Meas. Sci. Technol. 36 (2025) 066114 (15pp) https://doi.org/10.1088/1361-6501/add955

A deep graph convolution spatial-temporal attention learning model for traffic flow prediction

Liming Jiang, Baiyi Liu, Youfu Jiang, Shaomiao Chen, Huanyu Wang*[*] and Wei Liang

School of Computer Science and Engineering, Hunan University of Science and Technology, Xiangtan 411100, People's Republic of China

E-mail: huanyu@hnust.edu.cn

Received 1 March 2025, revised 10 May 2025
Accepted for publication 15 May 2025
Published 27 May 2025

Abstract

As an effective means to solve the challenges of online measurement of difficult-to-measure variables in complicated traffic processes, deep learning–based traffic flow prediction has emerged as one of the primary research objectives in the field of soft measurement with its strong data feature extraction ability. However, most of the existing traffic flow prediction methods follow a stacked structure of spatial-temporal blocks to capture traffic flow features, which encounter limitations in solving the over-smoothing problem caused by the deep-stacked graph convolution network (GCN), as well as the interaction effects endured by the spatial-temporal attention learning based on cascaded structure. In this paper, we adopt a sequential structure to connect the spatial module, the temporal module, and the attention module to achieve the spatial-temporal correlations learning. Along this line, we propose a deep graph convolution spatial-temporal attention learning network, which considers both large range spatial dependence and global joint spatial-temporal correlation, to predict traffic flows. In particular, a deep stacked GCN module is adopted in our model to capture multiscale spatial features of the traffic flow data, which are constructed by leveraging a combination of residual networks, jump connections and multilayer perceptrons. Afterward, the temporal features are learned using the gated temporal convolution. Finally, a spatial-temporal attention mechanism is introduced to simultaneously capture dynamic global correlations in both spatial and temporal dimensions. Our experimental results, based on four real-world traffic flow datasets called PeMSD3, PeMSD4, PeMSD7, and PeMSD8, reveal that the prediction accuracy of the proposed model outperforms other baseline models. Moreover, it can efficiently alleviate over-smoothing effects while maintaining manageable computational overhead.

Keywords: intelligent transportation, traffic prediction, graph convolutional networks, gated temporal convolution, attention mechanism

1. Introduction

Recent advancements in Internet of Things (IoT) technology and ubiquitous smart sensor deployments have revolutionized data acquisition in intelligent transportation systems (ITS). Researchers and practitioners can now capture multisource heterogeneous traffic data via diverse measurement modalities, providing data support for traffic analysis. However, the difficulties of traffic information measurement are reflected in two aspects. First, due to communication failures, sensor failures, routine maintenance, and malicious destruction, some

[*] Author to whom any correspondence should be addressed.

measurement results may have random missing points and blocks. Second, there are some traffic status parameters that cannot be measured by sensors at present, such as traffic congestion status and traffic safety risks. Although offline analysis can be applied to measure these parameters that cannot be directly measured, this will also cause lag in measurement, making it strenuous for the results to meet the online monitoring, optimization, and control needs of ITSs. In recent years, soft measurement methods based on deep learning have been widely applied in multivariate time series and traffic flow prediction [1–6].

Unlike multivariate time series, traffic flow prediction is extremely challenging due to the complicated spatial-temporal dependence and essential uncertainty induced by the irregular underlying road network and temporal-varying traffic conditions. In the temporal dimension, the traffic flow state in a certain road segment exhibits strong temporal-varying characteristics. In terms of spatial dimension, there is an apparent local-range spatial dependence between direct adjacent road segments within the road network. There is a long-distance spatial dependence between nondirect connected segments. How to effectively capture the spatial-temporal features and correlations from these complex and nonlinear spatial-temporal road sensors' data is crucial for effective traffic flow prediction modeling.

Existing traffic prediction models can be roughly divided into traditional methods and deep learning methods. Traditional statistical or machine learning methods, such as auto-regressive integrated moving average (ARIMA) [7], lack the ability to process high-dimensional spatial data and cannot learn sophisticated nonlinear information from traffic data. However, with the development of deep learning, these problems can be solved gradually [8]. Zhao *et al* adopt the long short-term memory (LSTM) network to model the temporal information in traffic data for achieving short-term traffic speed and traffic flow predictions, but it ignores the influence of spatial correlation on traffic prediction [9]. Convolutional LSTM network (ConvLSTM) [10] and deep spatial-temporal residual network (ST-ResNet) [11] can simultaneously model spatial and temporal correlations by combining convolutional neural network (CNN) and LSTM to achieve accurate traffic flow prediction. However, since CNNs cannot maintain translation invariance on the data of non-Euclidean structures, this makes CNNs unable to process traffic flow data under irregular road network structures. In this case, a graph convolution network (GCN) emerged and was subsequently applied to traffic flow prediction because of its excellent ability to process graph data. This leads to a series of spatial-temporal graph convolutional network (T-GCN) prediction models with good performance. For example, STGCN [12] adopts stacked spatial-temporal convolutional blocks to model spatial-temporal correlations, but it captures spatial correlations by using static predefined graphs. Another group of researchers proposed adaptive GCN for spatial feature learning based on data-adaptive dynamic graph learning for reducing the model's dependence on predefined graphs, such as DSTAGNN [13] and MTGNN [14]. However, these models have certain limitations. First, these models usually adopt a shallow stack of graph convolutional layers to learn the local spatial correlations between traffic network nodes, limiting their ability to learn a large range of spatial dependencies. This is attributed to the challenges in training network models with a more extensive and deeper architecture, often leading to the occurrence of over-smoothing, which results in a rapid degradation of prediction performance [15]. With the increasing number of GCN layers, the continuous multiplication effect in the gradient back propagation makes the update of the weight parameters unstable, which may lead to gradient disappearance, gradient explosion, and weight matrix degradation. In contrast, the aggregation range of nodes also expands and the output characteristics of each node tend to converge to the same value, resulting in over-smoothing problems. Second, existing methods usually use either a single spatial attention mechanism or a two-stage attention scheme divided by temporal and spatial. The former method cannot capture the dynamic global temporal correlation in traffic flow characteristics. The latter is usually constructed through a cascade of spatial and temporal attention, which allows spatial attention to interfere with temporal attention learning to a certain extent. Consequently, this setup can have a certain degree of influence on the accuracy of attention learning.

To address these issues, we propose a deep graph convolution spatial-temporal attention learning network (DGSTAN) for more accurate traffic flow prediction. Our model first captures spatial dependence and temporal correlation characteristics through a deep stacked residual graph convolutional network with gated linear units. Afterward, it adopts a spatial-temporal attention mechanism to enhance the dynamic spatial-temporal correlation learning and improve the accuracy and effectiveness of the model in predicting multiple temporal horizons.

The main contributions of this article are listed as follows:

- We propose a DGSTAN model. The proposed model incorporates residual networks, skip connections, and multilayer perceptron (MLP) to capture spatial dependencies of the range of neighbors with different hops. In addition, it utilizes gated linear units and one-dimensional convolution to learn temporal-based correlation characteristics.
- We design a joint spatial-temporal attention mechanism dedicated to traffic flow data processing. We demonstrate that our approach can effectively assess the significance of temporal channels and their corresponding spatial data, both independently and concurrently. This allows the model to capture the dynamic correlation between different road segments and between different time steps of the same road segment.
- We evaluate the model's performance on four real traffic datasets and the experimental results show that our method outperforms other benchmark models in both prediction accuracy and long-term traffic flow prediction performance.

The structure of this paper is detailed as follows. In section 2, we briefly review the existing research methods for traffic prediction models and discuss their dependence on spatial-temporal modeling. In section 3, we discuss the relevant concepts and definitions of multistep urban traffic flow prediction. Then, we introduce the proposed DGSTAN model and its components in detail in section 4. In section 5, we perform extensive experiments on real-world datasets and an in-depth analysis of the model. Finally, we conclude this paper and present the potential future work in section 6.

2. Related works

Researchers have conducted detailed research on traffic flow prediction using traditional statistical methods, machine learning, deep learning, and other predictive modeling methods. It was initially regarded as a simple time series prediction problem by researchers, without considering the spatial features. Some common conventional machine learning models are the vector auto-regressive (VAR) model [16], ARIMA [7], support vector machine (SVM) [17], and Kalman filtering [18]. All the mentioned approaches need to assume that the traffic flow sequence data meet certain preconditions, which impose limitations on the ability to extract complex nonlinear features of traffic flow data [19], consequently resulting in low prediction accuracy. Nonparametric models are proposed to address traffic flow nonlinearity, such as K-nearest neighbor (KNN) [20] and support vector regression (SVR) [21]. Due to the lack of consideration of spatial-temporal correlations in traffic flow data and the ability to process high-dimensional data, machine learning methods face more challenges in dealing with traffic flow data with complex spatial-temporal correlations.

Deep learning has achieved great success in computer vision [22, 23], speech recognition [24], natural language processing [25], and other fields in the past two decades. To effectively extract the dynamic, nonlinear, and time series characteristics of traffic flow data, researchers use recurrent neural networks (RNNs), LSTM and gate recurrent unit (GRU) [26, 27] to conduct traffic flow prediction modeling and put forward a series of prediction models. These models can effectively extract the time series characteristics of traffic flow data, but ignore the learning of spatial correlations in traffic flow data. Zheng *et al* [28] proposed a Conv–LSTM model constructed from the CNN and the LSTM network, which can extract spatial-temporal features more efficiently. However, there are still difficulties in capturing long-term temporal dependence and may suffer from performance degradation due to increased sequence length. In addition, a deep multiview spatial-temporal network (DMVST-Net) can effectively solve this problem [29]. It uses a transformer [30] to solve long-term temporal dependence and employs a local CNN to capture local features of regions with respect to their neighbors. Although the above methods have the ability of spatial feature learning, converting traffic flow data into grid data can still result in the loss of spatial correlation information, which inevitably leads to an increase in prediction loss.

