Adaptive Spatial-Temporal Fusion Graph Convolutional Networks for Traffic Flow Forecasting

Senwen Li[1,2,*], Liang Ge[1,2,*], Yongquan Lin[1,2], Bo Zeng[1,2]
[1]College of Computer Science, Chongqing University, China
[2]Chongqing Key Laboratory of Software Theory Technology, China
{senwenli, geliang, CquLyq, SimonZB}@cqu.edu.cn

2022 International Joint Conference on Neural Networks (IJCNN) | 978-1-7281-8671-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/IJCNN55064.2022.9892326

***Abstract*—Traffic flow forecasting is a significant issue in the field of transportation. Early works model temporal dependencies and spatial correlations, respectively. Recently, some models are proposed to capture spatial-temporal dependencies simultaneously. However, these models have three defects. Firstly, they only use the information of road network structure to construct graph structure. It may not accurately reflect the spatial-temporal correlations among nodes. Secondly, only the correlations among nodes adjacent in time or space are considered in each graph convolutional layer. Finally, it's challenging for them to describe that future traffic flow is influenced by different scale spatial-temporal information. In this paper, we propose a model called Adaptive Spatial-Temporal Fusion Graph Convolutional Networks to address these problems. Firstly, the model can find cross-time, cross-space correlations among nodes to adjust spatial-temporal graph structure by a learnable adaptive matrix. Secondly, it can help nodes attain a larger spatiotemporal receptive field through constructing spatial-temporal graphs of different time spans. At last, the results of various spatial-temporal scale graph convolutional layers are fused to produce node embedding for prediction. It helps find the different spatial-temporal ranges' influence for various nodes. Experiments are conducted on real-world traffic datasets, and results show that our model outperforms the state-of-the-art baselines.**

***Index Terms*—Traffic flow forecasting, Spatial-temporal data, Adaptive spatial-temporal fusion, Graph convolution network.**

I. Introduction

With the development of the Intelligent Transportation System (ITS), traffic data prediction becomes a significant part of ITS to help efficient transportation management. If traffic data can be predicted accurately, ITS will be able to help plan the route of vehicles and traffic lights more legitimately, which can ease traffic congestion problems.

Traffic flow is the number of traffic entities passing through a place or a lane in a selected period. In the scenario of urban life, people generally drive from one area to its adjacent areas at a specific time. For example, when people go to work, they drive from residential areas to commercial areas. So the flows of the residential areas have impacts on the flows of commercial areas. At the same time, because populations of residents are not entirely the same, different residents' flows have various influences on a commercial area. Thus traffic flow data have spatial dependencies. The flow of an area has different impacts on different areas, and the influence that an area's flow has on another area will change over time, which is defined as heterogeneities. For example, more vehicles move from residential areas to commercial areas in the commuting time but less at other times. On the other hand, for the same area, the traffic flow usually does not change dramatically in a short time. It generally is related to the previous flow change of the same area. Thus traffic data have temporal correlations. Moreover, when people drive from the departure to the destination, the traffic flows of the areas that people drive through will change and ultimately influence the flow of the destination. It means we can predict the destination's future flow based on the departure's historical flow. Thus traffic data have spatial-temporal dependencies.

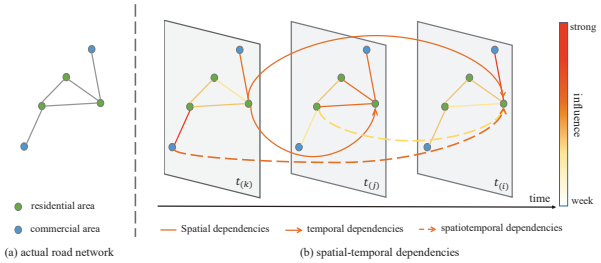


Fig. 1. The Spatial-temporal dependencies of traffic flow.

In conclusion, traffic flow data can be considered as specific spatial-temporal data and be described by spatial-temporal graphs. Fig. 1 shows an example of describing traffic flow data by the spatial-temporal graph. We use different line types to denote different spatiotemporal correlations in the figure. Solid lines indicate that the flow of an area is affected by the flow of other areas in one time period. Solid arrows denote that the historical flow of an area influences its future flow. Besides, dashed arrows indicate that the flow of an area is affected by the historical flow of other areas. The deeper the line color is, the stronger the correlation between the two areas is. If a model utilizes these characteristics well, it can have better prediction performance.

Early methods [1], [2] based on statistics or machine learning just consider temporal dependencies and ignore spatial

*Equal Contribution

978-1-7281-8671-9/22/$31.00 ©2022 IEEE

factors. To get better prediction performance, spatial-temporal models [3], [4], [5] take spatial correlations into account. Existing researches usually introduce Graph Convolution Networks (GCN) and its variants for capturing spatial dependencies. Simultaneously, they use other components like Recurrent Neural Networks (GRU) or Convolutional Neural Networks (CNN) to model temporal correlations. These models are designed to respectively study spatial dependencies and temporal correlations, which may not fully learn the space-time relevance in traffic data. Moreover, they use the same module for traffic flow series, so they cannot study heterogeneities well.

Spatial-temporal synchronous models [6], [7], [8] are proposed to capture spatial-temporal dependencies in traffic data simultaneously. They construct multiple localized spatial-temporal graphs for the traffic data series and apply the GCN module to graphs. So they only use one component to capture temporal and spatial correlations synchronously, which simplifies the model structure. Though they have better performance than previous models, there are still some shortcomings in them. Firstly, they construct the adjacent matrix of spatial-temporal graphs based on road network structure. It may not comprehensively reflect the correlations between nodes. Secondly, in each graph convolutional layer, they consider correlations of the same nodes only adjacent in time and the relevancies between nodes only adjacently in space. Although stacking several graph convolutional layers can attain a larger spatiotemporal receptive field, the spatial-temporal range is limited by the number of layers. At the same time, the number of graph convolutional layers cannot be too large because of the over-smoothing phenomenon in GCNs. So they cannot directly model the latent relevancies among nodes far apart in time and space. Finally, they just use the result of the last graph convolutional layer to predict traffic flow, which means only fixed-scale space-time information is utilized. It ignores the phenomenon that future traffic flow is influenced by different scale spatial-temporal information, which results in their inferior short-term prediction performance.

