}\left(x_i - \hat{x}_i\right)^2} \tag{14}$$

4.3. Baselines

We used eight mainstream methods to compared and validate the performance of our model. Our implementation of ASTGAT and ASTGCN(12) was based on the Pytorch 1.7.2, and they were trained and evaluated on a single NVIDIA Tesla V100 with 16 GB memory. Other results were referenced from the ASTGCN [35].

- **HA** [5]: We use the average of the 12 most recent moments of the time slice to predict the next value.
- **ARIMA** [6]: It is one of the most classical methods for forecasting the analysis of time-series data.
- **VAR** [7]: A relatively advanced time-series model that regresses all current period variables in the model on several lags of all variables.
- **LSTM** [8]: A special RNN model with three types of gating.
- **GRU** [9]: A special RNN model with two gates. Compared with LSTM, one gating was simplified.
- **STGCN** [31]: The spatiotemporal graph convolutional network which has a sandwich-like structure and performs traffic flow prediction by stacking spatiotemporal blocks. It completely applies convolutional layers to construct temporal blocks instead of applying recursive units, which improves the training speed.
- **GeoMan** [45]: A multilevel attention-based recurrent neural network that employs a multilevel attention mechanism for the dynamic spatiotemporal dependence model. It is a generic fusion module that incorporates multiple factors from different domains.
- **ASTGCN(12)** [35]: The attention based spatiotemporal graph convolutional network which uses a temporal attention mechanism and spatial attention to capture dynamic correlations in time and space. ASTGCN integrates three different components to model the periodicity of road traffic data. We reproduce the publicly available code of ASTGCN, and the remaining parameters are set as in the original paper except for the time span. Two layers of spatiotemporal blocks were stacked with a batch size of 64 and a learning rate of 0.0001 for training. During the training process the L2loss function is applied, and the Adam optimizer is applied for learning. To ensure the fairness of the experiment, we selected 12 time points as the input for each component.

5. Result and discussion

5.1. Results

We applied the network to two datasets (PeMSD4 and PeMSD8) for validation and metric evaluation using RMSE and MAE. As indicated in Table 2, our model is more effective than the results of all other reference methods on average. Compared with the best baseline method, our method reduced RMSE by approximately 7.4 % and MAE by approximately 5.8 % on PeMSD4; and RMSE by 1.8 % and MAE by about 1.5 % on PeMSD8. These results reveals that our ASTGAT model is outstanding for traffic flow forecasting.

We analyze the differences between our network and other networks from following three perspectives.

(1) Prediction precision.

Table 2 indicates that the predictive ability of traditional models such as HA, ARIMA, and VAR is not ideal. HA, ARIMA, and VAR need to make certain necessary assumptions that may not be satisfied in the actual road conditions, they achieve poor performance. For the average metrics, all deep learning methods are better than traditional methods, which indicates that deep learning can better cope with the nonlinearity and complexity of traffic data. However, considering only temporal feature deep learning methods such as LSTM and GRU, this improvement is not significant. Therefore, STGCN, GeoMan, ASTGCN (12), and our model considered the effects of changes in spatial features on traffic flow forecasting. Among these, GeoMan and ASTGCN consider the dynamic spatiotemporal changes through the attention mechanism, and therefore, they are more

Table 2
Average performance comparison of different methods on PeMSD4 and PeMSD8.

Dataset	Evaluation Metrics	HA	ARIMA	VAR	LSTM	GRU	STGCN	GeoMan	ASTGCN(12)	ASTGAT
PeMSD4	RMSE	54.14	68.13	51.73	45.82	45.11	38.41	37.84	36.88	**34.33**
	MAE	36.76	32.11	33.76	29.45	28.65	27.28	23.64	23.29	**22.02**
PeMSD8	RMSE	44.03	43.30	31.21	36.96	35.95	30.78	28.91	27.35	**26.87**
	MAE	29.52	24.04	21.41	23.18	22.20	20.99	17.84	17.95	**17.57**

effective than STGCN. Our network accomplishes dynamic spatiotemporal capture through the temporal attention layer and GAT. Further, our model considers the capture of deep spatiotemporal features, which helps achieve better accuracy than these two methods.

