*Quest Journals*
*Journal of Software Engineering and Simulation*
*Volume 7 ~ Issue 12 (2021) pp: 06-19*
*ISSN(Online) :2321-3795 ISSN (Print):2321-3809*
www.questjournals.org

**Research Paper**

# Research on Traffic Flow Forecasting Based on Spatio-temporal Convolutional Network

Hong Zhang[1], Linlong Chen[2], Jie Cao[3], Yuanmeng Zhao[4], Sunan Kan[5], Tianxin Zhao[6]
*[1,2,3,4,5,6](College of Computer & Communication, Lanzhou University of Technology)*
*Corresponding Author: Hong Zhang*

***ABSTRACT:*** *Spatio-temporal graph modeling is an important task to analyze the spatio-temporal correlation of traffic flow. However, due to the complex spatio-temporal dependence and uncertainty caused by dynamic traffic environment conditions, accurate and real-time traffic flow forecasting has always been a difficulty. In order to address this problem, this paper builds a spatio-temporal convolutional network (TS-GCN) model for traffic flow forecasting, which models the time dependence, spatial correlation and long time series of traffic flow respectively. It uses temporal convolutional network (TCN) and graph convolutional network (GCN) to capture the temporal dependence and spatial correlation of traffic flow, respectively. STA-Block module models the spatio-temporal correlation of long-term sequences through spatio-temporal attention mechanism and gating fusion mechanism. Construct an adaptive adjacency matrix in the TS-GCN model and learn through node embedded to capture the dynamic spatial correlation of traffic flow. The experimental results on the METR-LA and PEMS-BAY data sets demonstrate that, compared with the baseline methods, the TS-GCN model proposed in this paper has the best forecasting performance.*

***KEYWORDS:*** *Traffic flow forecasting, Deep learning, Temporal convolutional network, Graph convolutional network*

***R****eceived 08 Dec, 2021;* ***R****evised 21 Dec, 2021;* ***A****ccepted 23 Dec, 2021 © The author(s) 2021.*
***P****ublished with open access at* ***www.questjournals.org***

## I. INTRODUCTION

Traffic flow forecasting aims to forecast future traffic conditions in the road network based on historical observations (For example, traffic volume and speed). As an important part of the Intelligent Transportation System (ITS)[1], traffic forecasting can not only provide a scientific basis for traffic management personnel to perceive traffic congestion and restrict vehicle driving in advance, but also provide urban passengers with a choice of appropriate travel routes and improve travel efficiency[2].

In recent years, deep learning models with good learning capabilities and deep network structures have developed rapidly. Based on its excellent performance, deep learning models have made great progress in the field of traffic flow forecasting. For example, deep learning models such as Stacked Autoencoder (SAE)[3] and Deep Belief Network (DBN)[4]. With the development of graph neural networks, spatio-temporal graph modeling has received more and more attention. Spatio-temporal graph modeling has a wide range of applications in solving complex system problems, such as traffic speed forecasting[5], taxi demand forecasting[6] and human behavior recognition[7].

The basic assumption of spatio-temporal graph modeling is that the future information of a node depends on its historical information and the historical information of its neighbors. Therefore, how to capture spatial and temporal features at the same time has grown up to be a major challenge. At present, the research on spatio-temporal graph modeling mainly has two directions: one is Li[5] proposed integrating graph convolutional network (GCN) into recurrent neural network (RNN). The second is Yu[8] proposed integrating GCN into Convolutional Neural Network (CNN). Although these studies have proved the effectiveness of introducing the graph network structure of the data into the model, these methods still face two major challenges.

First of all, these studies assume that the graph structure of the data reflects the true dependencies between nodes. However, when the connection does not include the interdependence between the two nodes, and when the interdependence between the two nodes exists, there is a case where the connection is invalid.

Secondly, the current research on spatio-temporal graph modeling is not effective for learning time dependence. RNN-based methods are prone to gradient disappearance when capturing long-term sequences[9]. On the contrary, the CNN-based method has the advantage of a stable gradient. However, because CNN uses standard 1D convolution, its receptive field increases linearly with the increase in the number of hidden layers, which cause these models to use more layers to capture long-term sequences.

In this work, this paper proposes a traffic flow forecasting model based on spatio-temporal convolutional network (TS-GCN) to forecast traffic flow. The model includes a gated temporal convolutional network module, a graph convolutional network module and a spatio-temporal attention module (ST-Attention Block), each ST-Attention Block consists of a spatial attention mechanism (used to model dynamic spatial correlation), a temporal attention mechanism (used to model nonlinear temporal correlation), and a gated fusion mechanism (used to adaptively fuse spatial and time representation). At the same time, this paper proposes a graph convolutional network, which obtains an adaptive adjacency matrix from the data through an end-to-end supervised training method. The adaptive adjacency matrix captures the spatial correlation hidden in the traffic flow data. This paper uses stacked dilated casual convolution networks to capture time dependence. The receptive field of stacked dilated casual convolution networks grows exponentially as the number of hidden layers increases. With support of dilated casual convolution networks, the TS-GCN model can effectively process spatio-temporal graph data with long-term sequences.

The main contributions of this paper are summarized as follows:

1. This article constructs an adaptive adjacency matrix to learn the hidden spatial dependencies. The adaptive adjacency matrix can find hidden dynamic graph network structure from the data without any prior knowledge guide.
2. This paper proposes an effective framework to capture temporal and spatial dependence at the same time. This paper combines the proposed graph convolution network with dilated casual convolution networks, so that each graph convolution network processes dilated casual convolution networks of different granularity levels to extract the spatial characteristics of node information.
3. This paper introduces a spatial attention mechanism and time attention mechanism to learn the dynamic spatial correlation and nonlinear time dependence of traffic flow data respectively. In addition, this paper designs a gated fusion mechanism to adaptively fuse the information extracted through the spatio-temporal attention mechanism to reduce the iterative propagation of errors in the forecasting process.
4. This paper conducts a large number of comparative experiments on two sets of traffic data sets. The experimental results show that, compared with the existing baseline methods, the model in this paper has achieved better forecasting performance on different data sets.