In this paper, we propose a model called Adapted Spatial-Temporal Fusion Graph Convolutional Network (ASTFGCN) to tackle the above weaknesses. At first, this model can find the latent correlations among nodes to adjust the graph structure of the spatial-temporal graph. Secondly, it provides a way to construct spatial-temporal graphs of different time spans, which helps attain a larger spatiotemporal receptive field. At last, multi-scale spatial-temporal information fusion is offered to help nodes study the impacts of different spatial-temporal information on themselves.

Overall, the contributions of our work are as follows:

- We utilize an adaptive matrix to adjust the structure of the spatial-temporal graph. The matrix overlapped on the road network structure to get the adjacent matrix of the spatial-temporal graph. It can help find the cross-time, cross-space correlations among nodes in the spatial-temporal graph. Moreover, spatial-temporal graphs of different time spans are built by the dilated sliding window, which helps nodes attain a larger spatial-temporal receptive field.
- We introduce a multi-scale spatial-temporal information fusion mechanism. The results of all graph convolutional layers are retained so that the model can capture the influences of different spatial-temporal information for nodes.

II. Relative Works

Traffic data forecasting is a significant research direction of spatial-temporal data prediction. Traffic data prediction has been extensively studied for a few decades. Early methods such as the historical average model [9] do not require any assumption for traffic data. So their computations are simple and fast but have low performance. Early simple statistical methods, such as Vector Autoregression (VAR) [1], Autoregressive Integrated Moving Average (ARIMA) [10] and its variant [11], just model data based on data stationary assumption, so they cannot capture nonlinearity and uncertainty characteristics in traffic data. People utilized methods based on machine learning to solve these problems, such as support vector regression (SVR) [2], k-nearest neighbors (KNN) [12], and Kalman filtering (KF) [13]. With the development of deep learning, models based on deep neural networks are utilized for traffic forecasting to capture more complex temporal correlations. However, these methods only take temporal dependencies into account and ignore spatial correlations.

Spatial factors are considered in models to enhance forecast accuracy. For spatial-temporal grid data, Shi et al. [14] proposed a model called ConvLSTM that combines Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) to model the temporal correlations and spatial dependencies, respectively. Zhang et al. [15] designed a deep learning model to predict citywide crowd flows by using a convolution-based residual network. Guo et al. [16] proposed a model called ST-3DNet, which introduces 3D convolutions to effectively capture the localized spatial-temporal relationship. However, these methods can not be applied to graph-structured data since CNN is suitable for Euclidean data.

For spatial-temporal network data, it is not easy to utilize traditional convolution neural networks on the graphs. Graph convolution networks [17], [18] are proposed for graph-structured data and widely employed in node classification, link prediction, and graph classification. Traffic networks can be naturally represented as graphs, so many researchers utilize GCN to model spatial dependencies. DCRNN [4] uses diffusion convolution to study spatial correlations and captures temporal dependencies by Gated Recurrent Unit (GRU). Zhao et al. [19] used graph convolution to capture spatial dependencies and utilized GRU to model temporal dynamics. Pan et al. [20] proposed a sequence-to-sequence architecture that combines the graph attention network (GAT) [21] with the Recurrent Neural Networks (RNN) to model spatiotemporal dependencies. However, above methods based on RNN are prone to gradient explosion or gradient disappearance in backward transmission. Yu et al. [5] proposed a full convolution

model for traffic forecasting, which uses 1D convolution to extract the short-term temporal dependences and utilizes graph convolution to capture spatial correlations. Graph WaveNet [22] introduces WaveNet [23] to capture long-term temporal dependencies quickly. Some models introduce the attention mechanism to model complex spatial-temporal correlations. ASTGCN [24] designs two attention layers to model dynamic temporal dependencies and spatial correlations. GMAN [25] constructs the spatiotemporal embedding of nodes, then utilizes the multi-head attention mechanism to respectively learn the spatial dependence and temporal dependence of nodes which have been grouped. AGCRN [26] adaptively learns the relationship between nodes through the node embedding matrix not the road network structure, and tries to adaptively learn the corresponding traffic pattern for each node. However, it should artificially predefine the number of traffic patterns in advance. The number of traffic patterns is difficult to determine so the model is challenging to effectively learn all traffic patterns. Most of these methods utilize two components to capture temporal dependencies and spatial correlations, respectively. It may not fully learn the space-time relevance in traffic data. At the same time, they use the same GCN module for traffic flow series, which cannot study heterogeneities well.

Recently, some models are proposed to capture the spatiotemporal dependencies concurrently. STG2Seq [7] uses gated residual GCN modules to model the spatial-temporal correlations, but they only concatenate the nodes' features at adjacent time steps, which cannot comprehensively extract the spatiotemporal dependencies. To capture heterogeneities in traffic data, STSGCN [6] constructs localized spatial-temporal graphs and deals with each spatial-temporal graph separately, so it can study the localized spatial-temporal correlations and model the heterogeneities well. However, the model only considers localized spatial-temporal correlations and cannot find the latent relationships between nodes to adaptively adjust the spatial-temporal graph structure. In addition, it ignores that diverse nodes are influenced by different scale spatial-temporal information, which limits its learning ability. Li et al. [27] proposed STFGNN model which uses Dynamic Time Warping (DTW) algorithm to find latent correlation between nodes. It learns spatiotemporal dependencies by spatial-temporal fusion graphs. USTGCN [8] generates a spatial-temporal graph for the input series to directly capture spatiotemporal dependencies. When considering a longer input series, it needs to construct a large graph. So it should stack more GCN layers to be able to extract spatial-temporal dependencies between distant nodes, which increases the model complexity.