(2) Deep spatiotemporal feature capture.

STGCN, GeoMan, and ASTGCN(12) consider spatiotemporal features, but they ignore the advanced level of spatiotemporal features. From their original papers [31,35,45] or publicly available default codes, we learned that their models reached the optimal solution by stacking two layers of spatiotemporal blocks. Stacking more layers worsens performance because of network degradation. There are two reasons for this: (a). The third layer starts to degrade gradually because of the over-smoothing phenomenon [18] in the case of the simple stacking of GCN. (b). Deepening the network in the case of simple stacking or single residuals can lead to less than optimal results because of the extremely homogeneous spatiotemporal features. Therefore, we select GAT, double residual convolution, and high-low level feature concat as components of our network, which allow our network to be deepened further. Table 3 indicates that we substantially exceeded the results of the other baselines which considered spatial features within 30 min. Between 30 and 60 min, our method also exceeded most baseline results. This validates our design idea that deep spatiotemporal features are significant for traffic flow forecasting.

(3) Medium and long-term forecasting capability.

Fig. 5 shows that the prediction effectiveness of each method exhibits different decreasing trends with an increase in the prediction time. The HA and ARIMA methods are considerably effective for short-term forecasting; however, their forecasting effectiveness decreases significantly with longer forecasting times. In the case of LSTM and GRU, which are time series prediction methods, achieve better results than the traditional methods, but their prediction results are not satisfactory because the prediction time span increases owing to the lack of spatial correlation. STGCN, GeoMan, and ASTGCN consider both the temporal dependence and spatial dependence of traffic data, and thus, their long-term prediction is better than the deep learning models that only consider time. GeoMan, constructed using the attention mechanism, has a very smooth error curve, which proves the attention mechanism has a non-negligible contribution to traffic flow prediction. Although ASTGCN reduces the historical time span, the prediction results are still satisfactory compared with other models. In this case, our model introduces the attention mechanism and DGCN to aggregate the effective temporal information of a longer time span, and thus, the prediction effect is significantly better that those of the other methods. Compared with most models, the proposed model also achieves some improvement in 45–60 min; and the accuracy improvement around 30–45 min is more prominent compared to that obtained with GeoMan and ASTGCN.

In terms of data, all methods performed worse on PeMSD4 than on PeMSD8. Compared to PeMSD4, the performance enhancement between all models was smaller on the PeMSD8 dataset. The San Francisco Bay Area (PeMSD4 region), which is the second largest metropolitan area in California and second only to the Greater Los Angeles area in terms of population, has more complex traffic conditions. Therefore, forecasting using PeMSD4 is more challenging. After validation on PeMSD4, we found that the prediction error of ASTGAT was substantially reduced compared to that of the baseline model. This indicates that our model can handle more complex scenarios.

Table 3
Performance comparison of the performance of different methods on PeMSD4 and PeMSD8 for different periods.

Dataset	Model	15 min		30 min		45 min		60 min	
		RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
PeMSD4	HA	44.84	30.06	51.92	35.14	**	**	**	**
	ARIMA	43.35	24.24	69.54	30.56	**	**	**	**
	VAR	49.85	32.45	52.06	33.96	53.59	35.11	55.04	36.04
	LSTM	36.99	22.85	44.80	29.13	51.61	34.18	60.02	39.44
	GRU	38.30	23.56	43.80	27.89	54.81	34.43	55.94	36.90
	STGCN	35.59	25.11	37.67	26.66	40.29	28.73	43.53	31.30
	GeoMan	37.26	22.88	37.62	23.37	38.17	24.19	38.93	**24.98**
	ASTGCN(12)	33.92	21.59	36.61	23.30	38.87	24.70	41.58	26.58
	ASTGAT	**31.56**	**20.14**	**33.98**	**21.77**	**36.26**	**23.45**	**38.86**	25.48
PeMSD8	HA	35.40	23.86	41.49	28.09	48.56	33.30	**	**
	ARIMA	29.49	18.24	39.55	22.96	50.50	28.21	**	**
	VAR	26.91	18.43	31.36	21.42	34.79	23.86	37.38	25.86
	LSTM	28.18	17.41	36.93	23.21	43.24	27.01	50.01	33.61
	GRU	28.73	16.57	35.88	22.35	41.42	26.33	50.01	32.07
	STGCN	28.94	19.72	30.43	20.65	32.05	21.88	34.17	23.49
	GeoMan	27.80	17.40	28.70	17.69	29.68	**18.12**	30.71	**18.77**
	ASTGCN(12)	25.56	16.85	27.37	17.88	28.51	18.70	29.55	20.29
	ASTGAT	**24.70**	**16.09**	**26.72**	**17.45**	**28.42**	18.69	**29.45**	20.29