In recent years, graph representation and convolutional networks have been widely used in traffic prediction, and numerous research results have been obtained. Zhao *et al* proposed a T-GCN model for traffic data prediction, called T-GCN [31], which combines the graph convolutional network and the gated recursive unit. The former is used to learn sophisticated topological structures to capture spatial dependencies through shallow stacking of GCNs, and the latter is used to learn dynamic changes in traffic data to capture temporal dependencies. Compared with GRU's iterative processing of time series data, the gate linear unit (GLU) uses one-dimensional convolution to directly process the data in the time window in parallel, which greatly expedites the operation while ensuring the learning performance of temporal characteristics. This increases the broad applications of GLU in the modeling of spatial-temporal characteristics of traffic flow data. For example, the spatio-T-GCNs (STGCN) [12] proposed by Yu *et al* uses a GLU to process the temporal correlation of traffic data. However, the above methods usually use separate temporal and spatial convolution components to capture spatial and temporal correlations, ignoring heterogeneity in spatial-temporal data. Spatio-temporal synchronous graph convolutional network (STSGCN) [32] captures complex local spatial-temporal correlations via a spatial-temporal synchronization modeling mechanism. It also designs several modules with different temporal periods to capture the heterogeneity in the local spatial-temporal graph. Bai *et al* [33] proposes an adaptive graph convolutional recurrent network (AGCRN), which can capture node-specific spatial and temporal correlations for traffic prediction without a predefined graph. Multivariate correlation-aware STGCNs (MC-STGCN) [34] can construct a coarse-grained road graph based on topology closeness and traffic flow similarity. It employs a cross-scale spatial-temporal feature learning and fusion technique to handle both fine- and coarse-grained traffic data. Liu and Meidani [35, 36] developed an end-to-end model based on heterogeneous graph neural networks that effectively resolves the user equilibrium traffic allocation problem.

To better exploit dynamic spatial-temporal correlations in traffic data, researchers have introduced attention mechanisms into traffic prediction modeling. [37, 38] applied Graph Attention Network (GAT) to spatial correlation feature learning among road network sensors using a multiheaded attention mechanism to dynamically aggregate first-order neighbor information. However, GAT ignores the dynamic changes in the temporal relevance. Another research [39] proposed a dynamic adaptive deeper STGCNs (ASTGCNs), which uses a channel attention mechanism to calculate the dynamic weights of traffic features at different time steps to mine the temporal relevance of traffic data, but neglect the dynamic changes in the spatial relevance of traffic data. The spatial-temporal adaptive gated GCN (STAG-GCN) [40] constructed the road network as a dynamic weighted graph via an attention mechanism. It utilizes a multivariate self-attention temporal convolution network to capture temporal dependencies. Liu and Meidani [41] proposed a

Table 1. Summary of representative graph neural networks for traffic prediction.

Approach	Spatial module	Temporal module
DCRNN [42]	Spatial GNN	Recurrence
STGCN [12]	Spectral GNN	Convolution
Graph WaveNet [43]	Spatial GNN	Convolution
MTGNN [32]	Spatial GNN	Convolution
STSGCN [14]	Spatial GNN	Convolution
StemGNN [44]	Spectral GNN	Convolution(Frequency)
AGCRN [33]	Spatial GNN	Recurrence
STFGNN [45]	Spatial GNN	Convolution
DSTAGNN [13]	Spectral GNN	Attention+Convolution
STEP [46]	Spatial GNN	Attention

novel multiview heterogeneous graph neural network (M-HetGAT) framework that effectively resolves the multiclass traffic assignment problem using virtual links and an adaptive graph attention mechanism. The methodological framework of these models encompasses multivariate spatial interaction analysis and temporal dynamics characterization, with architectural implementations comparatively summarized in table 1.

Although these studies have effectively promoted the development of traffic flow data prediction modeling theory, through careful analysis, it is not arduous to find that the prediction model is difficult to learn the long-distance spatial dependence due to the limitations of the GCN itself. Meanwhile, existing attention networks rarely expand the modeling ability of dynamic spatial-temporal correlation from the perspective of spatial-temporal dimension attention mechanism. In view of these issues, this paper proposes a multiscale residual graph convolution module based on residual network and jump connection method to solve the multiscale feature spatial relationship learning of depth GCN. Combining temporal convolution and spatial-temporal attention mechanism to enhance the learning ability of dynamic deep spatial-temporal features of the model, the problem of dynamic spatial-temporal dependence in traffic flow prediction is effectively solved.

3. Preliminaries

Traffic flow prediction uses historical traffic flow information from a road network to predict the future traffic flow. The following are some concepts and definitions related to traffic flow prediction. Table 2 summarizes the primary notations mentioned later.

Definition 1 (traffic network graph). We use an undirected graph $G = (V, E, A)$ to describe the topological structure of traffic networks. Each road sensor in a road network is regarded as a node, where $V = \{v_1, v_2, \ldots, v_N\}$ is a set of road nodes, N is the number of nodes, and E is a set of edges. We use the adjacency matrix $A \in R^{N\times N}$ to represent the spatial correlation between the road nodes.

Definition 2 (traffic flow feature). The traffic flow features (e.g., speed or traffic flow) of the road nodes V_i at time t is represented as $x_t^i \in R^C$, where C represents the node traffic flow feature dimension. The traffic flow features of all road nodes at time t can be expressed as $x_t = \left(x_t^1, x_t^2, \ldots, x_t^{N-1}, x_t^N\right) \in R^{N\times C}$, and the historical traffic flow feature of all nodes for the past $T(T = 1, 2, 3, \ldots)$ time steps can be denoted as $X^{(t-T+1):t} \in R^{N\times T\times C}$. Similarly, the prediction result of the traffic flow feature for all nodes in the next M time steps can be expressed as $\widehat{X}^{(t+1):(t+M)} \in R^{N\times M\times C}$.

Definition 3 (traffic flow prediction). Traffic flow prediction problem aims to learning the function $f(\cdot)$ to graph the historical traffic flow features of the past T time steps to the traffic flow features in the subsequent one or multiple time steps. In this study, the function $f(\cdot)$ attempts to prediction the traffic flow features in the next M time steps, formally expressed as

$$\left[X^{(t-T+1):t}, G\right] \xrightarrow{f} \left[\widehat{X}^{(t+1):(t+M)}\right]. \qquad (1)$$

4. Methodology

Figure 1 shows the general framework of our proposed DGSTAN model for traffic flow prediction. It comprises four major parts, which are used to model the multiscale spatial characteristics, temporal characteristics, and global dynamic spatial-temporal correlation of learning prediction targets. Multiscale residual graph convolution solves the trainability problem of deep stack graph neural networks through residual connections, and aggregates the multiscale feature output of each GCN layer through jump connections, enhancing the node's representation learning ability with a broad range of spatial neighborhood nodes. The multiscale gated temporal convolution module captures the temporal correlation of traffic flow data with different scale feature expressions through one-dimensional convolution and GLU. The spatial-temporal attention module calculates the weight parameters from the spatial and the temporal dimensions, respectively, to dynamically adjust the correlation intensity between different time steps and nodes. The prediction output is obtained through a fully connected layer.

Table 2. Notations.

Notation	Description
G	Traffic network graph
D	Degree matrix
A	Adjacency matrix
$\tilde{A}$	Normalized laplace matrix
Θ	Set of all trainable parameters
X	Input data
$\widehat{X}$	prediction result
$H^{(\ell)}$	GCN block output without residual connection
$\widetilde{H}^{(\ell)}$	GCN block output With residual connection
H_{Cat}	Concatenation feature
H_{AGG}	Aggregation feature
H_{TC}	Output features of gated temporal convolution module
att_T	Temporal attention weight
H_{att}^{T}	Temporal attention features
att_S	Spatial attention weight
H_{att}^{S}	Spatial attention features
H_{att}	Output features of spatial-temporal attention module

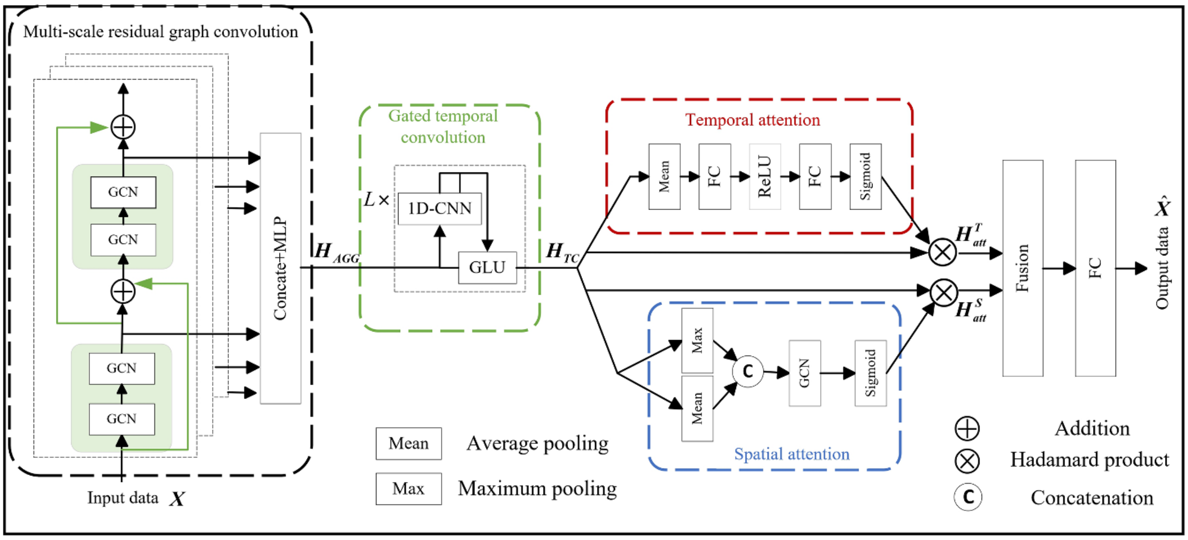


Figure 1. The framework of DGSTAN.