III. Methodology

A. Problem Definition

For traffic flow forecasting, the road network can be described as a spatial graph defined as $\mathcal{G}_S = (V, E, A_S)$. V is the set of nodes in the transportation network. E is the edge set of the graph, and the edges of the graph can be described by the adjacent matrix A_S. A_S is constructed by the nodes proximity or distance. $|V| = N$ is the number of road segments (or nodes). Edges can be directed or undirected. The adjacent matrix A_S does not change over time.

Historical traffic flow series of past T time steps is defined as $(\mathbf{X}^{(i-T+1)}, \mathbf{X}^{(i-T+2)}, ..., \mathbf{X}^{(i)})$. $\mathbf{X}^{(t)} \in \mathbb{R}^{N\times 1}$ is a traffic flow data matrix for n road segments at t-th time step. Function f uses historical observations to forecast the traffic flow of the next T' time steps. This process can be formulated as follow:

$$(\mathbf{X}^{(i+1)}, ..., \mathbf{X}^{(i+T')}) = f_\theta(\mathcal{G}_S; \mathbf{X}^{(i-T+1)}, ..., \mathbf{X}^{(i)}) \quad (1)$$

where θ is the set of learnable parameters.

B. Overview of ASTFGCN

Fig. 2 shows the architecture of ASTFGCN model. The input is the traffic flow of past T time steps. The model consists of an input layer, L Adapted Cross-Neighbourhood Graph Convolutional Layers (ACNGCLs), and a Spatial-Temporal Fusion Prediction Layer (STFPL). The input layer uses a fully-connect layer to expand nodes' expression ability. Each ACNGCL uses a Spatial-Temporal Graph Dilated-Sampling Module (STGDSM) to construct the spatial-temporal graphs and utilizes multiple independent Spatial-Temporal Synchronous Graph Convolutional Modules (STSGCMs) to capture spatial-temporal dependencies in graphs. At last, STFPL fuses multi-scale spatial-temporal information for forecasting. It uses T' prediction modules to forecast the traffic flow of the next T' time steps. Each prediction module consists of two fully-connect layers to predict traffic flow at a time step.

C. Adapted Cross-Neighbourhood Graph Convolutional Layer

Adapted Cross-Neighbourhood Graph Convolutional Layers (ACNGCLs) are used to effectively capture spatial-temporal dependencies. Each ACNGCL has a Spatial-Temporal Graph Dilated-Sampling Module (STGDSM) and multiple Spatial-Temporal Synchronous Graph Convolution Modules (STSGCMs). Firstly, the input series is split into several spatial-temporal graphs in the STGDSM. Then an STSGCM uses graph convolution operations for a spatial-temporal graph. At last, the outputs of all STSGCMs are formed into a new series that will be passed to the next layer.

1) Spatial-Temporal Graph Dilated-Sampling Module: Spatial-Temporal Graph Dilated-Sampling Module (STGDSM) constructs the adjacent matrix and samples the graph signals to construct spatial-temporal graphs. Thus there are two steps in the STGDSM: dilated sliding window sampling and spatial-temporal graph structure learning.

In the dilated sliding window sampling, multiple spatial-temporal series are constructed by the dilated sliding window. The dilated sliding window is utilized to attain larger spatiotemporal receptive field. It selects graph signals of three time steps for constructing a spatial-temporal graph. When the graph signal of the i-th time step is selected, the graph signals of the $(i + d)$-th time step and the $(i + 2d)$-th time step are also be selected where d is a dilated coefficient of the dilated sliding window. A spatial-temporal series is composed of these three graph signals. Thus, for the input series $\mathbf{H}_{l-1}$ of K time

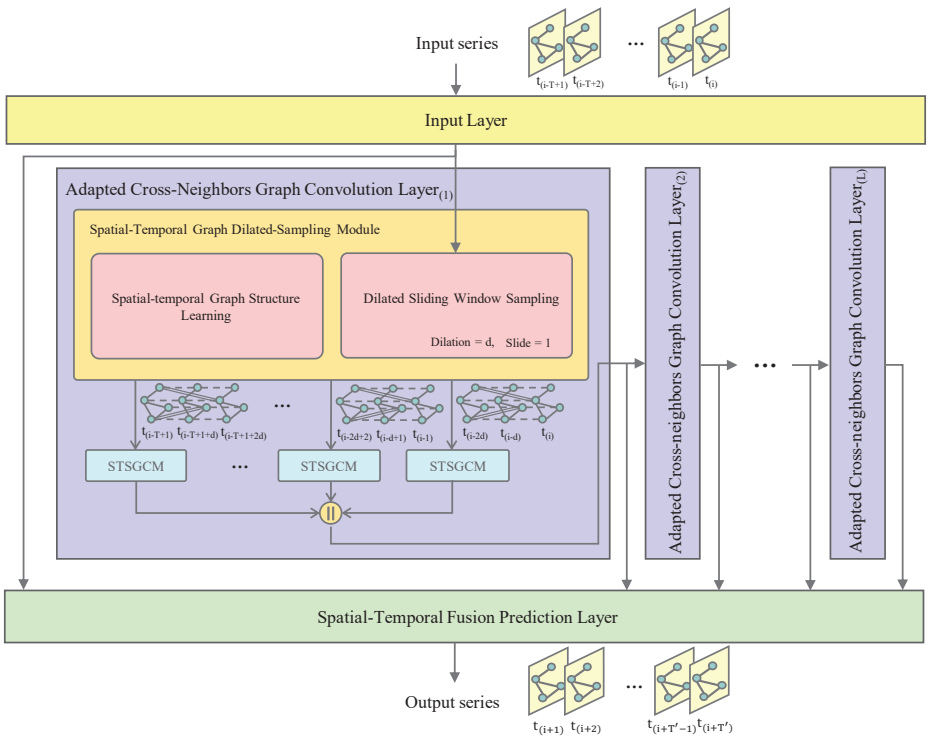


Fig. 2. The architecture of ASTFGCN.