"**" means that the values are too large to be negligible, which indicates that the prediction effect of the model is poor.

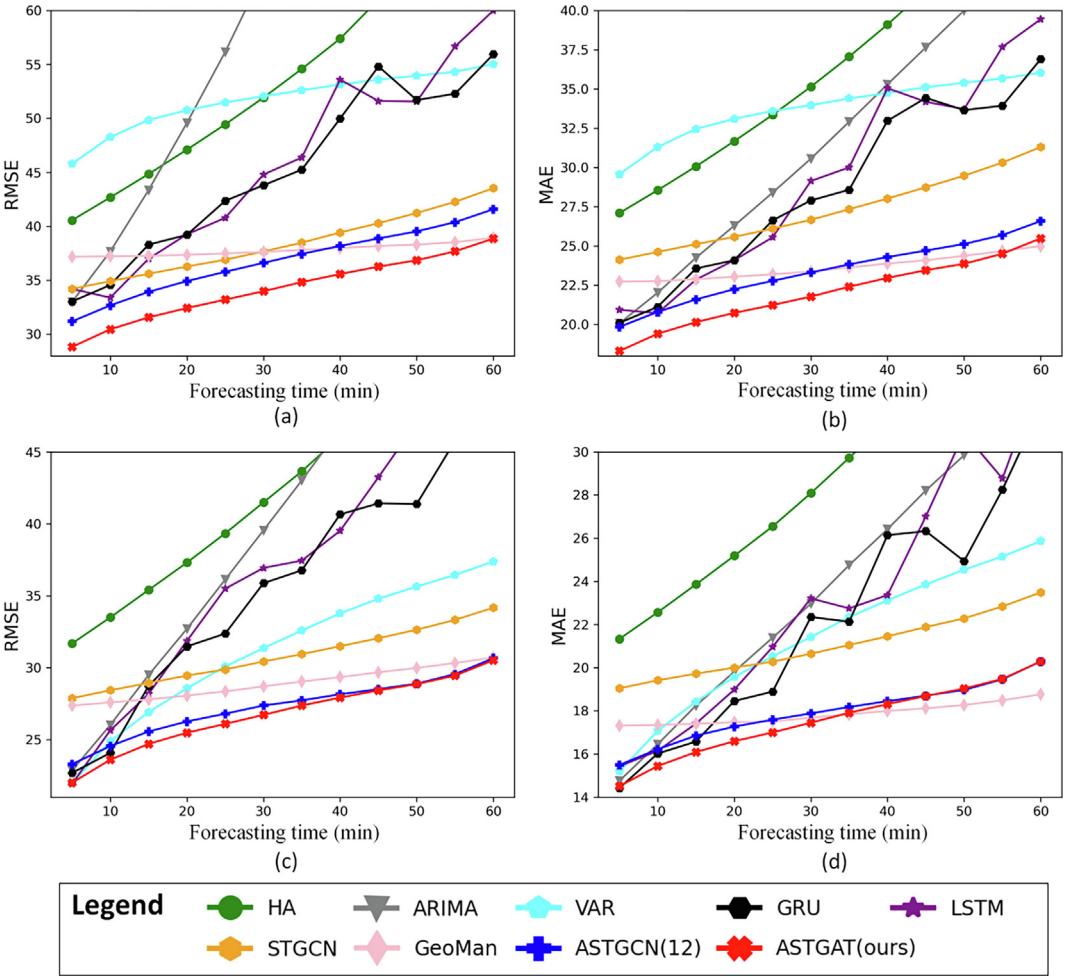

Fig. 5. Predictive performance visualization for different periods: (a) RMSE-PeMSD4, (b) MAE-PeMSD4, (c) RMSE-PeMSD8, (d) MAE-PeMSD8.