4.1. Multiscale residual graph convolution

The road node can aggregate representations of 1-hop neighbors by a graph convolution layer. We expanded the receptive neighborhood range by stacked multiple graph convolution layers. In particular, the output of GCN blocks at different levels corresponds to the aggregation of the characteristics of neighboring nodes from different spatial ranges. However, multiple graph convolution layers cannot adaptive select the appropriate neighbor range for each node, and also face over-smoothing problem. Therefore, we construct a residual graph convolution module by integrating the residual network, jump connection, and MLP to realize multiscale spatial feature learning. The detailed structure is depicted in figure 1. In our framework, the multiscale residual graph convolution module comprises K blocks and the output of the $\ell(1 \leqslant \ell \leqslant K)$ block is expressed as follows:

$$H^{(\ell)} = \sigma\left(\widehat{A}\left(\mathrm{relu}\left(\widehat{A}\widetilde{H}^{(\ell-1)}W_1^{\ell}\right)\right)W_2^{\ell}\right) \tag{2}$$

where $\widehat{A} = \widetilde{D}^{-\frac{1}{2}}\widetilde{A}\widetilde{D}^{\frac{1}{2}}, \widetilde{A} = A + I_n, \widetilde{D} = \sum_j \widetilde{A}_j$. In (2), A represents the adjacency matrix, I_n is the identity matrix, D is the degree matrix, $H^{(\ell)}$ represents the output of GCN block ℓ, $\widetilde{H}^{(\ell-1)}$ represents the output of GCN block $\ell - 1$ with residual connection, w_1^{ℓ}, w_2^{ℓ} is the learnable parameter, and $\sigma(\cdot)$ and $\mathrm{relu}(\cdot)$ are the activation function.

Afterward, we construct a residual connection between GCN blocks so that the output of each block is passed to the next block through a jump connection. Therefore, the output of each block is correlated to the representation of the node under different scales. Hence, the corresponding residual block output can be formalized as follows:

$$\widetilde{H}^{(\ell)} = H^{(\ell-1)} + \alpha_\ell * \widetilde{H}^{(\ell-2)} \tag{3}$$

where α_ℓ is the learnable parameter. When $\ell = 1, \widetilde{H}^{(1)} = H^{(\ell)} + \alpha_1 * X$.

We use concatenation and aggregation operations to effectively merge the output features of all graph convolutional blocks to suppress the occurrence of over-smoothing problems. First, the output of each GCN block is concatenated, making full use of the node representation at different scales:

$$H_{\text{Cat}} = \text{Concat}\left(H^{(1)}, H^{(2)}, \ldots, H^{(K)}\right). \tag{4}$$

Next, the MLP structure is used to perform a nonlinear transformation of the concatenation representation, and the output of the multiscale residual graph convolution module, denoted as H_{AGG}, is obtained. We formulate the calculation process as follows:

$$H_{\text{AGG}} = MLP\left(H_{\text{Cat}}\right) = W_{a2} * \sigma\left(W_{a1} * H_{Cat} + b_{a1}\right) + b_{a2} \tag{5}$$

where $W_{a1}, W_{a2}, b_{a1}, b_{a2}$ are learnable parameters.

4.2. Gated temporal convolution

The gated temporal convolution module comprises L gated temporal convolution layers stacked together. The gated temporal convolution layer uses 1D-Conv and linear gated units to extract the dynamic temporal features of traffic flow. We used the first gated temporal convolution layer as an example to give the calculation process of its output. $H_{\text{AGG}} \in R^{N\times T\times C}$ is the input of the first temporal convolution layer, and the 1D-Conv operation with a prepadding function is applied to H_{AGG} with convolution kernel $\Gamma \in R^{K_t\times C\times 2C}$ to obtain $\widetilde{H}_{\text{AGG}} \in R^{N\times(T-K_t+2P+1)\times 2C}$. We first divide $\widetilde{H}_{AGG}$ into two tensors $P \in R^{N\times(T-K_t+2P+1)\times C}$ and $Q \in R^{N\times(T-K_t+2P+1)\times C}$. Next, we calculate the output $H_{TC}^{(1)}$ of the first gated temporal convolution layer through the formula as follows:

$$H_{TC}^{(1)} = P \circ \sigma(Q) \in R^{N\times(T-K_t+1+2P)\times C} \tag{6}$$

where C is the number of input channels, $\circ$ represents the Hadamard product, $\sigma(\cdot)$ represents the activation function, K_t is the width of the convolution kernel, and P is the padding amplitude. When $P = \frac{K_t-1}{2}$, $(T-K_t+2P+1) = T$. This ensures that the input and output of the temporal convolution layer remain invariant across the temporal dimension T.

Based on the calculation process of the gated temporal convolution described above, the output of the Lth gated temporal convolution layer $H_{TC}^{(L)}$ represents the output of the entire gated temporal convolution module H_{TC}. It can be seen that by setting the appropriate padding amplitude and convolution kernel size, the gated temporal convolution module can extract the temporal correlations while ensuring that the length of the output feature tensor H_{TC} in the temporal dimension is consistent with its input feature tensor H_{AGG}.

4.3. Spatial-temporal attention

4.3.1. Temporal attention. Inspired by the correlations calculation method between the feature channels in SENet [47], Zhao *et al* proposed a traffic prediction model ADSTGCN [48] with a dynamic adjustment module, which divides channels in the temporal dimension, where one time step is one channel. The purpose is to dynamically adjust the spatial–temporal correlations by assigning dynamic weights to the features at different time steps. Following these works, we introduced a channel attention mechanism to learn the global correlations of temporal channel features to better capture the dynamic dependencies of spatial-temporal data and improve the model's prediction accuracy. We treat the time dimension as a channel and adopt an implementation method similar to the channel attention mechanism. First, the global average pooling operator is used to perform feature compression on the temporal convolution layer output $H_{TC} \in R^{N\times T\times C}$ to obtain the pooled feature $H_{\text{TC}-\text{Mean}}^T \in R^{T\times 1\times 1}$ as shown in (7).

$$H_{\text{TC}-\text{Mean}}^T = \text{MeanPool}\left(H_{TC}\right) = \frac{1}{N\times C}\sum_{i=1}^{N}\sum_{j=1}^{C} H_{TC}\left(i,j\right). \tag{7}$$

Then, the nonlinear correlation among temporal channels is learned by using an MLP and sigmoid activation function to obtain the temporal attention weights $att_T \in R^{T\times 1\times 1}$. This process can be formalized as (8):

$$\text{att}_T = \delta\left(MLP\left(H_{\text{TC}-\text{Mean}}^T\right)\right). \tag{8}$$

Finally, the Hadamard product operation of H_{TC} and att_T is performed to obtain the dynamic weight-adjusted feature H_{att}^T.

$$H_{\text{att}}^T = H_{TC} \circ \text{att}_T. \tag{9}$$

4.3.2. Spatial attention. We design a spatial attention module to capture dynamic spatial dependencies in traffic data. The output of the gated temporal convolution module (H_{TC}) is the input of the spatial attention module. First, the average pooling and maximum pooling of a temporal dimension are performed to obtain two feature expressions ($R^{N\times 1\times 1}$). Afterward, the obtained two features are concatenated together according to the channel followed by the execution of a graph convolution operation. Finally, the spatial attention weight is generated through the sigmoid activation function. The process can be formalized as follows:

$$H_{\text{TC}-\text{Avg}}^S = \text{MeanPool}\left(H_{TC}\right) \tag{10}$$

$$H_{\text{TC}-\text{Max}}^S = \text{MaxPool}\left(H_{TC}\right) \tag{11}$$

$$\text{att}_S = \sigma\left(GCN\left(\text{Concat}\left[\left(H_{\text{TC}-\text{Avg}}^S; H_{\text{TC}-\text{Max}}^S\right), dim = 1\right]\right)\right). \tag{12}$$

The Hadamard product operation of H_{TC} and att_S is performed to obtain the dynamic weight-adjusted feature H^S_{att}. The formula is expressed as follows:

$$H^S_{\mathrm{att}} = H_{TC} \circ \mathrm{att}_S. \tag{13}$$

Generally speaking, temporal and spatial correlations have different degrees of impact on node representation. Therefore, instead of concatenation, we use two weight coefficients to dynamically sum the output of the temporal attention module and the spatial attention module.

$$H_{att} = W_T \circ H^T_{att} + W_S \circ H^S_{att} \tag{14}$$

where $W_T \in R^{N\times T}$ and $W_S \in R^{N\times T}$ are learning parameters. After dynamic fusion, we obtain the node traffic flow representation H_{att} by spatial-temporal attention learning.