## II. RELATED WORKS

Traffic flow forecasting has been extensively studied in the past few decades. As a typical deep neural network, CNN[10] has made many breakthroughs in image processing. Some researchers use CNN to capture spatial features in traffic flow data[11]. Gated recurrent unit network (GRU)[12] and long short-term memory neural network (LSTM)[13] are both good at processing time series and successfully applied to traffic flow forecasting. At the same time, some researchers have combined CNN and LSTM networks to propose a fusion framework CLTFP[14], which is employed to short-term traffic flow forecasting by combining CNN and LSTM networks. On this basis, Shi[15] proposed a convolutional LSTM network, which embeds the extended fully connected LSTM network into the convolutional LSTM network (FC-LSTM). Compared with the above methods, CLTFP and FC-LSTM can obtain better forecasting performance. CNN can effectively capture the spatial features in the data. However, when the traffic road network and the detectors deployed on the road network are used as the topology map, CNN cannot capture the spatial characteristics of the road network.

In addition, many researchers use GCN to forecast traffic flow[5]. Yu[8] proposed a spatio-temporal graph convolutional network (STGCN), which constructs a fixed Laplacian matrix based on the spatial distance between the detector nodes. Guo[16] proposed a spatio-temporal graph convolutional network (ASTGCN) based on the attention mechanism, and used the attention mechanism[17] to capture the dynamic correlation between nodes. However, the construction of the Laplacian matrix required by the GCN in the above method all depends on the spatial distance between the detector nodes in the road network, which causes the model to have great limitations.

As a new technology, attention mechanism has developed rapidly in recent years and is widely used in fields such as natural language processing and speech recognition. The core idea of the attention mechanism is to adaptively focus on the most relevant features according to the input data. For example, Bahdanau[18] proposed a calibration model to evaluate the match between input and output. On this basis, Feng[17] proposed a neural network structure composed of two memory networks, which can model the semantic association and relationship between each word and two entities. Based on the above research, Veličković[19] proposed a graph attention network (GAT). GAT does not need to know the structure of the graph in advance, but only needs to pay attention to the feature data of the nodes, and uses a self-attention mechanism to specify the weights between nodes. In traffic flow forecasting, in order to extract spatial features, Liu[20] performs a 2D convolution operation on each feature map to obtain the corresponding attention matrix. The maximum average pooling is performed on each feature map, and the result is used as the input of the feedforward neural network to obtain the channel attention. Although the above research uses the attention mechanism to capture the impact between the traffic network[9], the attention score only depends on the traffic speed information and ignores other factors.

Driven by the above research, consider the topological structure of the transportation network and the dynamic spatio-temporal correlation of traffic flow data. This paper uses both temporal convolutional network (TCN) and graph convolutional network (GCN) to model traffic flow data. In this paper, TCN and GCN are used to capture the temporal and spatial characteristics of traffic flow respectively, and the gated fusion mechanism is used to adaptively fuse the temporal and spatial information of traffic flow extracted by the temporal and spatial attention mechanism to reduce the propagation of errors in the forecasting process.

## III. METHODOLOGY

### A. Problem Definition

In this study, graph $G = (V, E)$ is used to describe the topological structure of the road network, and each sensor detection point is regarded as a node, and the connection relationship between any two sensors is regarded as an edge between two corresponding nodes. Among them, $V$ is a set of road nodes, $E$ is a set of edges, and $A \in R^{N \times N}$ represents the adjacency matrix of graph $G$. If $v_i$, $v_j \in V$ and $(v_i, v_j) \in E$, then $A_{ij}$ is 1, otherwise it is 0. In each time step $t$, the graph $G$ has a dynamic feature matrix $X^{(t)} \in R^{N \times D}$, and the traffic flow forecasting problem aims to learn a function $f$, which maps the historical graph signal $S$ to the future $T$ graph signal. Given a graph $G$, the mapping relationship is as following:

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

Among them, $X^{(t-S):t} \in R^{N \times D \times S}$ and $X^{(t+1):(t+T)} \in R^{N \times D \times T}$.

### B. Framework of TS-GCN

This paper shows in the overall framework of the TS-GCN model in Figure 1. It consists of an input layer, a stacked spatio-temporal convolutional network layer, ST-Conv Block, SA-Attention Block, and an output layer. Each network layer of the stack is skip-connected to ST-Conv Block. The spatio-temporal layer is composed of a GCN network and a gated time convolutional network (Gated TCN). Gated TCN is composed of two parallel time convolutional networks (TCN-a and TCN-b). ST-Conv Block contains three spatio-temporal convolution blocks, which capture many-to-one effects from three different angles corresponding to space, temporal, and spatio-temporal. SA-Attention Block combines spatial attention mechanism and temporal attention mechanism through gated fusion. By stacking multiple spatio-temporal layers, TS-GCN can handle spatial correlations of different temporal levels. The GCN can capture the spatial features in the historical traffic flow data. The input $h$ of the graph convolutional layer is a three-dimensional tensor of size $[N, C, L]$, where $N$ is the number of nodes, $C$ is the dimension of the hidden layer, and $L$ is the length of the data sequence. This paper applies GCN to each $h[:,:,i] \in R^{N \times C}$.

This paper chooses to use the mean absolute error (MAE) as the loss function of TS-GCN, which is defined as:

$$L\left( X^{(t+1):(t+T)}; \Theta \right) = \frac{1}{TND} \sum_{i=1}^{i=T} \sum_{j=1}^{j=N} \sum_{k=1}^{k=D} \left| X_{jk}^{(t+i)} - X_{jk}^{(t+i)} \right| \qquad (2)$$

The TS-GCN model in this paper outputs $X^{(t+1):(t+T)}$ as a whole, and the size of the receptive field of the TS-GCN model is designed to be equal to the length of the input sequence, so that in the final spatio-temporal layer, the time dimension of the output is equal to 1. In this paper, the number of output channels of the last layer is set as the step size to obtain the required dimensions of the output.