5.2. Discussion

(1) Influence of model components on prediction accuracy.

We performed some experiments on ASTGAT, including the stacking of different numbers of blocks, replacement of activation functions, and addition or absence of the DropPath mechanism. We control the variables to investigate the effect of different configurations on our model. These are defined as variants of our model; the detailed information is as follows:

- Two blocks (ReLU): We stack two spatiotemporal blocks, and all activation functions within the spatiotemporal blocks (except GAT) use ReLU as the activation function. In addition, there was no DropPath added to DGCN.
- Two blocks (GELU): Replace all ReLU functions in the spatiotemporal block with GELU functions.
- Three blocks (GELU): Stack one more spatiotemporal block on the base of the two Blocks (GELU).
- Four blocks (GELU): Stack one more spatiotemporal block on the base of the three Blocks (GELU).
- Four blocks (GELU + DropPath): Add DropPath to all DGCNs in the four spatiotemporal blocks based on the four blocks (GELU). This is our ASTGAT.

We analyzed these abovementioned variants with the overall forecasting ability. Fig. 6 shows the comparison results for two blocks (GELU), three blocks (GELU) and four blocks (GELU); the results indicate that the predicted effect improves as the number of blocks increases, which proves that our deeper network is effective. As the depth increases within a certain range, the deep spatiotemporal feature information is more suitable for traffic flow forecasting. In addition, the GELU [43] function is used in Bert [46] in the field of NLP. Compared to the ReLU function, the GELU function solves the problem of gradient explosion and the problem that most parameters will sometime not be updated. The comparison of two blocks (GELU) and two blocks (ReLU) shows that the improved effect of changing the activation function is approximately 1 %. We found

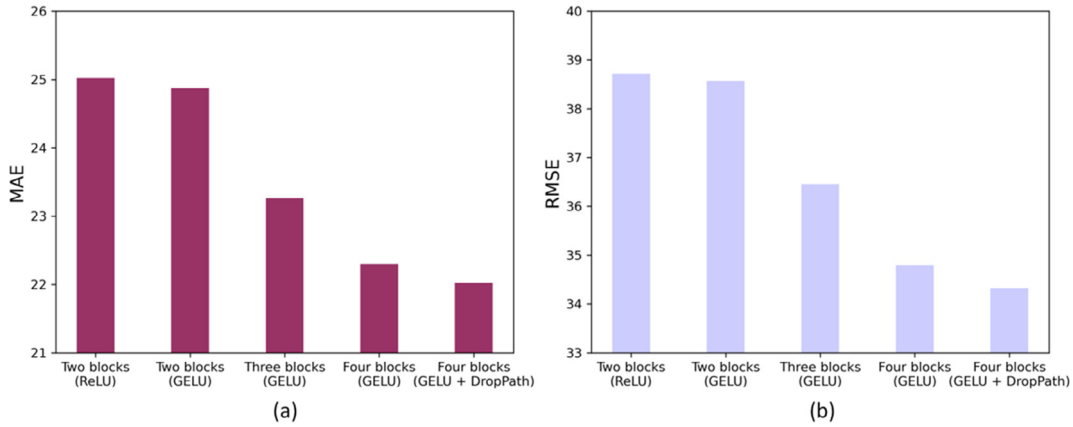

Fig. 6. Performance of different configurations of ASTGAT on PeMSD4, (a) MAE (b) RMSE.

that the accuracy was improved approximately 2 % after adding the DropPath mechanism when comparing four blocks (GELU) with four blocks (GELU + DropPath). This proves that the DropPath can introduce more randomness to the model, which ensures that the depth is increased with less overfitting; this improves the prediction accuracy of ASTGAT.