4.4. Fully connected layer

H_{att} is linearly transformed through the fully connected network layer to obtain the prediction result $\widehat{X}$, which is expressed as follows:

$$\widehat{X} = W_p * H_{\mathrm{att}} + b_p \tag{15}$$

where W_p and b_p are learnable parameters.

In this paper, the model is trained by minimizing the MSE loss function, which is defined as the mean square error between the true and predicted values.

$$\mathrm{Loss} = \sum_{k=1}^{M} \left\| \widehat{X}_{t+k} - X_{t+k} \right\|^2 \tag{16}$$

where M, X_{t+k}, and $\widehat{X}_{t+k}$ represent the prediction time steps, ground truth, and predicted values, respectively.

5. Experiments

5.1. Dataset description

We conduct experiments on four large-scale real-world traffic prediction datasets, called PeMSD3, PeMSD4, PeMSD7, and PeMSD8. All these datasets are collected from Coltrane's performance measure system (PeMS). The traffic data are aggregated into time intervals of 5 min. In both training and testing process, the traffic flow values of each dataset are normalized with the Z-score normalization method. Table 3 presents the detailed information corresponding to the four datasets.

5.2. Baselines

We compare our model with the following representative baseline methods:

- **SVR** [21]: SVR is a variant method of support vector machine model, which is often used for time series prediction. We use the linear kernel and the penalty term is 0.001.
- **GRU** [49]: gated recurrent unit model is a variant of the traditional RNN.
- **T-GCN** [31]: T-GCN model is in combination with a graph convolutional network and a gated recurrent unit.
- **JK-Net** [50]: jumping knowledge network model implements a deep stacked graph neural network structure and has better performance in suppressing over-smoothing problems.
- **STGCN** [12]: STGCNs model the traffic network by a general graph and employ a fully convolutional structure on the time axis.
- **STSGCN** [32]: spatial-temporal synchronous graph convolutional networks can simultaneously capture the localized spatial temporal correlations directly.
- **AGCRN** [33]: AGCRN model combines node adaptive parameter learning and data adaptive graph generation with recurrent networks.
- **MTGNN** [14]: it combines the stacked block and jumping connection of spatial-temporal convolutional blocks to achieve the learning of multiscale spatial-temporal correlations.
- **STFGNN** [45]: spatial-temporal fusion graph neural network model uses a spatial-temporal fusion graph and a gated dilated convolution to capture local and global complex spatial-temporal dependencies.
- **DSTAGNN** [13]: dynamic spatio-temporal aware graph neural network combined with a multihead attention and a dynamic spatial-temporal aware graph boosts the awareness of dynamic spatial-temporal dependency in time series data.
- **ST-AE** [51]: spatial-temporal autoencoder model designs an autoencoder specifically to learn the intrinsic patterns from traffic flow data and encodes the current traffic flow information into a low-dimensional representation.

5.3. Evaluation metrics

To evaluate the prediction performance of the DGSTAN model, we use mean absolute error (MAE), root mean squared error (RMSE), and mean absolute percentage error (MAPE) to evaluate the difference between the real traffic information and the prediction,

$$\mathrm{MAE} = \sum_{i=1}^{\mathrm{n}} |\hat{y}_{\mathrm{i}} - y_{\mathrm{i}}| \tag{17}$$

$$\mathrm{MAPE} = \frac{100\%}{n} \sum_{i=1}^{n} \left| \frac{\hat{y}_i - y_i}{y_i} \right| \tag{18}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (\hat{y}_i - y_i)^2} \tag{19}$$

where n, y_i, and $\widehat{y}_i$ represent the sample size, ground truth, and predictions of the test set, respectively.

Table 3. Details of four datasets.

Dataset	Time span	#Records	#Nodes
PeMSD3	9/11/2018-11/30/2018	26 208	358
PeMSD4	1/1/2018-2/28/2018	16 992	307
PeMSD7	5/1/2017-8/31/2017	28 224	883
PeMSD8	7/1/2016-8/31/2016	17 856	170

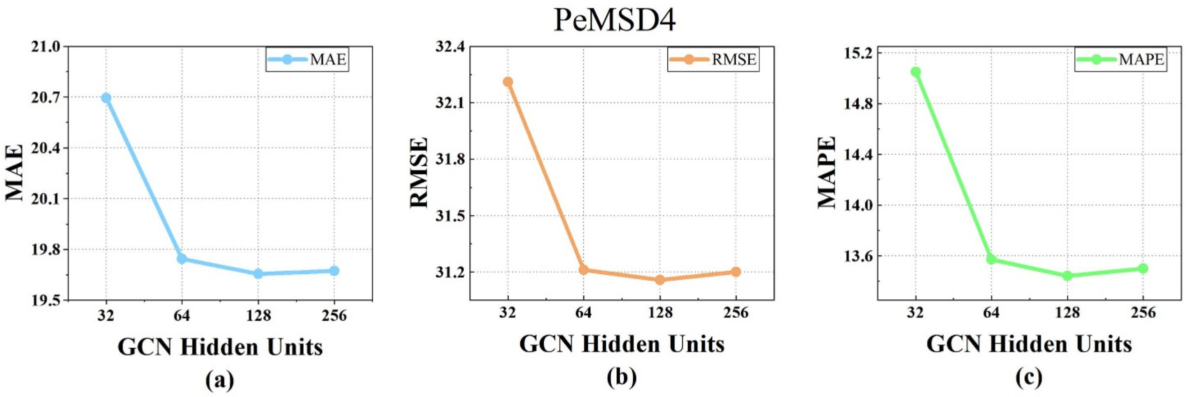


Figure 2. Comparison of prediction performance of different hidden layer units on PeMSD4 dataset.

5.4. Experimental setup

Our model is implemented based on the deep learning framework Pytorch GPU. All the experiments are performed on NVIDIA GeForce GTX 3090 with a memory of 24 GB. We chose Adam as the training optimizer, and the initialized learning rate and mile stones were set to 0.002 and [100, 150, 230], respectively. The batch size was set to 64, and the model was trained for 300 epochs. We used the mean square error as the loss function and selected the Adam optimizer for optimization. We split all datasets with a ratio of 6:2:2 into training, validation, and testing sets. The number of blocks of a multiscale residual graph convolution module K and the number of layers of a gated temporal convolution L were set to $K = 16$ and $L = 1$, respectively. Temporal convolution kernel sizes are set to 3 on all datasets. The padding amplitude is set to 1. In addition, some baseline results (marked with * in table 4) were derived from running their respective open source code, whereas the other baseline results are taken directly from the corresponding papers. The code is available at https://github.com/hnasa/DGSTAN.

Different hidden units may greatly affect the prediction accuracy of our model; therefore, we performed a grid search to explore different hidden units to select the optimal value. In our experiments, for the PeMSD4 and PeMSD8 datasets, we selected the number of hidden layer units from [32,64,128,256] and analyzed the changes in prediction accuracy. Figures 2 and 3 show the results of the RMSE, MAE, and MAPE of the model with different hidden layer units on the PeMSD4 and PeMSD8 datasets, respectively. When the number of hidden layer units is set to 128, our experimental results show that the three evaluation metrics achieve the smallest prediction error.

5.5. Experimental results

5.5.1. Prediction performance analysis. Table 4 shows the prediction results of DGSTAN and other 11 baseline methods for 15 min, 30 min, and 60 min on the datasets PeMSD3, PeMSD4, PeMSD7, and PeMSD8. We find that the proposed model achieves the best performance under all three evaluation metrics. Compared with machine learning methods such as SVR, prediction models based on deep learning have obvious advantages because SVR lacks the ability to learn complex nonlinear features. At the same time, we also observe that under the framework of deep learning, whether it is the GRU model that only captures temporal-dependent features or the JK-Net model that only considers spatial correlation feature learning, their prediction performance is not ideal. The STGCN model can effectively extract the spatial correlation and temporal dynamic features of traffic flow data, resulting in a better prediction performance.

Although our DGSTAN model is usually better than the baseline method, it is worth noting that ST-AE exhibits its extremely excellent performance in certain scenarios, as shown in table 4. The ST-AE model combines pretraining with a spatiotemporal auto-encoder framework to effectively capture inherent patterns and dependencies in data, which may help it compete for accuracy, especially in capturing short-term fluctuations on certain datasets with specific data distributions. Compared with STSGCN, STFGNN, AGCRN, MTGNN, DSTAGNN, and other STGCN models, DGSTAN adopts a deep stacked graph convolutional network structure to improve the spatial feature capturing ability. Simultaneously, it introduces a joint spatial-temporal attention mechanism to improve the dynamic spatial-temporal feature learning ability of the model. Table 4 shows that compared with other baseline

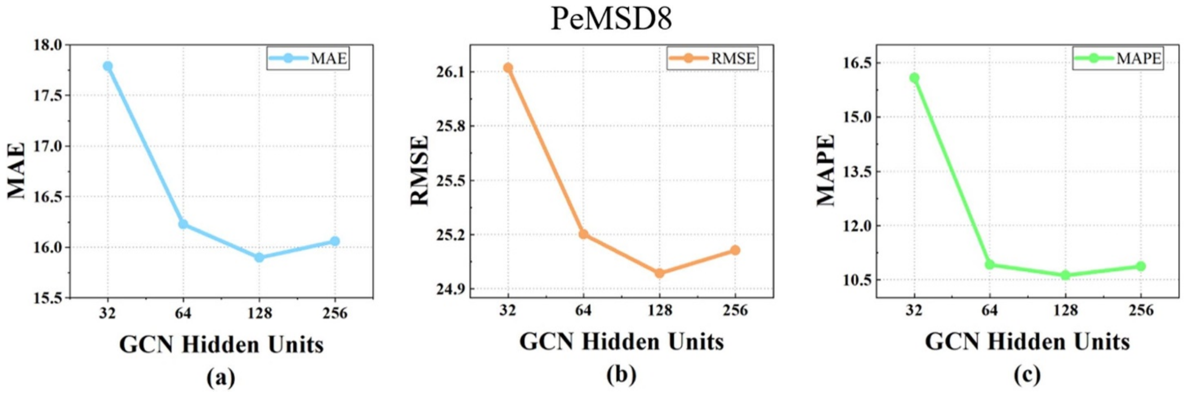


Figure 3. Comparison of prediction performance of different hidden layer units on PeMSD8 dataset.

models, as the prediction duration extends, our model exhibits a more gradual increase in prediction error. In short, the above analysis shows that our model has advantages in medium- and long-term traffic flow prediction.