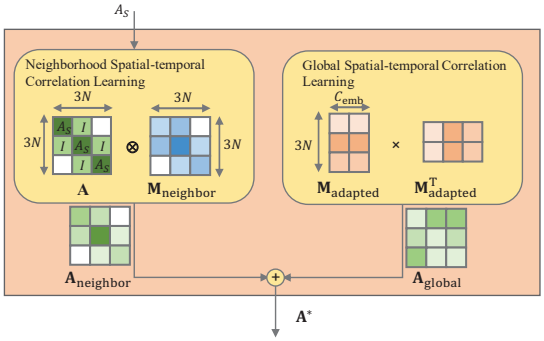


Fig. 3. Spatial-Temporal Graph Structure Learning

steps, the sliding window can cut out $K-2d$ spatial-temporal series in the l-th ACNGCL.

The spatial-temporal graph structure learning is utilized to construct the adjacent matrix of the spatial-temporal graphs. As shown in Fig. 3, it consists of two parts: neighborhood spatial-temporal correlation learning and global spatial-temporal correlation learning.

Neighborhood spatial-temporal correlation learning utilizes the method of previous works [5], [6]. In this step, nodes are connected when adjacent in time and space. Thus nodes at the middle time step connect themselves at other two time steps, and two nodes are connected if the distance between two nodes is less than the predetermined threshold Dis. So a simple adjacent matrix $\mathbf{A} \in \mathbb{R}^{3N\times 3N}$ is got. If a node v_i connects another node v_j, the item $\mathbf{A}_{i,j}$ is 1 otherwise 0. So $\mathbf{A}$ can be composed by the adjacent matrix of the spatial graph $\mathbf{A}_S$ and the identity matrix $\mathbf{I}$ shown in Fig. 3. At last, an element-wise product of $\mathbf{A}$ and a learnable edge-weight matrix $\mathbf{M}_{neighbor}$ is done to adjust the edges' weight and get $\mathbf{A}_{neighbor} \in \mathbb{R}^{3N\times 3N}$.

However, $\mathbf{A}_{neighbor}$ is constructed by considering correlations of nodes adjacently in time and space in this step. Only using $\mathbf{A}_{neighbor}$ cannot directly model the latent relevancies of nodes far apart in time and space. Although $\mathbf{A}_{neighbor}$ can adjust edges' weight by $\mathbf{M}_{neighbor}$, it can only affect exited edges, and the structure of the spatial-temporal graph will not change. To address above problems, the model uses the global spatial-temporal correlation learning to find the latent edges in the spatiotemporal graph.

In the global spatial-temporal correlation learning, a learnable matrix $\mathbf{A}_{global} \in \mathbb{R}^{3N\times 3N}$ is introduced to find the cross-time, cross-space correlations among nodes. The item $\mathbf{A}^{(i,j)}_{global}$ denotes the correlation coefficient between node i and node j. By training $\mathbf{A}_{global}$ with historical traffic flow data, the correlation coefficient of traffic flow change between any two spatiotemporal nodes can be obtained so that new edges can be found in the spatial-temporal graph. Nevertheless, the size of $\mathbf{A}_{global}$ is affected by the number of nodes. To reduce the parameters of $\mathbf{A}_{global}$, we multiply a learnable matrix $\mathbf{M}_{adapted} \in \mathbb{R}^{3N\times C_{emb}}$ and the transpose of $\mathbf{M}_{adapted}$ to get $\mathbf{A}_{global}$. Each row of $\mathbf{M}_{adapted}$ can be regarded as the node embedding representation of the traffic flow change. C_{emb}

is the dimension of node embedding, and it is less than the number of nodes.

Finally, the module adds $\mathbf{A}_{neighbor}$ and $\mathbf{A}_{global}$ to get the final adjacent matrix $\mathbf{A}^*$ that can effectively reflect the correlations of nodes in the spatial-temporal graph. It can be formulated as:

$$\mathbf{A}^* = \mathbf{A}_{neighbor} + \mathbf{A}_{global} = \mathbf{A} \otimes \mathbf{M}_{neighbor} + \mathbf{M}_{adapted} \cdot \mathbf{M}_{adapted}^T \tag{2}$$

where $\otimes$ is the element-wise product.

2) Node Embedding Learning of Spatial-Temporal Graphs: After dilated sliding window sampling and spatial-temporal graph structure learning, the local spatial-temporal graphs can be described by the local spatial-temporal series and the adjacent matrix $\mathbf{A}^*$. Compared to previous works, the spatial-temporal graphs are constructed by considering the local and global correlations so that the model can learn more spatial-temporal dependencies. To learn node embedding, the component called STSGCM of previous work [6] is used in our model.

An STSGCM has three graph convolutional layers, a max-pooling layer and a cropping layer. The graph signal $\mathbf{h}_0 \in \mathbb{R}^{3N \times C}$ and the adjacent matrix $\mathbf{A}^*$ are inputs for an STSGCM. Each graph convolutional layer uses the graph convolution operations to study spatial-temporal correlations. The graph convolution operation can be described as follow:

$$\mathbf{h}_i = (\mathbf{A}^* \mathbf{h}_{i-1} \mathbf{W}_1 + \mathbf{b}_1) \otimes sigmoid(\mathbf{A}^* \mathbf{h}_{i-1} \mathbf{W}_2 + \mathbf{b}_2) \tag{3}$$

where $sigmoid$ is the sigmoid function. The max-pooling layer utilizes element-wise max operation for all graph convolutional layers' output to get $\mathbf{h}_{max} \in \mathbb{R}^{3N \times C}$. Lastly, the cropping layer preserves the graph signal in the middle time step to get the output $\mathbf{h} = \mathbf{h}_{max}[N : 2N, :] \in \mathbb{R}^{N \times C}$.