The following conclusions were obtained by analyzing those variants from different time spans, as indicated in Fig. 7. On comparing two blocks (GELU), three blocks (GELU) and four blocks (GELU), we find that the curve becomes flatter as the number of stacked spatiotemporal blocks increases, especially in the medium and long term. This proves that the extracted deep spatiotemporal features have greater gains in the medium and long term. This is because long-term forecasting is more difficult than short-term forecasting, which requires more parameters to fit. The short term does not require a fitting approach as complex, and therefore, the improvement is not very large. We analyzed above that the activation function replacement will introduce a small amount of improvement. In these two figures comparing two blocks (GELU) and two blocks (ReLU), we observe that the gains from the GELU function are in the long-term prediction. Further, the DropPath represents overall instead of local when comparing four blocks (GELU) with four blocks (GELU + DropPath) because of the increased stochasticity, which allows the model to better resist overfitting. The overfitting phenomenon is independent of the time span, and thus, its improvement is more uniform across time spans.

(2) Forecasting capability over different spans.

We visualized the prediction of node 201 in the PeMSD4 dataset to better understand the forecasting effect of the ASTGAT model. The following four sets of plots show the visualization results for the 15, 30, 45, and 60 min predicted horizons. These results are shown in Fig. 8 reveal the following:

- Fig. 8 (a) and (b) shows that our network effect of traffic flow forecasting is very good in the short term (less than or equal to 30 min), where the predicted value fits perfectly with the ground truth. This can substantially facilitate travel by enabling people to know the exact amount of traffic at a destination within 30 min.

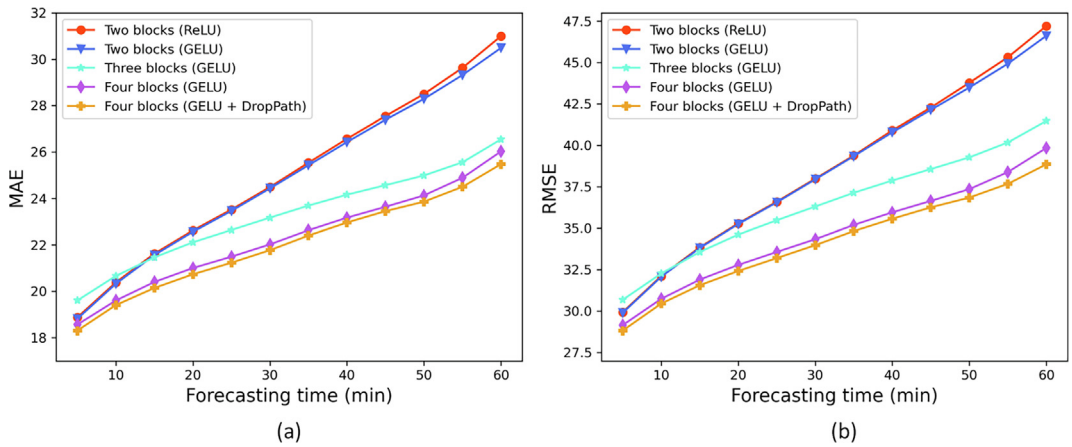

Fig. 7. Performance of different variants under different time spans on PeMSD4, (a) MAE and (b) RMSE.