5.5.2. Visualization of forecast results. Figure 4 provides a qualitative comparison of the uncertainty characterizations, showing example confidence intervals for 15 min time step at node 196 on the first day of the test set in PeMSD4 dataset. We find that the proposed DGSTAN model provides considerably tighter intervals while still achieving coverage of the observed values.

5.5.3. Ability to mitigate oversmoothing. To verify the suppression ability of the DGSTAN model on the over-smoothing problem, we compared and analyzed the prediction results of the GCN stacked model, the JK-Net model, and the DGSTAN model on PeMSD4 dataset with different numbers of graph convolution layers for 15 min, 30 min, and 60 min prediction durations. Table 5 shows that the prediction performance of GCN simple stacked models decreases significantly with increasing number of graph convolution layers. Results for different prediction cycles are unusually close to the same number of GCN layers. This indicates that the lack of differentiation of node features after multiple graph convolutions. However, our model and the JK-Net model exhibit a slight improvement in prediction accuracy rather than a decrease. This conclusion indicates that the deep stacked graph convolutional network structure described in this paper, along with the JK-Net model, is capable of learning long-range spatial dependencies and exhibits improved over-smoothing suppression.

To better reflect intuitively the prediction results of the GCN stacked model, the JK-Net model, and the proposed DGSTAN model in this paper, we provide actual data for a 60 min prediction period using the PeMSD4 dataset and visualize the distribution probability of the prediction outcomes for each model. To this end, we set the number of graph convolution layers in each model to 32. Figure 5 shows the experimental results.

Figure 5(b) shows that GCN stacked model predicts traffic flow results for all road segments with a 90% distribution [0–100] in range of values. This indicates that most road sections' traffic flow characteristics lack differentiation, meaning there is an over-smoothing problem. Probability distributions of prediction results of the JK-Net model and the DGSTAN model are consistent with real data. This indicates that these two models can maintain personalized features in traffic flow data during the learning process of long-range dependencies (multiscale spatial features). By comparing and analyzing figures 5(c) and (d) , we can find that our model demonstrates a more distinct advantage compared to the JK-Net model in preserving feature differentiation for low-frequency data, as indicated by the traffic flow data below 40 in figure 5. This also illustrates that our model enhances the learning of dynamic spatial dependencies between nodes through MLP-based fusion feature learning and spatial-temporal attention mechanism. It avoids excessive influence of neighboring nodes' information on low-frequency data, and to some extent, preserves the nodes' inherent spatial-temporal characteristics of traffic flow.

5.6. Ablation study

To further investigate why DGSTAN performs well in traffic prediction, we construct a series of ablation studies to examine the performance gain of the four main components (i.e., multiscale residual graph convolution, gated temporal convolution, and spatial-temporal attention) in the DGSTAN model. Several variants of DGSTAN are obtained by dynamically configuring these modules as follows:

- **DGSTAN-I:** it replaces the multiscale residual graph convolution module in DGSTAN with a two-layer graph convolutional network.
- **DGSTAN-II:** it replaces the multiscale residual graph convolution module's MLP module in DGSTAN with a linear layer.
- **DGSTAN-III:** It removes the gated temporal convolution module in DGSTAN.

Table 4. Performance comparison of different models on four datasets.

		15 min			30 min			60 min		
Datasets	Methods	MAE	RMSE	MAPE(%)	MAE	RMSE	MAPE(%)	MAE	RMSE	MAPE(%)
PeMSD3	SVR	17.14	29.63	17.07	21.16	35.41	21.39	29.77	47.12	30.17
	GRU	18.34	31.02	17.48	23.78	36.01	21.89	28.65	45.03	29.05
	T-GCN	15.01	26.11	15.89	19.75	31.45	18.62	25.11	28.91	25.95
	JK-NET	19.01	27.95	21.22	20.45	30.25	30.08	21.91	33.61	23.74
	STGCN	17.43	28.43	16.99	19.25	31.33	18.4	23.48	37.51	21.95
	STSGCN	15.52	25.25	15.22	17.28	28.49	16.66	20.41	33.75	19.04
	AGCRN	14.81	25.99	14.95	15.96	27.95	16.38	18.5	31.58	18.86
	MTGNN	*14.95	*25.52	*14.69	*16.24	*27.62	*16.03	*18.96	*31.02	*18.67
	STFGNN	15.19	25.3	15.01	16.69	27.75	16.12	19.38	32.05	18.29
	DSTAGNN	*14.71	*25.65	*14.65	*16.07	*28.20	*15.64	*18.18	*31.16	*17.65
	ST-AE	* **14.06**	*23.95	*14.32	* **15.71**	*26.84	*15.60	* **17.98**	*30.47	*17.66
	DGSTAN	14.51	**23.63**	**14.16**	15.91	**26.73**	**15.54**	18.15	**30.30**	**17.62**
PeMSD4	SVR	22.76	35.71	15.07	26.36	41.01	17.97	37.23	55.67	26.57
	GRU	25.78	41.18	15.99	26.46	42.23	15.95	27.49	43.62	16.22
	T-GCN	21.96	33.03	17.10	22.77	34.31	17.45	25.54	37.86	20.88
	JK-NET	27.43	40.88	21.38	28.07	41.76	22.65	31.53	46.16	23.97
	STGCN	20.04	31.65	12.69	23.09	35.97	14.33	29.51	43.97	18.11
	STSGCN	19.69	31.05	13.07	21.18	33.32	13.91	24.61	38.13	16.15
	AGCRN	19.14	30.60	13.19	20.4	32.43	13.72	23.11	36.22	15.82
	MTGNN	*19.07	*30.47	*13.08	*20.32	*32.23	*13.66	*22.86	*35.95	*15.42
	STFGNN	18.72	29.98	13.32	19.76	31.7	13.85	21.85	34.34	14.91
	DSTAGNN	*18.87	*30.03	*12.86	*19.85	*31.73	*13.98	*22.47	*35.24	*14.83
	ST-AE	* **18.47**	*29.84	*13.73	*19.72	*31.14	*14.99	*22.01	*34.12	*17.94
	DGSTAN	18.58	**29.47**	**12.82**	**19.63**	**31.09**	**13.43**	**21.75**	**34.04**	**14.59**
PeMSD7	SVR	24.33	37.32	10.32	29.97	45.41	12.84	41.33	60.38	18.31
	GRU	26.09	39.41	10.45	31.06	47.99	13.83	38.67	55.46	16.28
	T-GCN	23.55	35.91	10.11	27.87	43.75	12.04	36.88	55.04	15.88
	JK-NET	26.78	39.29	12.05	29.35	42.81	13.14	33.50	48.44	15.07
	STGCN	25.12	39.61	10.81	27.73	43.48	11.92	33.80	51.74	14.75
	STSGCN	21.39	34.00	8.96	24.01	38.53	10.04	28.99	46.32	12.27
	AGCRN	20.54	32.92	8.63	22.43	35.98	9.49	25.75	40.54	10.85
	MTGNN	*20.43	*32.76	*8.69	*22.39	*36.06	*9.45	*25.52	*40.49	*10.82
	STFGNN	20.46	32.89	8.58	22.21	36.09	9.46	25.60	40.76	10.89
	DSTAGNN	*20.58	*32.51	*8.96	*22.44	*35.85	*9.88	*25.84	*41.14	*11.53
	ST-AE	* **20.32**	*32.19	* **8.27**	*22.45	*35.46	*9.64	*26.07	*40.30	*11.41
	DGSTAN	20.41	**32.16**	9.17	**22.00**	**35.13**	**9.41**	**25.03**	**39.89**	**10.67**
PeMSD8	SVR	18.29	28.43	11.12	21.12	33.1	12.99	30.56	45.69	19.2
	GRU	24.83	41.97	13.53	25.55	42.6	13.91	26.7	43.8	14.13
	T-GCN	18.8	27.18	15.89	19.7	28.58	16.62	22.33	32.09	18.8
	JK-NET	26.17	36.51	19.41	26.23	36.68	20.51	27.93	39.15	21.54
	STGCN	15.88	24.44	9.93	18.59	28.21	11.14	24.06	35.64	13.77
	STSGCN	15.73	24.23	10.11	17	26.37	10.82	19.7	30.34	12.39
	AGCRN	15.11	23.65	9.88	16.49	25.72	10.89	18.16	28.7	12.74
	MTGNN	*15.06	*23.45	*9.98	*16.21	*25.63	*10.73	*18.85	*29.03	*12.87
	STFGNN	15.41	23.93	9.92	16.77	26.31	10.67	19.43	30.25	12.23
	DSTAGNN	*15.26	*23.66	*9.96	*16.53	*26.05	*10.86	*18.88	*29.51	*12.35
	ST-AE	*15.04	*23.42	*9.85	*16.67	*25.97	*10.78	*18.80	*28.95	*12.15
	DGSTAN	**14.82**	**22.93**	**9.82**	**15.90**	**24.92**	**10.52**	**17.92**	**27.93**	**12.09**

Note: * denotes re-implementation or re-training. In each column, the best is highlighted in bold, and the second-best is underlined.