Spatial-temporal graphs are constructed by the input series. Because of the heterogeneities in traffic data, each ACNGCL utilizes a group of STSGCMs to capture the spatial-temporal correlations of spatial-temporal graphs. An independent STSGCM is applied to a spatial-temporal graph, so an ACNGCL consists of $K - 2d$ STSGCMs. The i-th spatial-temporal graph's node embedding can be described as $\mathbf{h}^i \in \mathbb{R}^{3 \times N \times C}$, and it is reshaped to $\mathbf{h}^i_{reshape} \in \mathbb{R}^{3N \times C}$ that is the input of the i-th STSGCM. The outputs of all STSGCMs compose a new series $\mathbf{H}_l$, so the spatiotemporal dependencies of the complete input series can be learned. $\mathbf{H}_l$ is the output of the l-th ACNGCL and the input of the $(l+1)$-th ACNGCL. It can be formulated as follow:

$$\mathbf{H}_l = [\mathbf{H}_l^1, \mathbf{H}_l^2, ..., \mathbf{H}_l^{K_l - 2d_l}] \in \mathbb{R}^{(K_l - 2d_l) \times N \times C} \tag{4}$$

where $\mathbf{H}_l^i$ is the output of the i-th STSGCM, K_l is the length of the input series and d_l is the size of dilated coefficient in the l-th ACNGCL.

D. Spatial-temporal Fusion Prediction Layer

By stacking multiple ACNGCLs, our model can learn the long-term spatial-temporal correlations. To avoid the information loss between ACNGCLs and effectively utilize multi-scale spatial-temporal information, STFPL concatenates the outputs of all ACNGCLs and the input layer to a matrix $\mathbf{H}_{final}$. $\mathbf{H}_{final}$ can be formulated as follow:

$$\mathbf{H}_{final} = [\mathbf{H}_0, \mathbf{H}_1, \mathbf{H}_2, ..., \mathbf{H}_L] \in \mathbb{R}^{K' \times N \times C} \tag{5}$$

where $\mathbf{H}_0$ is the output of the input layer and L is the number of ACNGCLs. $\mathbf{H}_{final}$ is the input of all prediction modules, so modules can capture the influences of different scale spatial-temporal information for nodes. The size of $K^{'}$ can be computed as $K^{'} = K_1 + \sum_{l=1}^{L} K_l - 2 * d_l$.

This layer has $T^{'}$ prediction modules. The i-th prediction module is used to predict the traffic flow of the future i-th time step. So our model avoids error propagation by the multi-step prediction. Each prediction module has two fully connected layers. In the i-th prediction module, the input $\mathbf{H}_{final} \in \mathbb{R}^{K^{'} \times N \times C}$ is reshaped to $\mathbf{H}_{reshape} \in \mathbb{R}^{N \times K^{'} C}$. Then $\mathbf{H}_{reshape}$ is feed into two fully connected layers to get the prediction $\hat{\mathbf{y}}^{(i)} \in \mathbb{R}^{N \times 1}$. This process can be formulated as follow:

$$\hat{\mathbf{y}}^{(i)} = ReLU(\mathbf{H}_{reshape} \mathbf{W}_1^{(i)} + \mathbf{b}_1^{(i)}) \mathbf{W}_2^{(i)} + \mathbf{b}_2^{(i)} \tag{6}$$

At last, the layer concatenates the predictions of all time steps into a matrix to get the output $\hat{\mathbf{Y}}$ of ASTFGCN:

$$\hat{\mathbf{Y}} = [\hat{\mathbf{y}}^{(1)}, \hat{\mathbf{y}}^{(2)}, ..., \hat{\mathbf{y}}^{(T^{'})}] \in \mathbb{R}^{N \times T^{'}} \tag{7}$$

E. Loss Function

Huber loss is used as the loss function. The Huber loss is less sensitive to outliers in data than the squared error loss, and it can be formulated as follow:

$$L(\mathbf{Y}, \hat{\mathbf{Y}}) = \begin{cases} \frac{1}{2}(\mathbf{Y} - \hat{\mathbf{Y}})^2, & |\mathbf{Y} - \hat{\mathbf{Y}}| \leq \delta \\ \delta |\mathbf{Y} - \hat{\mathbf{Y}}| - \frac{1}{2}\delta^2, & otherwise \end{cases} \tag{8}$$

where $\mathbf{Y}$ is the ground truth, $\hat{\mathbf{Y}}$ denotes the prediction of our model and δ is a threshold that judges whether the squared error loss is anomalous.

IV. Experiments

A. Dataset

We conduct our experiments on four real-world datasets from the Caltrans Performance Measurement System (PeMS) [28], namely PEMS03, PEMS04, PEMS07 and PEMS08. The description of these datasets is shown in Table I.

TABLE I
DATASET DESCRIPTION.

Datasets	Number of nodes	Time range
PEMS03	358	9/1/2018-11/30/2018
PEMS04	307	1/1/2018-2/28/2018
PEMS07	883	5/1/2017-8/31/2017
PEMS08	170	7/1/2016-8/30/2016

The standard time interval of all dataset is 5 minutes, which means there are 12 points each hour in the flow data. We standardize the feature, which converts the data distribute to a normal distribution.