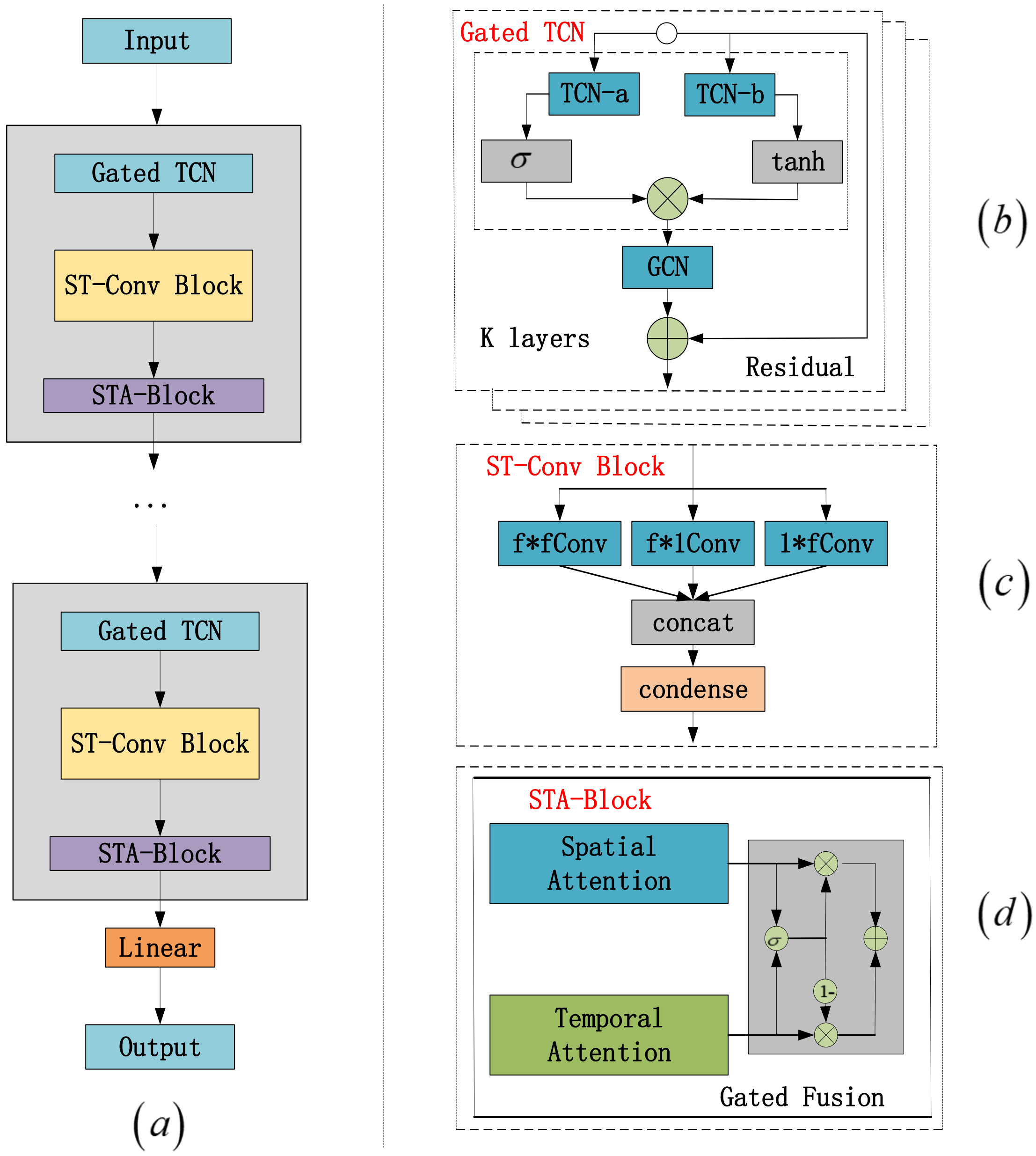

**Figure 1:** The overall framework of the TS-GCN model

### C. Temporal Convolution Network

This paper uses dilated casual convolution as a temporal convolutional network (TCN) to capture the temporal trend of nodes. Dilated casual convolution network achieves a larger receptive field by increasing the depth of the layer. Compared with the RNN method, dilated casual convolution network can correctly process long-term sequences, facilitate parallel computing, and alleviate the problem of gradient disappearance. Dilated casual convolution maintains temporal causal order by filling zeros into the input, so that forecasting made at the current time step only involve historical information. As a special case of standard 1D convolution, the dilated casual convolution operation slides the input by skipping values in specific steps, as showed in Figure 2. Mathematically, Given a one-dimensional sequence input $x \in R^{T}$ and filter $f \in R^{K}$, the expression of the expanded casual convolution operation of $x$ and $f$ at step $t$ is shown in equation (3):

$$x * f(t) = \sum_{s=0}^{K-1} f(s) x(t - d \times s) \quad (3)$$

Where $d$ is the dilation factor that controls the jump distance. By stacking the dilated casual convolution network with dilation factor in increasing order, the model's receptive field grows exponentially. It enables the dilated casual convolution network to capture longer sequences with fewer layers, thereby saving computational

*Corresponding Author: Hong Zhang

resources.