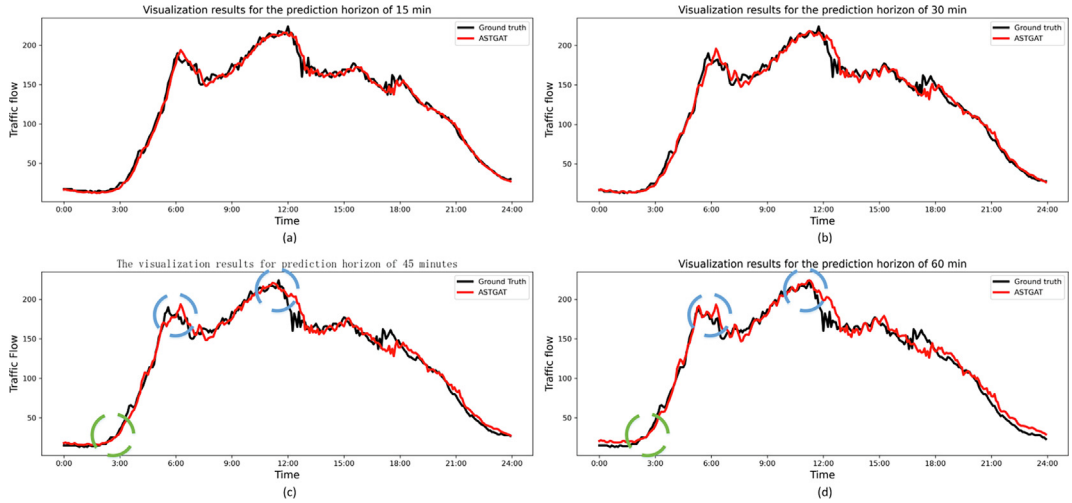

Fig. 8. Visualization results of prediction horizon for different periods, (a) 15 min, (b) 30 min, (c) 45 min, (d) 60 min prediction horizon for 1 days.

- The construction principle of the GCN network will lead to a gap between the predicted and actual peaks [4]. Our high-low feature concat, double residual convolution, and GAT developed their role, which effectively solves the peak over-smoothing phenomenon caused by the smoothing filter [47] defined in the GCN model. Therefore, the peak and valley in the prediction are close to the actual values. From Fig. 8 (c) and (d), we see that there is a good fit for the positions of blue dashed (peak value) and green dashed circle (valley value) which means that our model performs well in the predicting peaks and valleys even within a 45-or 60-min time span. This demonstrates the ability of our model to predict longer spans of congestion levels at times of traffic congestion that provide a more reliable basis for traffic management and evacuation.

6. Conclusions

A new deepened spatiotemporal graph neural network model (ASTGAT) was proposed and used for traffic flow prediction. This model uses a graph attention layer and a temporal attention layer to solve the problem of dynamic spatiotemporal information capture. Double residual convolution and high-low feature concat were developed to solve network degradation and over-smoothing problems when deepening the network. Thus, our network can further deepen the network based on dynamic spatiotemporal information capture. The experimental results obtained with two real-world traffic datasets demonstrated that ASTGAT can obtain a higher prediction accuracy compared with those of the baseline models. This work can improve prediction accuracy and be extended to other domains. However, the influence of multi-scale information for traffic flow forecasting is yet to be validated, and it will be covered in a future work.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work was supported by the Beijing Natural Science Foundation (8222009), the Training Program for Talents by Xicheng, Beijing (202137), and the Pyramid Talent Training Project of the Beijing University of Civil Engineering and Architecture (JDJQ20200306).

References

[1] N. Buch, S.A. Velastin, J. Orwell, A review of computer vision techniques for the analysis of urban traffic, IEEE Trans. Intell. Transp. Syst. 12 (3) (2011) 920–939.
[2] J. Zhang, F.-Y. Wang, K. Wang, W.-H. Lin, X. Xu, C. Chen, Data-driven intelligent transportation systems: A survey, IEEE Trans. Intell. Transp. Syst. 12 (4) (2011) 1624–1639.
[3] C. Song, Y. Lin, S. Guo, H. Wan, Spatial-temporal synchronous graph convolutional networks: a new framework for spatial-temporal network data forecasting, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2020, pp. 914–921.
[4] L. Zhao, Y. Song, C. Zhang, Y.u. Liu, P.u. Wang, T. Lin, M. Deng, H. Li, T-gcn: A temporal graph convolutional network for traffic prediction, IEEE Trans. Intell. Transp. Syst. 21 (9) (2020) 3848–3858.