B. Baselines

We select following models as baselines:

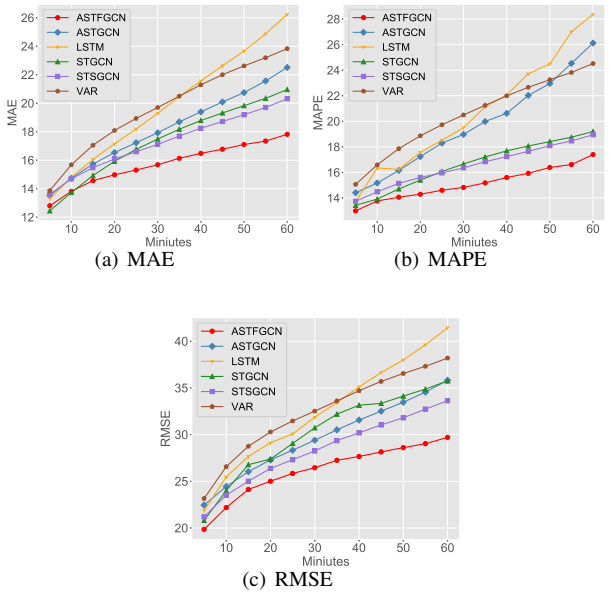
- **VAR** Vector Auto-Regression.
- **SVR** Support Vector Regression.
- **LSTM** Long Short-Term Memory Networks for time series prediction.
- **DCRNN** [4] Diffusion Convolutional Recurrent Neural Network utilizes diffusion graph convolutional to model spatial correlations and based-GRU seq2seq to capture temporal information.
- **STGCN** [5] Spatial-Temporal Graph Convolutional Networks uses ChebNet to study the spatial dependencies and 1D convolution to capture the temporal correlations.
- **ASTGCN** [24] Attention Based Spatial-Temporal Graph Convolutional Networks models the spatial-temporal dynamics by the attention mechanism. ASTGCN utilizes three components to capture the periodicity of highway traffic data. We only use its recent component in the experiments for fairness.
- **STSGCN** [6] Spatial-Temporal Synchronous Graph Convolutional Networks stacks multiple spatial-temporal synchronous layers. In each layer, independent modules are used for localized spatial-temporal graphs to capture localized spatial-temporal dependencies.
- **STFGNN** [27] Spatial-Temporal Fusion Graph Neural Networks uses DTW algorithm to find the correlations among nodes and it learns spatial-temporal dependencies by constructing spatial-temporal fusion graphs.

C. Experimental settings

We use one-hour historical flow data to predict the data of the next hour. Each dataset is split into a training set, a validation set, and a testing set according to the ratio 6: 2: 2. For each dataset, the embedding size C_{emb} of the embedding matrix $\mathbf{M}_{adapted}$ is 128. After getting the output of the input layer, we do the padding operation for this output in the time dimension, so the size (K_1) of the time dimension is 13. Four ACNGCLs are used in our model, and the dilated coefficients of sliding windows for four ACNGCLs are 1, 2, 2, 1, respectively. All STGDSMs share one Spatial-temporal Graph Structure Learning. Each STSGCM utilizes three graph convolution layers, and the number $(c^{'})$ of kernels for each Graph convolutional operation is 64.

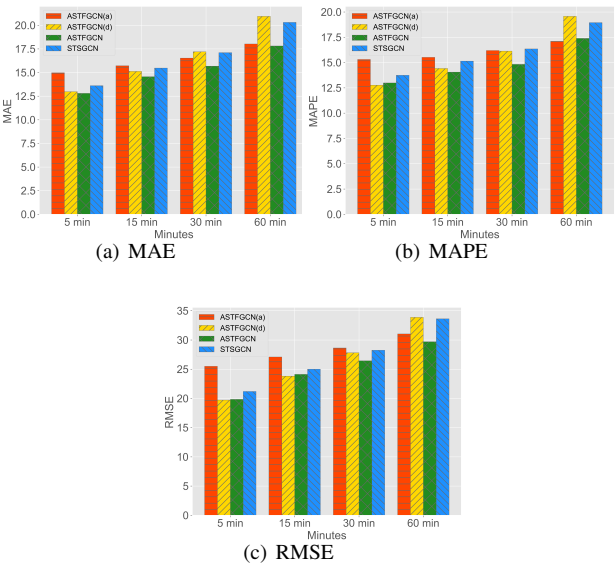
D. Experimental Results

1) Prediction Performance Comparison: As shown in Table II, our model outperforms all baseline models in four datasets: (1) VAR, SVR and LSTM show poor performance because they only take temporal dependencies into account. (2) DCRNN, STGCN and ASTGCN consider both temporal dependencies and spatial correlations, making them have better performance than previous models. (3) STSGCN uses GCN to model the localized spatial-temporal dependencies and utilizes multiple modules to capture the heterogeneities in localized spatial-temporal graphs, so it achieves better performance than above models. STFGNN constructs new spatial-temporal graphs based on new information, but it cannot comprehensively represent correlations among nodes. (4) Our model further considers more complex spatial-temporal correlations in the spatial-temporal graphs and efficiently fuses the multi-scale spatiotemporal information, which results in that our model outperforms all baseline models. We also compare the prediction performances of some models at different time steps on the PEMS03 dataset. As shown in Fig. 4, our model achieves better short-term and long-term forecasting performance than others.



(a) MAE (b) MAPE (c) RMSE

Fig. 4. Prediction performance comparison at each time step on the PEMS03 dataset.