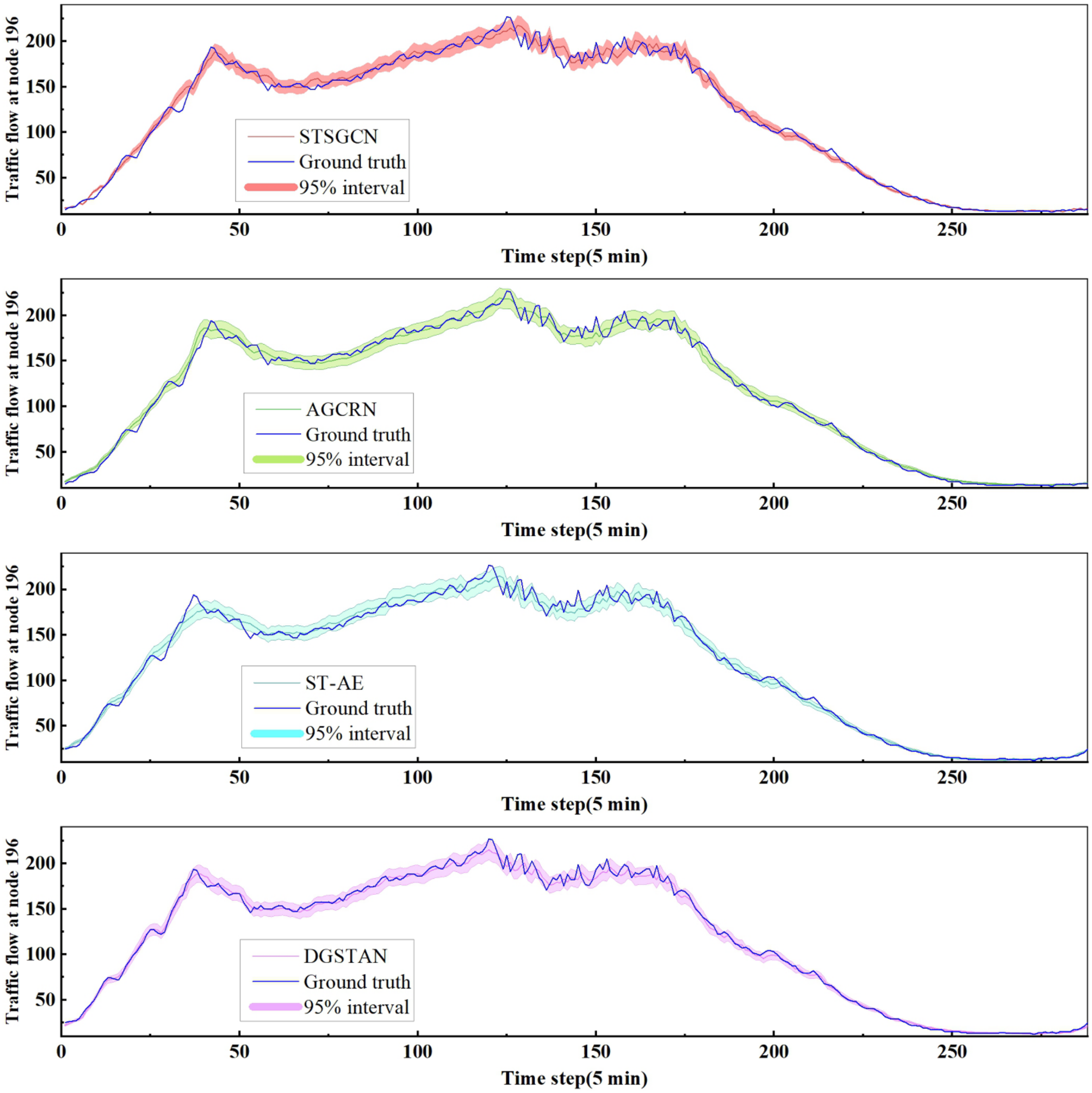


Figure 4. Qualitative comparison of the uncertainty of prediction results.

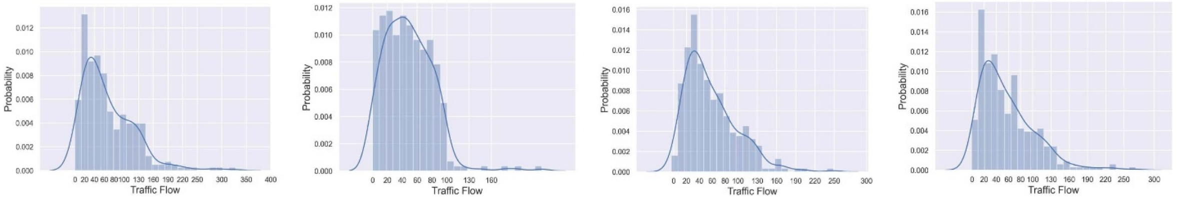
- **DGSTAN-IV:** it removes the spatial-temporal attention module in DGSTAN.
- **DGSTAN-V:** it removes the spatial attention module in DGSTAN.
- **DGSTAN-VI:** it removes the temporal attention module in DGSTAN.

Figures 6 and 7 show the comparison between different model variants, from which we can draw the following conclusions.

- As an important part of DGSTAN, the multiscale residual graph convolution learns the multirange spatial correlation between nodes. Accordingly, DGSTAN-I can only model the local range relation for traffic prediction. Figures 6 and 7 show that DGSTAN-I did not show better results, which reflects that the residual graph convolution learning in a large range has a positive contribution to the performance gain of flow prediction. Furthermore, we analyze the importance of MLP in DGSTAN. Figures 6 and 7 show that DGSTAN-II performs worse than DGSTAN. This indicates that the MLP module is crucial for effectively integrating the multiscale features. DGSTAN-II replaces the MLP module with a linear layer and is insufficient to integrate the multiscale features, leading to a degradation in prediction accuracy.
- Multiscale residual graph convolution alone cannot capture the temporal-varying features of traffic flow data. The performance degradation of DGSTAN-III verifies that the gated temporal convolution module also plays an important role in capturing the time series dependencies, which can further enhance the accuracy of traffic flow prediction.
- Multiscale residual graph convolution cannot capture the importance of a certain segment of node traffic flow to the

Table 5. Prediction results of different models as the number of graph convolution layers increases.

		GCN stacked			JK-Net			DGSTAN		
Time steps	Number of	MAE	RMSE	MAPE(%)	MAE	RMSE	MAPE(%)	MAE	RMSE	MAPE(%)
15 min	2	**32.29**	**47.06**	**25.85**	32.02	47.05	25.75	19.43	30.57	13.34
	4	32.99	48.02	28.23	31.51	46.06	25.36	18.93	29.87	13.47
	8	35.38	51.69	28.71	28.70	42.15	21.86	18.75	29.71	12.94
	16	37.41	54.39	31.09	**27.07**	**39.88**	**21.60**	18.71	29.69	12.87
	32	109.11	133.83	179.32	27.43	40.88	21.38	**18.58**	**29.47**	**12.82**
30 min	2	**33.19**	**48.26**	**26.63**	33.09	48.19	26.44	21.01	32.81	14.29
	4	33.70	48.79	27.76	32.57	47.46	26.25	20.17	31.71	14.09
	8	34.70	50.21	28.92	27.62	40.96	21.47	19.98	31.54	13.73
	16	37.52	53.71	31.21	**27.57**	**40.94**	**22.07**	19.74	31.27	13.63
	32	109.11	133.84	179.33	28.07	41.76	22.65	**19.63**	**31.09**	**13.43**
60 min	2	**36.11**	**52.78**	**27.64**	35.11	52.38	27.94	24.11	37.02	16.37
	4	37.72	54.73	27.84	34.55	50.07	28.08	22.67	35.17	15.65
	8	37.53	54.56	29.17	**29.96**	**44.31**	**23.86**	22.37	34.92	15.49
	16	39.41	57.02	31.13	30.42	44.70	24.01	21.87	34.17	14.79
	32	125.91	162.38	166.46	31.53	46.16	23.97	**21.75**	**34.04**	**14.59**



(a) Distribution of real data

(b) Distribution of prediction results of GCN model

(c) Distribution of prediction results of JK-Net model

(d) Distribution of prediction results of DGSTAN model

Figure 5. Comparison of the distribution similarity between the predicted result of different models with the real data.

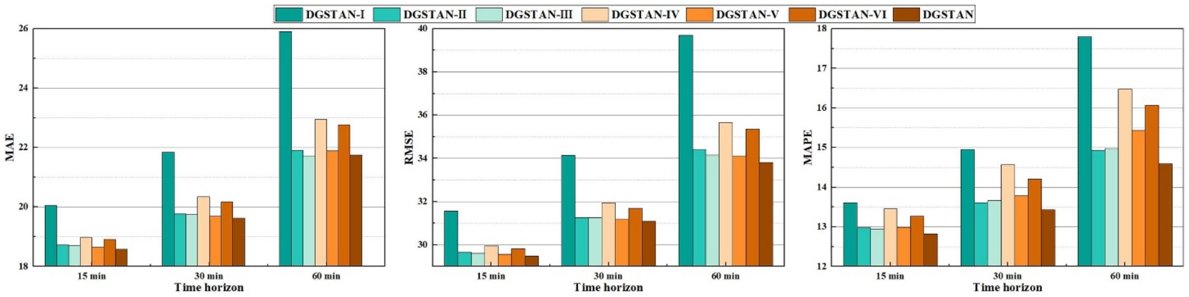


Figure 6. Ablation studies on the PeMSD4 dataset.

entire road network. In addition, especially for long-term time series inputs, gated temporal convolution cannot know the importance of a certain time step to the global time series. The decline in DGSTAN-IV performance verifies that the spatial-temporal attention mechanism also plays an important role in capturing global spatial-temporal dynamic weights. Regarding the question of who is more important for temporal and spatial attention, we find through experiments on DGSTAN-V and DGSTAN-VI that their contribution varies with the prediction horizons. More specifically, for longer prediction horizons (such as 60 min), we find that the temporal attention mechanism is more important.