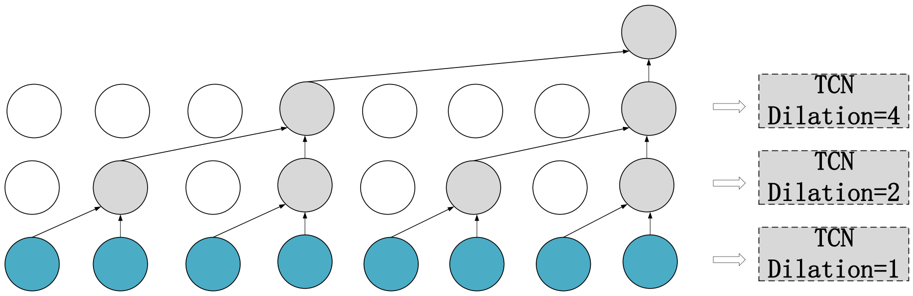

**Figure 2:** Dilated casual convolution

Figure 2 is a Dilated casual convolution with a core size of 2. Using dilation factor k, it selects the input every k step and applies the standard 1D convolution to the selected input.

Gated TCN: The gating mechanism is very important in the RNN. It can control the information of each layer in the TCN. A simple gating TCN contains only one output gate, as showed in Figure 3. Given input $X \in R^{N \times D \times S}$, its form is:

$$h = g(\theta_1 * X + b) \odot \sigma(\theta_2 * X + c) \quad (4)$$

Where $\theta_1$, $\theta_2$, $b$ and $c$ are model parameters, $\odot$ is element-wise product, $g(\cdot)$ is the output activation function, $\sigma(\cdot)$ is the sigmoid function, which determines the ratio of information passed to the next layer. In the model in this paper, Gated TCN is used to learn complex temporal dependence.

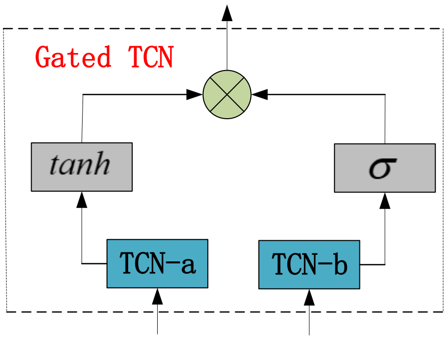

**Figure 3:** Gated TCN framework diagram

### D. Graph Convolution Network

Graph Convolutional Network (GCN) is the basic operation of extracting node features based on node structure information. GCN smooths the signal of a node by aggregating and transforming its neighborhood information. The filter of GCN is localized in space and supports multi-dimensional input. Let $A \in R^{N \times D}$ denote the normalized adjacency matrix, $X \in R^{N \times D}$ denote the input signal, $Z \in R^{N \times M}$ denote the output, and $W \in R^{D \times M}$ denote the model parameter matrix. The GCN is defined as:

$$Z = AXW \quad (5)$$

Literature [5] proposed the concept of diffusion convolutional network, which proved the effectiveness of diffusion convolutional network in spatio-temporal modeling. This paper uses K finite steps to model the diffusion process of the graph signal. In this paper, the diffusion convolutional network is extended to equation (5), and the result is expressed as:

$$Z = \sum_{k=0}^{K} P^k X W_k \quad (6)$$

Among them, $P^k$ represents the power series of the transition matrix. In undirected graphs, $P = A / rowsum(A)$. In the directed graph, the diffusion process is divided into two directions, forward and backward, in which forward transfer matrix $P_f = A / rowsum(A)$ and backward transfer matrix $P_b = A^T / rowsum(A)$. Therefore, the diffusion graph convolutional network can be defined as:

$$Z = \sum_{k=0}^{K} P_f^k X W_{k1} + P_b^k X W_{k2} \quad (7)$$

### 1. Self-adaptive Adjacency Matrix

In this paper, an adaptive adjacency matrix $A_{adp}$ is proposed. This kind of adaptive adjacency matrix does not require any prior knowledge and is learned end-to-end through stochastic gradient descent. By using learnable parameters $E_1 \in R^{N \times c}$ and $E_2 \in R^{N \times c}$, two node embedding dictionaries are randomly initialized to mine the hidden spatial features of traffic flow. The adaptive adjacency matrix proposed in this paper is shown in equation (8):

$$A_{adp} = softmax\left(ReLU\left(E_1 E_2^T\right)\right) \quad (8)$$

Among them, $E_1$ represents the source node embedding, and $E_2$ represents the target node embedding. Multiply $E_1$ and $E_2$ to obtain the spatial dependence weight between the source node and the target node. Use the $ReLU$ activation function to eliminate weak connections, and the $softmax$ function to normalize the adaptive adjacency matrix. Therefore, the normalized adaptive adjacency matrix can be regarded as a transition matrix for hiding the diffusion process. By combining predefined spatial dependencies and self-learning hidden spatial features, this paper proposes the following graph convolutional network:

$$Z = \sum_{k=0}^{K} P_f^k X W_{k1} + P_b^k X W_{k2} + A_{adp}^k X W_{k3} \quad (9)$$

When the graph network structure is not available, this paper proposes using the adaptive adjacency matrix alone to capture the hidden spatial dependencies, as shown in equation (10):

$$Z = \sum_{k=0}^{K} A_{adp}^k X W_k \quad (10)$$

### E. ST-Convolution Block

The traffic flow data detected by each sensor have a specific periodic time traffic pattern, such as peak and off-peak periods. The distance between different sensors has a potential impact on the spatial correlation of traffic flow, which is not affected by time dependence.