(a) MAE (b) MAPE (c) RMSE

Fig. 5. Component analysis of ASTFGCN on the PEMS03 dataset

TABLE II
OVERALL PERFORMANCE OF DIFFERENT METHODS FOR TRAFFIC FLOW FORECASTING

Models		VAR	SVR	LSTM	DCRNN	STGCN	ASTGCN(r)	STSGCN	STFGNN	ASTFGCN
Dataset	Metrics									
PEMS03	MAE	23.65	21.97	21.33	18.18±0.15	17.46±0.32	17.69±1.43	17.48±0.15	16.77	**15.59±0.15**
	MAPE(%)	24.51	21.51	23.33	18.91±0.82	16.77±0.47	19.40±2.24	16.78±0.2	16.30	**14.90±0.25**
	RMSE	38.26	35.19	35.11	30.31±0.25	30.25±0.60	29.66±1.68	29.21±0.56	28.34	**26.09±0.55**
PEMS04	MAE	23.75	28.70	27.14	24.70±0.22	22.24±0.31	22.93±1.29	21.19±0.10	19.83	**19.62±0.10**
	MAPE(%)	18.09	19.20	18.20	17.12±0.37	14.40±0.21	16.56±1.36	13.90±0.05	13.02	**12.85±0.10**
	RMSE	36.66	44.56	41.59	38.12±0.26	35.11±0.24	39.70±0.04	33.65±0.20	31.88	**31.50±0.20**
PEMS07	MAE	75.63	32.49	29.98	25.30±0.52	24.67±0.64	28.05±2.34	24.26±0.14	22.10	**21.82±0.10**
	MAPE(%)	32.22	14.26	13.20	11.66±0.33	10.13±0.35	13.92±1.65	10.21±0.05	**9.21**	9.30±0.04
	RMSE	115.24	50.22	45.84	38.58±0.70	38.67±0.90	42.57±3.31	39.03±0.27	35.80	**35.08±0.22**
PEMS08	MAE	23.46	23.25	22.20	17.86±0.03	17.93±0.25	18.61±0.40	17.13±0.09	16.64	**15.90±0.25**
	MAPE(%)	15.42	14.64	14.20	11.45±0.03	11.48±0.43	13.08±1.00	10.96±0.07	10.60	**10.20±0.16**
	RMSE	36.33	36.16	34.06	27.83±0.05	28.24±0.40	28.16±0.48	26.80±0.18	26.22	**24.87±0.38**

TABLE III
COMPONENT ANALYSIS OF ASTFGCN ON THE PEMS03 DATASET

Models	30 Min			60 Min			Average		
	MAE	MAPE(%)	RMSE	MAE	MAPE(%)	RMSE	MAE	MAPE(%)	RMSE
ASTFGCN(a)	16.59	16.56	28.86	17.98	17.23	31.03	16.63	16.50	28.86
ASTFGCN(d)	17.24	16.26	28.03	21.02	19.79	34.06	17.36	16.42	28.34
ASTFGCN	15.68	14.82	26.44	17.81	17.39	29.69	15.59	14.90	26.09

2) Ablation Study: To prove the effectiveness of our improved model, we do ablation studies on the PEMS03 dataset. Two variants of ASTFGCN are designed, namely ASTFGCN(a) and ASTFGCN(d). ASTFGCN(a) retains the adaptive matrix. It uses the output of the last ACNGCL as the input of the prediction layer, and the dilated coefficients are all 1. ASTFGCN(d) discards the adaptive matrix compared to ASTFGCN. The result is shown in Table III. ASTFGCN(a) and ASTFGCN(d) both outperform STSGCN on overall performance. More detailed performance comparison of these models is shown in Fig. 5. Although ASTFGCN(d) and ASTFGCN(a) have better overall performance than STSGCN, they have diverse performance on short-term and long-term predictions. ASTFGCN(a) has better long-term forecasting performance than STSGCN and ASTFGCN(d), but it gets worst results than the other three models on short-term prediction because it just uses the result of the last ACNGCL for multi-step forecasting. Since ASTFGCN(d) utilizes the multi-scale spatial-temporal information, it has better short-term prediction performance than STSGCN and ASTFGCN(a). Although dilated sliding windows help expand nodes' receptive filed, the adjacent matrix cannot comprehensively reflect correlations of nodes in the spatial-temporal graph, which results in worse long-term prediction of ASTFGCN(d). Our model combines the advantages of ASTFGCN(a) and ASTSGCN(d), so it almost outperforms the other three models at all time steps.

We compare the prediction performance of our model by using different embedding sizes of C_{emb}. $[8, 16, 32, 64, 128]$ are used as the embedding size in the experiments, respectively, and the result is shown in Fig. 6. There are no apparent differences on three metrics, but the iteration time to get the best validation performance varies with different embedding sizes, as shown in Table IV. Overall, the iteration time decrease with the increase of embedding size.

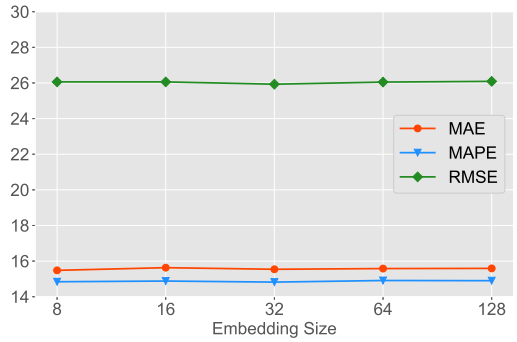


Fig. 6. Overall performance comparison with different embedding size on the PEMS03 dataset.

TABLE IV
THE ITERATION TIME TO GET THE BEST VALIDATION MODEL WITH DIFFERENT EMBEDDING SIZES ON PEMS03 DATASET.

Embedding Size	8	16	32	64	128
Iteration Time(epochs)	46±9	49±6	41±7	35±4	31±5

3) Training Efficiency: We compare the the computational cost of DCRNN, ASTGCN(r), ASTFGCN and STSGCN. These models are implemented by PyTorch. The result is shown in Table V. The training time is the time that the model takes to train one epoch. The fastest model is ASTGCN(r), followed by ASTFGCN, STSGCN and DCRNN. Although ASTFGCN has best prediction performance, it doesn't run fastest time. ASTFGCN needs to capture spatial-temporal

dependencies on each spatial-temporal graphs, which increases its time cost.