5.7. Effectiveness analysis of temporal and spatial attention

We analyze the importance of the connection structure for temporal and spatial attention. In this experiment, we have implemented two different arrangements of spatial and temporal attention components, which correspond to the replacement of the parallel structure of the spatial-temporal joint attention

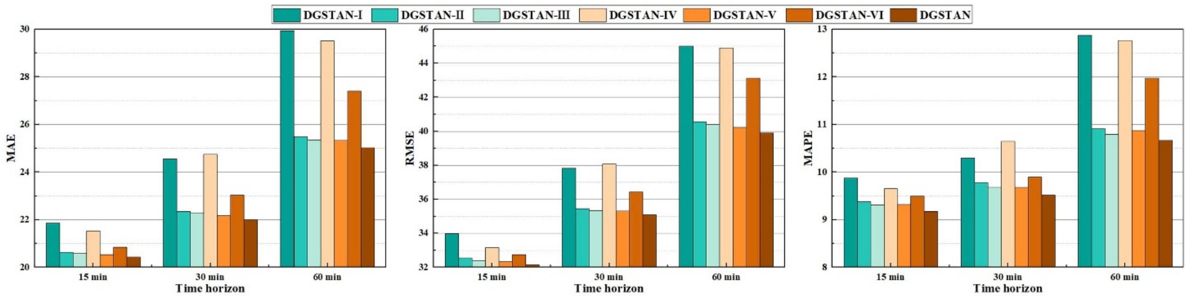
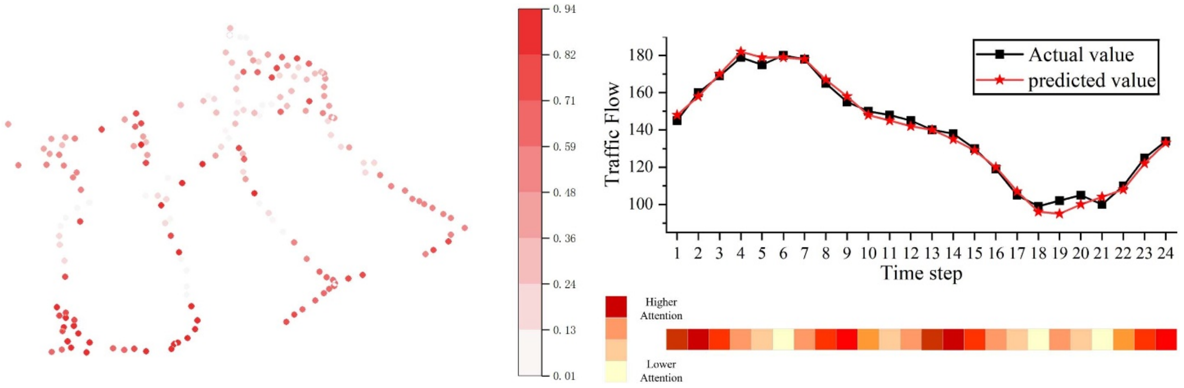


Figure 7. Ablation studies on the PeMSD7 dataset.

Table 6. Combining methods of temporal and spatial attention.

	PeMSD4			PeMSD7		
Description	MAE	RMSE	MAPE(%)	MAE	RMSE	MAPE(%)
Spatial attention + Temporal attention	20.21	31.78	13.54	22.75	35.89	9.71
Temporal attention + Spatial attention	19.89	31.50	13.83	22.31	35.56	9.85
Spatial attention and Temporal attention in parallel	**19.68**	**31.17**	**13.43**	**22.06**	**35.24**	**9.61**



(a) Spatial Attention visualization

(b) Temporal Attention visualization and prediction results over 24-time steps.

Figure 8. Spatial-temporal dependency obtained by DGSTAN on the PeMSD4 dataset.

with the cascaded structure of temporal + spatial or spatial + temporal.

Table 6 summarizes the experimental results on the connection structure of different attention components. From the results, we find that our model with spatial attention and temporal attention in parallel can generate a more accurate attention graph than the cascaded structure. This also shows that in the cascade structure, temporal and spatial feature information may be lost or blurred during the transfer process, which affects the accuracy of capturing spatial-temporal relationships.

In addition, to qualitatively analyze the utility of the spatial-temporal attention mechanism, we have visualized the spatial and temporal attention acquired by our model. Figure 8(a) shows that the proposed model has ability in identifying complex traffic conditions such as road network intersections. In addition, figure 8(b) shows that the traffic flow prediction results over a 24-time steps period after a certain peak period in the PeMSD4 test datasets and the temporal attention heat map is visualized. In summary, our model achieves highly promising performance in traffic flow prediction, extracts complex information within the road network and effectively identifies the time-varying trends in traffic flow.

5.8. Computation cost analysis

We measure the computational cost of the models by comparing the parameter size and training time (prediction time is set to 60 min) of STSGCN, AGCRN, DSTAGNN, ST-AE, and DGSTAN on PeMSD4 and PeMSD8 datasets. Table 7 shows that the parameter size and training time per epoch of the DGSTAN model are smaller than those of STSGCN, AGCRN, and DSTAGNN models. Based on our experimental results, we obtain the optimal number of hidden units for the DGSTAN model from a certain level of grid search. Moreover, the consistency of the input and output dimensions of the graph

Table 7. Comparison of calculation costs of different models on PeMSD4 and PeMSD8 datasets.

	PeMSD4		PeMSD8	
Model	Parameters	Training Time(epoch)	Parameters	Training Time(epoch)
STSGCN	2024 445	18.163 s	1401 232	17.068 s
AGCRN	748 810	35.965 s	150 112	23.411 s
DSTAGNN	3579 728	106.712 s	2296 860	52.201 s
ST-AE	165 851	14.433 s	160 371	13.339 s
DGSTAN	**75 581**	**12.702 s**	**47 907**	**10.897 s**

convolutional block is maintained by an implicit dimensional transformation, which reduces the parameter size of the model to some extent. AGCRN and STSGCN have more parameters than other models as a sacrifice for learning node-specific patterns because both models use graph convolution operations that do not share parameters to learn node-specific patterns. In terms of training time, AGCRN runs slightly slower than STSGCN, which is due to its large iterative calculation with the GRU component rather than the parallel calculation based on the gated time convolution in STSGCN. In terms of model parameters and training time overhead, ST-AE achieves the second best among others. Compared to our DGSTAN model, ST-AE takes more time to perform temporal convolution and long-term attention calculation.

6. Conclusion

A deep graph convolution spatial-temporal attention learning network model DGSTAN is proposed in this paper. The model jointly adopts deep stacked graph convolution, gated temporal convolution and spatial-temporal attention mechanism to capture the spatial-temporal features of traffic flow data. Experiments on four real traffic datasets reveal that the proposed model is superior to other existing traffic data prediction methods in terms of prediction accuracy. Afterward, it is verified that the model has advantages in capturing spatial-temporal features and dynamic spatial-temporal correlations. Meanwhile, compared with other STGCNs, we show that the proposed model also performs better in terms of parameter scale and training speed.

Future work will include three aspects. The first aim is to conduct data replenishment and recovery work for the data deficiencies and noise data problems in the traffic flow dataset to improve the quality of the input data. The second objective is to collect data related to the evolution of traffic flow, and perform multisource data feature correlation and collaborative traffic flow predictions modeling to improve the model's medium- and long-term traffic flow prediction accuracy in complex traffic environments. Finally, constructing a dynamic and adaptive adjacency matrix might be a good way to go. This matrix aims to enhance the graph convolution operator, thus further improving the dynamic spatial-temporal feature learning capabilities of our GCN model.

Data availability statement

The data cannot be made publicly available upon publication because they are owned by a third party and the terms of use prevent public distribution. The data that support the findings of this study are available upon reasonable request from the authors.