After ST-Attention Block, this paper designs a spatio-temporal convolution block containing three kernels to capture the many-to-one impact from three different angles corresponding to space, temporal, and spatio-temporal. Temporal kernel captures the time dependence of traffic flow at the same location, and spatial kernel captures the spatial correlation of traffic flow at adjacent locations at the same time step. Each spatio-temporal convolution block takes the output of the previous spatio-temporal attention block as input, namely $X_N^{(l)} \in R^{C(l)} \times \left|V_N\right| \times T_h$. The output $X_N^{(l+1)} \in R^{C(l+1)} \times \left|V_N\right| \times T_h$ is calculated by equation (11):

$$H = LeakyReLU\left[\Theta_{st}^{[l+1]} * X_N^{(l)}; \Theta_t^{[l+1]} * X_N^{(l)}; \Theta_s^{[l+1]} * X_N^{(l)};\right] \quad (11)$$

$$X_N^{(l+1)} = LeakyReLU\left(\Theta_o^{[l+1]} * H\right) \quad (12)$$

Among them, $\Theta_{st}^{[l+1]}$, $\Theta_t^{[l+1]}$, and $\Theta_s^{[l+1]}$ are the spatio-temporal kernels of $f \times f$, $f \times 1$ is the time kernel, and $1 \times f$ is the space kernel. $LeakyReLU(\cdot)$ represents $Leaky$ modified linear unit function, $*$ represents the convolution operation. For the above three convolution kernels, this paper takes $f = 3$ in the experiment. In addition, padding is used to ensure that the input and output have the same size. Finally, connect the outputs of the three convolution kernels, and use $1 \times 1$ convolution $\Theta_o^{[l+1]}$ to compress features and limit the number of channels.

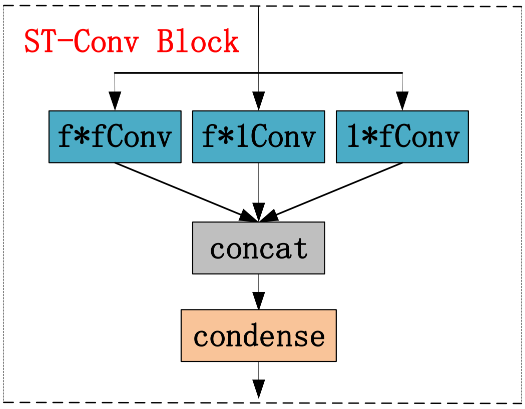

**Figure 4:** ST-Conv Block frame diagram

### F. ST-Attention Block

As showed in Figure 5, ST-Attention Block includes a spatial attention mechanism, a temporal attention mechanism and a gated fusion mechanism. In this paper, the input of the $l$ -th block is denoted as $H^{(l\text{-}1)}$ , and the hidden state of vertex $v_i$ at time step $t_j$ is denoted as $h^{(l-1)}_{v_i,t_j}$ . The output of the spatial attention mechanism and the temporal attention mechanism in the $l$ -th block is denoted as $H_S^{(l)}$ and $H_T^{(l)}$ , respectively, and the hidden state of the vertex $v_i$ at time step $t_j$ is denoted as $hs^{(l)}_{v_i,t_j}$ and $ht^{(l)}_{v_i,t_j}$ , respectively. After the gated fusion, the output of the $l$ -th block is obtained, denoted as $H^{(l)}$ .

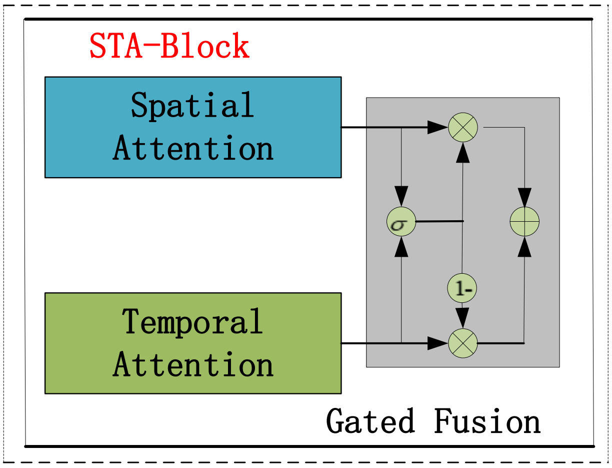

**Figure 5:** STA-Block framework diagram

### 1. Spatial Attention

The traffic condition of a road will be impacted differently by other roads. This spatial correlation is highly dynamic and changes over time. In order to model these attributes, this paper designs a spatial attention mechanism to adaptively capture the correlation between sensors in the road network. The key idea is to dynamically assign different weights to different road sections (such as sensors) at different time steps. For vertex $v_i$ at time step $t_j$ , calculate the weighted sum of all vertices:

$$hs^{(l)}_{v_i,t_j} = \sum_{v \in V} \alpha_{v_i,v} \cdot h^{(l-1)}_{v,t_j} \quad (13)$$

Where $V$ represents the set of all vertices, $\alpha_{v_i,v}$ is the attention score, which indicates the importance of vertex $v$ to $v_i$ , and the sum of attention scores is equal to 1: $\sum_{v \in V} \alpha_{v_i,v} = 1$ .

At a certain time step, the current traffic conditions and road network structure will affect the correlation between sensors. In this paper, the attention score is learned by considering traffic characteristics and graph network structure, and the scaling dot product method is used to calculate the correlation between vertices $v_i$ and $v$ :

$$s_{v_i,v} = \frac{\left\langle h^{(l-1)}_{v_i,t_j}, h^{(l-1)}_{v,t_j} \right\rangle}{\sqrt{2D}} \qquad (14)$$

Where $||$ represents the connection operation, $\langle \cdot,\cdot \rangle$ represents the inner product operator, and $2D$ is the dimension of $h^{(l-1)}_{v_i,t_j}$. Then, through the softmax function, $s_{v_i,v}$ is normalized to:

$$\alpha_{v_i,v} = \frac{exp\left(s_{v_i,v}\right)}{\sum_{v_r \in V} exp\left(s_{v_i,v_r}\right)} \qquad (15)$$

After the attention score $\alpha_{v_i,v}$ is obtained, the hidden state is updated by equation (13).