[5] J. Liu, W. Guan, A summary of traffic flow forecasting methods, J. Highway Transport. Res. Dev. 3 (2004) 82–85.
[6] B.M. Williams, L.A. Hoel, Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: Theoretical basis and empirical results, J. Transp. Eng. 129 (6) (2003) 664–672.
[7] E. Zivot, J. Wang, Vector autoregressive models for multivariate time series, Modeling financial time series with S-PLUS®, (2006) 385-429.
[8] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.
[9] J. Chung, C. Gulcehre, K. Cho, Y. Bengio, Empirical evaluation of gated recurrent neural networks on sequence modeling, in: NIPS 2014 Workshop on Deep Learning, December 2014, 2014.
[10] J. Zhang, Y. Zheng, D. Qi, Deep spatio-temporal residual networks for citywide crowd flows prediction, in, in: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, 2017, pp. 1655–1661.
[11] J. Bruna, W. Zaremba, A. Szlam, Y. LeCun, Spectral networks and locally connected networks on graphs, in: 2nd International Conference on Learning Representations, Canada, 2013.
[12] T.N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, in: 5th International Conference on Learning Representations, OpenReview.net, Toulon, France, 2017.
[13] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, Y. Bengio, Graph Attention Networks, in: International Conference on Learning Representations, 2018.
[14] J. Ye, J. Zhao, K. Ye, C. Xu, How to build a graph-based deep learning architecture in traffic domain: A survey, IEEE Trans. Intell. Transp. Syst. (2020).
[15] H. Zhou, F. Zhang, Z. Du, R. Liu, Forecasting PM2. 5 using hybrid graph convolution-based model considering dynamic wind-field to offer the benefit of spatial interpretability, Environ. Pollut. 273 (2021) 116473, https://doi.org/10.1016/j.envpol.2021.116473.
[16] Y. Zhang, T. Cheng, U. Systems, Graph deep learning model for network-based predictive hotspot mapping of sparse spatio-temporal events, Comput. Environ. 79 (2020) 101403.
[17] W. Zi, W. Xiong, H. Chen, L. Chen, TAGCN: Station-level demand prediction for bike-sharing system via a temporal attention graph convolution network, Inf. Sci. 561 (2021) 274–285.
[18] Q. Li, Z. Han, X.-M. Wu, Deeper insights into graph convolutional networks for semi-supervised learning, in: proceedings of the thirty-second AAAI conference on artificial intelligence (AAAI-18), in: Association for the Advancement of Artificial Intelligence, 2018, pp. 3538–3545.
[19] X.-Y. Xu, J. Liu, H.-Y. Li, J.-Q. Hu, Analysis of subway station capacity with the use of queueing theory, Transportation research part C: emerging technologies 38 (2014) 28–43.
[20] M.S. Ahmed, A.R. Cook, Analysis of freeway traffic time-series data by using Box-Jenkins techniques, 1979.
[21] I. Okutani, Y.J. Stephanedes, Dynamic prediction of traffic volume through Kalman filtering theory, Transportation Research Part B: Method. 18 (1) (1984) 1–11.
[22] H. Drucker, C.J. Burges, L. Kaufman, A. Smola, V. Vapnik, Support vector regression machines, Adv. Neural Inform. Process. Systems 9 (1996) 155–161.
[23] X.-L. Zhang, G.-G. He, H.-P. Lu, Short-term traffic flow forecasting based on K-nearest neighbors non-parametric regression, J. Syst. Eng. 24 (2009) 178–183.
[24] S. Sun, C. Zhang, G. Yu, A Bayesian network approach to traffic flow forecasting, IEEE Trans. Intell. Transp. Syst. 7 (1) (2006) 124–132.
[25] B. Yi, X. Shen, H. Liu, Z. Zhang, W. Zhang, S. Liu, N. Xiong, Deep matrix factorization with implicit feedback embedding for recommendation system, IEEE Trans. Ind. Inf. 15 (8) (2019) 4591–4601.
[26] J. Sun, X. Wang, N. Xiong, J. Shao, Learning sparse representation with variational auto-encoder for anomaly detection, IEEE Access 6 (2018) 33353–33361.