ORCID iDs

Liming Jiang https://orcid.org/0000-0003-0520-702X
Baiyi Liu https://orcid.org/0009-0004-3779-223X
Huanyu Wang https://orcid.org/0000-0001-9630-5869

References

[1] Bickel P J, Chen C, Kwon J, Rice J, van Zwet E and Varaiya P 2007 Measuring traffic *Stat. Sci.* 581–97

[2] Yang X, Luo S, Gao K, Qiao T and Chen X 2019 Application of data science technologies in intelligent prediction of traffic congestion *J. Adv. Trans.* **2019** 2915369

[3] Hou Z and Li X 2016 Repeatability and similarity of freeway traffic flow and long-term prediction under big data *IEEE Trans. Intell. Transp. Syst.* **17** 1786–96

[4] Wu J, He D, Jin Z, Li X, Li Q and Xiang W 2024 Learning spatial–temporal pairwise and high-order relationships for short-term passenger flow prediction in urban rail transit *Expert Syst. Appl.* **245** 123091

[5] He D, Zhao J, Jin Z, Huang C, Zhang F and Wu J 2025 Prediction of bearing remaining useful life based on a two-stage updated digital twin *Adv. Eng. Inf.* **65** 103123

[6] Feng W, Qi S, Guo J, Zuo X, Chen Y and Zhu Y 2024 Traffic signal current prediction algorithm based on CNN and LSTM *Meas. Sci. Technol.* **36** 015032

[7] Smith B L, Williams B M and Oswald R K 2002 Comparison of parametric and nonparametric models for traffic flow forecasting *Transp. Res. C* **10** 303–21

[8] LeCun Y, Bengio Y and Hinton G 2015 Deep learning *Nature* **521** 436–44

[9] Zhao Z, Chen W, Wu X, Chen P C Y and Liu J 2017 LSTM network: a deep learning approach for short-term traffic forecast *IET Intell. Transp. Syst.* **11** 68–75

[10] Shi X, Chen Z, Wang H, Yeung D Y, Wong W K and Woo W 2015 *Advances in Neural Information Processing Systems* vol 28

[11] Zhang J, Zheng Y and Qi D 2017 Deep spatio-temporal residual networks for citywide crowd flows prediction *Proc. AAAI Conf. on Artificial Intelligence* vol 31

[12] Yu B, Yin H and Zhu Z 2018 Spatio-Temporal graph convolutional networks: a deep learning framework for

traffic forecasting *Proc. 27th Int. Joint Conf. on Artificial Intelligence* pp 3634–40

[13] Lan S, Ma Y, Huang W, Wang W, Yang H and Li P 2022 Dstagnn: dynamic spatial-temporal aware graph neural network for traffic flow forecasting *Int. Conf. on Machine Learning* (PMLR) pp 11906–17

[14] Wu Z, Pan S, Long G, Jiang J, Chang X and Zhang C 2020 Connecting the dots: multivariate time series forecasting with graph neural networks *Proc. 26th ACM SIGKDD Int. Conf. on Knowledge Discovery & Data Mining* pp 753–63

[15] Bingbing X, Keting C, Junjie H, Hua-Wei S and Xue-Qi C 2020 A survey on graph convolutional neural network *Chin. J. Comput.* **43** 755–80

[16] Kamarianakis Y and Prastacos P 2003 Forecasting traffic flow conditions in an urban network: comparison of multivariate and univariate approaches *Transp. Res. Rec.* **1857** 74–84

[17] Wu C-H, Ho J-M and Lee D T 2004 Travel-time prediction with support vector regression *IEEE Trans. Intell. Transp. Syst.* **5** 276–81

[18] Okutani I and Stephanedes Y J 1984 Dynamic prediction of traffic volume through Kalman filtering theory *Transp. Res. B* **18** 1–11

[19] Gu Y, Lu W, Xu X, Qin L, Shao Z and Zhang H 2019 An improved Bayesian combination model for short-term traffic prediction with deep learning *IEEE Trans Intell. Transp. Syst.* **21** 1332–42

[20] Kramer O 2013 K-nearest neighbors *Dimensionality Reduction with Unsupervised Nearest Neighbors* (Springer) pp 13–23

[21] Awad M, Khanna R, Awad M and Khanna R 2015 Support vector regression *Efficient Learning Machines: Theories, Concepts and Applications for Engineers and System Designers* (Apress) pp 67–80

[22] Zhong Z, Li J, Luo Z and Chapman M 2017 Spectral–spatial residual network for hyperspectral image classification: a 3-D deep learning framework *IEEE Trans. Geosci. Remote Sens.* **56** 847–58

[23] Cheng G, Yang C, Yao X, Guo L and Han J When deep learning meets metric learning: Remote sensing image scene classification via learning discriminative CNNs 2018 *IEEE Trans. Geosci. Remote Sens.* **56** 2811–21

[24] Saon G, Tüske Z, Bolanos D and Kingsbury B 2021 Advancing RNN transducer technology for speech recognition *ICASSP 2021-2021 IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)* (IEEE) pp 5654–8

[25] Gu Y, Tinn R, Cheng H, Lucas M, Usuyama N, Liu X, Naumann T, Gao J and Poon H 2021 Domain-Specific language model pretraining for biomedical natural language processing *ACM Trans. Comput. Healthcare* **3** 1–23

[26] Guo J, Liu Y, Yang Q, Wang Y and Fang S 2021 GPS-based citywide traffic congestion forecasting using CNN-RNN and C3D hybrid model *Transportmetrica A* **17** 190–211

[27] Ounoughi C and Yahia S B 2024 Sequence to sequence hybrid Bi-LSTM model for traffic speed prediction *Expert Syst. Appl.* **236** 121325

[28] Zheng H, Lin F, Feng X and Chen Y 2020 A hybrid deep learning model with attention-based Conv-LSTM networks for short-term traffic flow prediction *IEEE Trans. Intell. Transp. Syst.* **22** 6910–20

[29] Yao H, Wu F, Ke J, Tang X, Jia Y, Lu S, Gong P, Ye J and Li Z 2018 Deep multi-view spatial-temporal network for taxi demand prediction *Proc. AAAI Conf. on Artificial Intelligence* vol 32

[30] Wang X, Ma Y, Wang Y, Jin W, Wang X, Tang J, Jia C and Yu J 2020 Traffic flow prediction via spatial temporal graph neural network *Proc. Web Conf. 2020* pp 1082–92

[31] Zhao L, Song Y, Zhang C, Liu Y, Wang P, Lin T, Deng M and Li H 2019 T-GCN: a temporal graph convolutional network for traffic prediction *IEEE Trans. Intell. Transp. Syst.* **21** 3848–58

[32] Song C, Lin Y, Guo S and Wan H 2020 Spatial-Temporal synchronous graph convolutional networks: a new framework for spatial-temporal network data forecasting *Proc. AAAI Conf. on Artificial Intelligence* vol 34 pp 914–21

[33] Bai L, Yao L, Li C, Wang X and Wang C 2020 *Advances in Neural Information Processing Systems* vol 33 pp 17804–15

[34] Wang S, Zhang M, Miao H, Peng Z and Yu P S 2022 Multivariate correlation-aware spatio-temporal graph convolutional networks for multi-scale traffic prediction*ACM Trans. Intell. Syst. Technol.* **13** 1–22

[35] Liu T and Meidani H 2024 arXiv:2408.04131

[36] Liu T and Meidani H 2024 End-to-end heterogeneous graph neural networks for traffic assignment *Trans. Res. C* **165** 104695

[37] Feng R, Chen M and Song Y 2023 Learning traffic as videos: short-term traffic flow prediction using mixed-pointwise convolution and channel attention mechanism *Expert Syst. Appl.* **240** 122468

[38] Yu W, Huang X, Qiu Y, Zhang S and Chen Q 2023 GSTC-Unet: a U-shaped multi-scaled spatiotemporal graph convolutional network with channel self-attention mechanism for traffic flow forecasting *Expert Syst. Appl.* **232** 120724

[39] Guo S, Lin Y, Feng N, Song C and Wan H 2019 Attention based spatial-temporal graph convolutional networks for traffic flow forecasting *Proc. AAAI Conf. on Artificial Intelligence* vol 33 pp 922–9

[40] Lu B, Gan X, Jin H, Fu L, Wang X and Zhang H 2022 Using LSTM and GRU neural network methods for traffic flow prediction *ACM Trans. Intell. Syst. Technol.* **13** 1–25

[41] Liu T and Meidani H 2025 arXiv:2501.09117

[42] Li Y, Yu R, Shahabi C and Liu Y 2018 Diffusion convolutional recurrent neural network: Data-Driven traffic forecasting *Int. Conf. on Learning Representations (ICLR'18)*

[43] Wu Z, Pan S, Long G, Jiang J and Zhang C 2019 Graph wavenet for deep spatial-temporal graph modeling *Proc. 28th Int. Joint Conf. on Artificial Intelligence* pp 1907–13

[44] Cao D *et al* 2020 *Advances in Neural Information Processing Systems* vol 33 pp 17766–78

[45] Li M and Zhu Z 2021 Spatial-Temporal fusion graph neural networks for traffic flow forecasting *Proc. AAAI Conf. on Artificial Intelligence* vol 35 pp 4189–96

[46] Shao Z, Zhang Z, Wang F and Xu Y 2022 Pre-training enhanced spatial-temporal graph neural network for multivariate time series forecasting *Proc. 28th ACM SIGKDD Conf. on Knowledge Discovery and Data Mining* pp 1567–77

[47] Hu J, Shen L and Sun G 2018 Squeeze-and-excitation networks *Proc. IEEE Conf. on Computer Vision and Pattern Recognition* pp 7132–41

[48] Zhao J, Liu Z, Sun Q, Li Q, Jia X and Zhang R 2022 Attention-based dynamic spatial-temporal graph convolutional networks for traffic speed forecasting *Expert Syst. Appl.* **204** 117511

[49] Fu R, Zhang Z and Li L 2016 Using Lstm and GRU neural network methods for traffic flow prediction *2016 31st Youth Academic Annual Conf. of Chinese Association of Automation (YAC)* (IEEE) pp 324–8

[50] Xu K, Li C, Tian Y, Sonobe T, Kawarabayashi K i and Jegelka S 2018 Representation learning on graphs with jumping knowledge networks *Int. Conf. on Machine Learning* (PMLR) pp 5453–62

[51] Liu M, Zhu T, Ye J, Meng Q, Sun L and Du B 2023 Spatio-temporal autoencoder for traffic flow prediction *IEEE Trans. Intell. Transp. Syst.* **24** 5516–2