In order to stabilize the learning process, this paper extends the spatial attention mechanism to a multi-head attention mechanism. Specifically, connect $K$ parallel attention mechanisms with different learnable projections:

$$s^{(k)}_{v_i,v} = \frac{\left\langle f^{(k)}_{s,1}\left(h^{(l-1)}_{v_i,t_j}\right), f^{(k)}_{s,2}\left(h^{(l-1)}_{v,t_j}\right) \right\rangle}{\sqrt{d}} \qquad (16)$$

$$\alpha^{(k)}_{v_i,v} = \frac{exp\left(s^{(k)}_{v_i,v}\right)}{\sum_{v_r \in V} exp\left(s^{(k)}_{v_i,v_r}\right)} \qquad (17)$$

$$hs^{(l)}_{v_i,t_j} = \Big\|_{k=1}^{K} \left\{ \sum_{v \in V} \alpha^{(k)}_{v_i,v} \cdot f^{(k)}_{s,3}\left(h^{(l-1)}_{v,t_j}\right) \right\} \qquad (18)$$

Among them, $f^{(k)}_{s,1}(\cdot)$, $f^{(k)}_{s,2}(\cdot)$ and $f^{(k)}_{s,3}(\cdot)$ represent three different non-linear mappings in the $k$ -th head attention mechanism, which produce an $d = D / K$ dimensional output.

**2. Temporal Attention**

The traffic condition of a certain location is temporal-dependent with the previous observations, and this correlation varies non-linearly with the time step. In order to capture these characteristics, this paper designs a time attention mechanism to adaptively simulate the nonlinear correlation between different time steps. By considering the traffic characteristics and time to measure the correlation between different time steps, using the multi-head method to calculate the attention score, considering the vertex $v_i$, the correlation between the time step $t_j$ and $t$ is defined as:

$$u^{(k)}_{t_j,t} = \frac{\left\langle f^{(k)}_{t,1}\left(h^{(l-1)}_{v_i,t_j}\right), f^{(k)}_{t,2}\left(h^{(l-1)}_{v_i,t}\right) \right\rangle}{\sqrt{d}} \qquad (19)$$

$$\beta^{(k)}_{t_j,t} = \frac{exp\left(u^{(k)}_{t_j,t}\right)}{\sum_{t_r \in N_{t_j}} exp\left(u^{(k)}_{t_j,t_r}\right)} \qquad (20)$$

Among them, $u^{(k)}_{t_j,t}$ represents the correlation between time step $t_j$ and $t$, $\beta^{(k)}_{t_j,t}$ is the attention score of the $k$ -th head, which represents the importance of time step $t_j$ to $t$, $f^{(k)}_{t,1}(\cdot)$ and $f^{(k)}_{t,2}(\cdot)$ represent two different learnable conversions , $N_{t_j}$ represents a set of time steps before $t_j$. After obtaining the attention score, the hidden state of vertex $v_i$ at time step $t_j$ is updated as follows:

$$ht^{(l)}_{v_i,t_j} = \Big\|_{k=1}^{K} \left\{ \sum_{t\in N_{t_j}} \beta^{(k)}_{t_j,t} \cdot f^{(k)}_{t,3}\left(h^{(l-1)}_{v_i,t}\right) \right\} \quad (21)$$

Among them, $f^{(k)}_{t,3}(\cdot)$ represents a non-linear projection, and the learnable parameters in equations (19), (20) and (21) are shared among all vertices and time steps through parallel calculation.

### 3. Gated Fusion

The traffic condition of a road at a specific point in time and its previous value has temporal and spatial correlation with other road traffic conditions. As showed in Figure 4, this paper designs a gated fusion mechanism to adaptively merge the spatial attention mechanism and the temporal attention mechanism. In the $L$ -th STA-Block, the output of the spatial attention mechanism and the temporal attention mechanism is denoted as $H^{(L)}_{S}$ and $H^{(L)}_{T}$, respectively. $H^{(L)}_{S}$ and $H^{(L)}_{T}$ are fused by equation (22):

$$H^{(L)} = z \cdot H^{(L)}_{S} + (1 - z) \cdot H^{(L)}_{T} \quad (22)$$

$$z = sigmoid\left(H^{(L)}_{S} W_{z,1} + H^{(L)}_{T} W_{z,2} + b_z\right) \quad (23)$$

Among them, $W_{z,1} \in R^{D\times D}$, $W_{z,2} \in R^{D\times D}$, $b_z \in R^{D}$ are learnable parameters, and $z$ is the gate. The gating fusion mechanism adaptively controls the spatial correlation and time dependence of the traffic flow in each node and time step.

## IV. EXPERIMENT

### A. Data description

This paper uses two sets of traffic data sets, namely loop detector data set METR-LA in Los Angeles and the PEMS-BAY data set in California[5] to verify the performance of the TS-GCN model proposed in this paper. The experimental traffic data set records the detection location, detection date, data type, etc. The detailed information of the experimental data set is shown in Table 1:

Table 1. Description of experimental data set

| Data | METR-LA | PEMS-BAY |
|---|---|---|
| Type | sequentially | sequentially |
| Attribute | speed | speed |
| Location | highways of Los Angeles | the Bay Area |
| Edges | 1515 | 2369 |
| Time Steps | 34272 | 52116 |
| Nodes | 207 | 325 |

METR-LA recorded four months of traffic speed statistics through 207 sensors on Los Angeles County highways. PEMS-BAY contains 6-month traffic speed information from 325 sensors in the San Francisco Bay Area. The traffic speed is summarized every 5 minutes. In the experiment, all data sets are divided into a ratio of 7:2:1, which are used as training set, test set and verification set respectively. It is used to forecast traffic flow speed of 15 minutes, 30 minutes and 60 minutes.

Since the METR-LA dataset contains some missing data, this article uses linear interpolation to fill in the missing values. Before inputting the data into the forecasting model, this paper uses the min-max normalization method to normalize the data and limit it to [0, 1]. The normalization formula is

$$X^{norm}_{i} = \frac{x_i - x_{min}}{x_{max} - x_{min}} \quad (24)$$

$x_i$ represents the $i$ -th original data, $x_{max}$ and $x_{min}$ represent the maximum and minimum values of the original data, and $X^{norm}_{i}$ represents the normalized input data.