[27] W. Huang, G. Song, H. Hong, K. Xie, Deep architecture for traffic flow prediction: deep belief networks with multitask learning, IEEE Trans. Intell. Transp. Syst. 15 (5) (2014) 2191–2201.
[28] D. Impedovo, V. Dentamaro, G. Pirlo, L. Sarcinella, TrafficWave: Generative deep learning architecture for vehicular traffic flow prediction, Appl. Sci. 9 (24) (2019) 5504, https://doi.org/10.3390/app9245504.
[29] M. Defferrard, X. Bresson, P. Vandergheynst, Convolutional neural networks on graphs with fast localized spectral filtering, Adv. Neural Inform. Process. Syst. 29 (2016) 3844–3852.
[30] S. Zhang, H. Tong, J. Xu, R. Maciejewski, Graph convolutional networks: a comprehensive review, Comput. Soc. Networks 6 (2019) 1–23.
[31] B. Yu, H. Yin, Z. Zhu, Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting, in: Proceedings of the 27th International Joint Conference on Artificial Intelligence, 2018, pp. 3634–3640.
[32] J. Zhu, Q. Wang, C. Tao, H. Deng, L. Zhao, H. Li, AST-GCN: attribute-augmented spatiotemporal graph convolutional network for traffic forecasting, IEEE Access 9 (2021) 35973–35983.
[33] Z. Wu, S. Pan, G. Long, J. Jiang, C. Zhang, in: Graph Wavenet for Deep Spatial-temporal Graph Modeling, in, AAAI Press, 2019, pp. 1907–1913.
[34] C. Zheng, X. Fan, C. Wang, J. Qi, Gman, A graph multi-attention network for traffic prediction, in: In: Proceedings of the AAAI Conference on Artificial Intelligence, 2020, pp. 1234–1241.
[35] S. Guo, Y. Lin, N. Feng, C. Song, H. Wan, Attention based spatial-temporal graph convolutional networks for traffic flow forecasting, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2019, pp. 922-929.
[36] F. Huang, P. Yi, J. Wang, M. Li, J. Peng, X. Xiong, A dynamical spatial-temporal graph neural network for traffic demand prediction, Inf. Sci. 594 (2022) 286–304.
[37] H. Peng, B. Du, M. Liu, M. Liu, S. Ji, S. Wang, X. Zhang, L. He, Dynamic graph convolutional network for long-term traffic flow prediction with reinforcement learning, Inf. Sci. 578 (2021) 401–416.
[38] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, in: International Conference on Medical image computing and computer-assisted intervention, Springer, 2015, pp. 234-241.
[39] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.
[40] X. Feng, J. Guo, B. Qin, T. Liu, Y. Liu, Effective deep memory networks for distant supervised relation extraction, in: In: Proceedings of the 26th International Joint Conference on Artificial Intelligence, 2017, pp. 4002–4008.
[41] X. Kong, W. Xing, X. Wei, P. Bao, J. Zhang, W. Lu, STGAT: spatial-temporal graph attention networks for traffic flow forecasting, IEEE Access 8 (2020) 134363–134372.
[42] G. Larsson, M. Maire, G. Shakhnarovich, Fractalnet: Ultra-deep neural networks without residuals, in: 5th International Conference on Learning Representations, OpenReview.net, Toulon, France, 2016.
[43] D. Hendrycks, K. Gimpel, Gaussian error linear units (gelus), arXiv preprint arXiv:.08415, (2016).
[44] C. Chen, K. Petty, A. Skabardonis, P. Varaiya, Z. Jia, Freeway performance measurement system: mining loop detector data, Transp. Res. Rec. 1748 (1) (2001) 96–102.
[45] Y. Liang, S. Ke, J. Zhang, X. Yi, Y. Zheng, GeoMAN: multi-level attention networks for geo-sensory time series prediction, in: Proceedings of the 27th International Joint Conference on Artificial Intelligence, 2018, pp. 3428–3434.
[46] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT, Pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), 2019, pp. 4171–4186.
[47] J. Chen, Y. Wang, M. Zeng, Z. Xiang, Y. Ren, Graph Attention Networks with LSTM-based Path Reweighting, arXiv preprint arXiv:.10866, (2021).