TABLE V
THE TRAINING TIME ON THE PEMS03 DATASET.

Models	DCRNN	ASTGCN(r)	STSGCN	ASTFGCN
Training Time(s/epoch)	387	95	166	150

V. CONCLUSION

We propose a model called ASTFGCN, which improves some shortcomings in previous works. Firstly, the model provides a learnable mechanism to adjust the spatial-temporal graph structure. So it can find the cross-time, cross-space correlations among nodes in the spatial-temporal graph. Secondly, the spatial-temporal graphs are constructed by the dilated sliding window, which helps attain a larger spatiotemporal receptive field. At last, the model fuses multi-scale spatial-temporal information to find the influence of different spatial-temporal ranges for diverse spots. Our experiments are conducted on four real-world datasets. Experimental results show that the prediction performance of ASTFGCN is better than baselines. Finally, because ASTFGCN is a general framework for spatiotemporal network data prediction, it can also be applied in other related applications, such as air quality forecasting.

REFERENCES

[1] E. Zivot and J. Wang, "Vector autoregressive models for multivariate time series," *Modeling financial time series with S-PLUS*, pp. 385–429, 2006.

[2] H. Drucker, C. J. Burges, L. Kaufman, A. Smola, and V. Vapnik, "Support vector regression machines," *Advances in Neural Information Processing Systems*, vol. 9, pp. 155–161, 1996.

[3] X. Luo, D. Li, Y. Yang, and S. Zhang, "Spatiotemporal traffic flow prediction with knn and lstm," *Journal of Advanced Transportation*, 2019.

[4] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: data-driven traffic forecasting," in *International Conference on Learning Representations*, 2018.

[5] B. Yu, H. Yin, and 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.

[6] C. Song, Y. Lin, S. Guo, and 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.

[7] L. Bai, L. Yao, S. S. Kanhere, X. Wang, and Q. Z. Sheng, "Stg2seq: spatial-temporal graph to sequence model for multi-step passenger demand forecasting," in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019*, 2019.

[8] A. Roy, K. K. Roy, A. A. Ali, M. A. Amin, and A. Rahman, "Unified spatio-temporal modeling for traffic forecasting using graph neural network," in *Proceedings of International Joint Conference on Neural Networks*, 2021.

[9] J. Liu and W. Guan, "A summary of traffic flow forecasting methods," *Journal of Highway and Transportation Research and Development*, vol. 3, pp. 82–85, 2004.

[10] B. M. Williams and L. A. Hoel, "Modeling and forecasting vehicular traffic flow as a seasonal arima process: theoretical basis and empirical results," *Journal of transportation engineering*, vol. 129, no. 6, pp. 664–672, 2003.

[11] J. Chen, D. Li, G. Zhang, and X. Zhang, "Localized space-time autoregressive parameters estimation for traffic flow prediction in urban road networks," *Applied Sciences*, vol. 8, no. 2, p. 277, 2018.

[12] Z. Zheng and D. Su, "Short-term traffic volume forecasting: a k-nearest neighbor approach enhanced by constrained linearly sewing principle component algorithm," *Transportation Research Part C: Emerging Technologies*, vol. 43, pp. 143–157, 2014.

[13] M. Lippi, M. Bertini, and P. Frasconi, "Short-term traffic flow forecasting: an experimental comparison of time-series analysis and supervised learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 2, pp. 871–882, 2013.

[14] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo, "Convolutional lstm network: a machine learning approach for precipitation nowcasting," in *Advances in Neural Information Processing Systems*, 2015.

[15] J. Zhang, Y. Zheng, and D. Qi, "Deep spatio-temporal residual networks for citywide crowd flows prediction," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2017.

[16] S. Guo, Y. Lin, S. Li, Z. Chen, and H. Wan, "Deep spatial-temporal 3d convolutional neural networks for traffic data forecasting," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 10, pp. 3913–3926, 2019.

[17] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Advances in Neural Information Processing Systems*, 2016.

[18] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *International Conference on Learning Representations*, 2017.

[19] L. Zhao, Y. Song, C. Zhang, Y. Liu, P. Wang, T. Lin, M. Deng, and H. Li, "T-gcn: a temporal graph convolutional network for traffic prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 9, pp. 3848–3858, 2019.

[20] Z. Pan, Y. Liang, W. Wang, Y. Yu, Y. Zheng, and J. Zhang, "Urban traffic prediction from spatio-temporal data using deep meta learning," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019.

[21] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *International Conference on Learning Representations*, 2018.

[22] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, "Graph wavenet for deep spatial-temporal graph modeling," in *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, 2019.

[23] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: a generative model for raw audio," in *The 9th Speech Synthesis Workshop*, 2016.

[24] S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan, "Attention based spatial-temporal graph convolutional networks for traffic flow forecasting," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019.

[25] C. Zheng, X. Fan, C. Wang, and J. Qi, "Gman: a graph multi-attention network for traffic prediction," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020.

[26] L. BAI, L. Yao, C. Li, X. Wang, and C. Wang, "Adaptive graph convolutional recurrent network for traffic forecasting," in *Advances in Neural Information Processing Systems*, 2020.

[27] M. Li and Z. Zhu, "Spatial-temporal fusion graph neural networks for traffic flow forecasting," in *Proceedings of the AAAI conference on artificial intelligence*, 2021.

[28] C. Chen, K. Petty, A. Skabardonis, P. Varaiya, and Z. Jia, "Freeway performance measurement system: mining loop detector data," *Transportation Research Record*, vol. 1748, no. 1, pp. 96–102, 2001.