### B. Experimental environment and parameter settings

This experiment was prepared and run on a Linux server (CPU: Intel(R)Xeon(R)CPU E5-2620 v4 @ 2.10GHz, GPU: NVIDIA GeForce GTX 1080). Based on the Pytorch deep learning framework, the construction and training of the traffic flow forecasting model is completed in the PyCharm development environment.

This paper uses an 8-layer TS-GCN network, and its dilation factors sequence is 1, 2, 1, 2, 1, 2, 1. Use equation (7) as the graph convolutional network layer in this paper, and the diffusion step size K=2. Use Adam optimizer to train the model, the initial learning rate is 0.001, dropout p=0.3. In order to better analyze the experimental results and evaluate the forecasting performance of the model, this paper evaluates the error between the actual traffic flow speed and the predicted results based on the following evaluation indicators：

1) Mean Absolute Error (MAE):

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - y_i\right| \quad (25)$$

2) Root Mean Square Error (RMSE):

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(y_i - y_i\right)^2} \quad (26)$$

3) Mean absolute percentage error (MAPE):

$$MAPE = \frac{100}{n}\sum_{i=1}^{n}\left|\frac{y_i - y_i}{y_i}\right| \quad (27)$$

In the formula, $y_i$ and $\hat{y}_i$ respectively represent the actual traffic speed and the predicted traffic speed. $n$ is the number of observations. Use MAE, RMSE and MAPE to measure the forecasting error. The smaller the forecasting value, the better the forecasting effect.

## C. Baselines

This paper compares the TS-GCN model with the following models:

(1)HA[20]:Historical average model. Use average traffic information for historical periods as forecasting.

(2)VAR[21]:Vector Auto-Regression.

(3)SVR[14]:Support vector regression uses linear support vector machines to train the model to obtain the relationship between input and output to forecast traffic flow.

(4)FNN: Feedforward neural network with two hidden layers and L2 regularization.

(5)ARIMA[5]:Auto-Regressive Integrated Moving Average model with Kalman filter.

(6)FC-LSTM[5]:Recurrent neural network with fully connected LSTM hidden units.

(7)WaveNet[22]:A convolution network architecture for sequence data.

(8)Graph WaveNet[23]: Graph WaveNet that combines graph convolution with dilated casual convolution.

(9)STGCN[8]:Spatial-temporal graph convolution network, which combines graph convolution with 1D convolution.

(10)ASTGCN[16]: Attention-based spatio-temporal graph convolutional network, which further integrates spatial and temporal attention mechanisms to STGCN for capturing dynamic spatial and temporal patterns.

(11)STSGCN[24]: Spatial-Temporal Synchronous Graph Convolutional Network that captures spatial-temporal correlations by stacking multiple localized GCN layers with adjacent matrix over the time axis.

## D. Experimental Results

As showed in Table 2, the 15-minute, 30-minute, and 60-minute forecasting performance of the TS-GCN model and various baseline models on the METR-LA and PEMS-BAY data sets are compared. The TS-GCN model in this paper has obtained excellent forecasting results on both data sets.

It can be observed from Table 2: (1) Deep learning methods are superior to traditional time series methods and machine learning models, which proves the ability of deep neural networks to model nonlinear traffic flow data. (2) In the deep learning method, the performance of graph-based network models (including WaveNet, Graph WaveNet, STGCN, ASTGCN and STSGCN) is usually better than the FC-LSTM model. Graph WaveNet has obtained excellent results on both data sets, and its performance far exceeds time models such as HA, ARIMA and FC-LSTM. The TS-GCN model in this paper is superior to the previous convolution-based methods (STGCN, ASTGCN and STSGCN), indicating that the TS-GCN model can better capture the temporal and spatial dependence of traffic flow data. (3) The TS-GCN model achieves the best forecasting performance, whether it is in the range of 15 minutes, 30 minutes, or 60 minutes, it shows excellent forecasting performance. In contrast to the baseline model, the TS-GCN model uses stacked space-time layers, which contain GCN layers with different parameters. Therefore, each GCN layer in the TS-GCN model can focus on its own time input range.

It is shown from Table 2 that the traffic flow forecasting task will be significantly different depending on the traffic flow data set. The traffic flow forecasting on the METR-LA data set is more challenging than the traffic flow forecasting on the PEMS-BAY data set. As a result, the forecasting performance of all models is not so good as that on the PEMS-BAY data set. Therefore, the forecasting task on the PEMS-BAY data set is simpler, and the forecasting performance of the model is significantly better.

Table 2. Performance comparison of different traffic flow forecasting models on the METR-LA and PEMS-BAY datasets

| Data | Models | | 15min | | | 30min | | | 60min | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | MAE | RMSE | MAPE | MAE | RMSE | MAPE | MAE | RMSE | MAPE |
| METR-LA | HA | 4.16 | 7.80 | 13.00% | 4.16 | 7.80 | 13.00% | 4.16 | 7.80 | 13.00% |
| | VAR | 4.42 | 7.89 | 10.20% | 5.41 | 9.13 | 12.7% | 6.52 | 10.11 | 15.80% |
| | SVR | 3.99 | 8.45 | 9.30% | 5.05 | 10.87 | 12.10% | 6.72 | 13.76 | 16.70% |
| | FNN | 3.99 | 7.94 | 9.90% | 4.23 | 8.17 | 12.90% | 4.49 | 8.69 | 14.00% |
| | ARIMA | 3.99 | 8.21 | 9.60% | 5.15 | 10.45 | 12.70% | 6.90 | 13.23 | 17.40% |
| | FC-LSTM | 3.44 | 6.30 | 9.60% | 3.77 | 7.23 | 10.90% | 4.37 | 8.69 | 13.20% |
| | WaveNet | 2.99 | 5.89 | 8.04% | 3.59 | 7.28 | 10.25% | 4.45 | 8.93 | 13.62% |
| | Graph WaveNet | 2.98 | 5.90 | 7.92% | 3.59 | 7.29 | 10.26% | 4.43 | 8.97 | 13.64% |
| | STGCN | 2.88 | 5.74 | **7.62%** | 3.47 | 7.24 | 9.57% | 4.59 | 9.40 | 12.70% |
| | ASTGCN | 4.86 | 9.27 | 9.21% | 5.43 | 10.61 | 10.13% | 6.51 | 12.52 | **11.64%** |
| | STSGCN | 3.31 | 7.62 | 8.06% | 4.13 | 9.77 | 10.29% | 5.06 | 11.66 | 12.91% |
| | **TS-GCN** | **2.71** | **5.61** | 8.17% | **3.33** | **7.08** | **9.47%** | **4.17** | **8.54** | 12.34% |
| PEMS-BAY | HA | 2.88 | 5.59 | 6.80% | 2.88 | 5.59 | 6.80% | 2.88 | 5.59 | 6.80% |
| | VAR | 1.74 | 3.16 | 3.60% | 2.32 | 4.25 | 5.00% | 2.93 | 5.44 | 6.50% |
| | SVR | 1.85 | 3.59 | 3.80% | 2.48 | 5.18 | 5.50% | 3.28 | 7.08 | 8.00% |
| | FNN | 2.20 | 4.42 | 5.19% | 2.30 | 4.63 | 5.43% | 2.46 | 4.98 | 5.89% |
| | ARIMA | 1.62 | 3.30 | 3.50% | 2.33 | 4.76 | 5.40% | 3.38 | 6.50 | 8.30% |
| | FC-LSTM | 2.05 | 4.19 | 4.80% | 2.20 | 4.55 | 5.20% | 2.37 | 4.96 | 5.70% |
| | WaveNet | 1.39 | 3.01 | 2.91% | 1.83 | 4.21 | 4.16% | 2.35 | 5.43 | 5.87% |
| | Graph WaveNet | 1.39 | 3.01 | 2.89% | 1.83 | 4.21 | 4.11% | 2.35 | 5.43 | 5.78% |
| | STGCN | 1.36 | 2.96 | 2.90% | 1.81 | 4.27 | 4.17% | 2.49 | 5.69 | 5.79% |
| | ASTGCN | 1.52 | 3.13 | 3.22% | 2.01 | 4.27 | 4.48% | 2.61 | 5.42 | 6.00% |
| | STSGCN | 1.44 | 3.01 | 3.04% | 1.83 | 4.18 | 4.17% | 2.26 | 5.21 | **5.40%** |
| | **TS-GCN** | **1.30** | **2.86** | **2.84%** | **1.73** | **4.03** | **4.11%** | **2.21** | **5.13** | 5.58% |

(1) Ablation experiment

In order to further study the performance of different modules of the TS-GCN model, this paper designs two variants of the TS-GCN model, and studies the attention mechanism, the gated fusion mechanism and the influence of the spatio-temporal convolutional layer on the model's forecasting performance. The two variants are compared with the TS-GCN model on the METR-LA and PEMS-BAY data sets, and 15-minute, 30-minute, and 60-minute traffic flow forecasting are performed, as showed in Table 3. The differences between these two variant models and the STAGCN model are:

Without STA-Block and ST-Conv Block: This model has no spatio-temporal attention mechanism module, no gated fusion mechanism, and no spatio-temporal convolutional network module.

Without ST-Conv Block: This model does not have a spatio-temporal convolutional network module.

Table 3. The forecasting performance of the TS-GCN model and the two variant models at different time points

| Data | Models | | 15min | | | 30min | | | 60min | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | MAE | RMSE | MAPE | MAE | RMSE | MAPE | MAE | RMSE | MAPE |
| METR-LA | Without STA-Block and ST-Conv Block | 2.98 | 5.89 | 7.98% | 3.58 | 7.29 | 10.24% | 4.43 | 8.98 | 13.58% |
| | Without ST-Conv Block | 2.78 | 5.65 | **7.66%** | 3.36 | **7.00** | 9.94% | 4.20 | 8.66 | **11.71%** |
| | **TS-GCN** | **2.71** | **5.61** | 8.17% | **3.33** | 7.08 | **9.47%** | **4.17** | **8.54** | 12.34% |
| PEMS-BAY | Without STA-Block and ST-Conv Block | 1.43 | 3.04 | 3.07% | 1.88 | 4.29 | 4.42% | 2.37 | 5.46 | 5.96% |
| | Without ST-Conv Block | 1.40 | 2.98 | 2.92% | 1.85 | 4.25 | 4.26% | 2.34 | 5.37 | 5.85% |
| | **TS-GCN** | **1.30** | **2.86** | **2.84%** | **1.73** | **4.03** | **4.11%** | **2.21** | **5.13** | **5.58%** |

In 15 minutes, compared with the Without STA-Block and ST-Conv Block and Without ST-Conv Block models, the MAE of the TS-GCN model on the PEMS-BAY data set is reduced by about 9.09% and 7.14%, respectively, and RMSE was reduced by approximately 5.92% and 4.03% respectively. In 30 minutes, the MAE was reduced by about 7.98% and 6.49%, and the RMSE was decreased by about 6.06% and 5.18%, respectively. In 60 minutes, the MAE was reduced by about 6.75% and 5.56%, respectively, and the RMSE was reduced by about 6.04% and 4.47%, respectively. In the same way, the TS-GCN model also achieved better forecasting performance on the METR-LA data set. In addition, depending to Table 3, the TS-GCN model has better forecasting performance at different time points, which proves that the STA-Block and ST-Conv Block modules effectively alleviate the impact of error propagation.

In order to better explain the TS-GCN model, visualize the experimental results of the TS-GCN model and FNN, FC-LSTM, Graph WaveNet and STGCN on the PEMS-BAY data set, as showed in Figure 6, Figure 7 and Figure 8. It can be observed from the three figures that the TS-GCN model has always been better than FNN, FC-LSTM, Graph WaveNet and STGCN, indicating that the TS-GCN model proposed in this paper is successful in capturing the temporal and spatial correlation of traffic flow data.

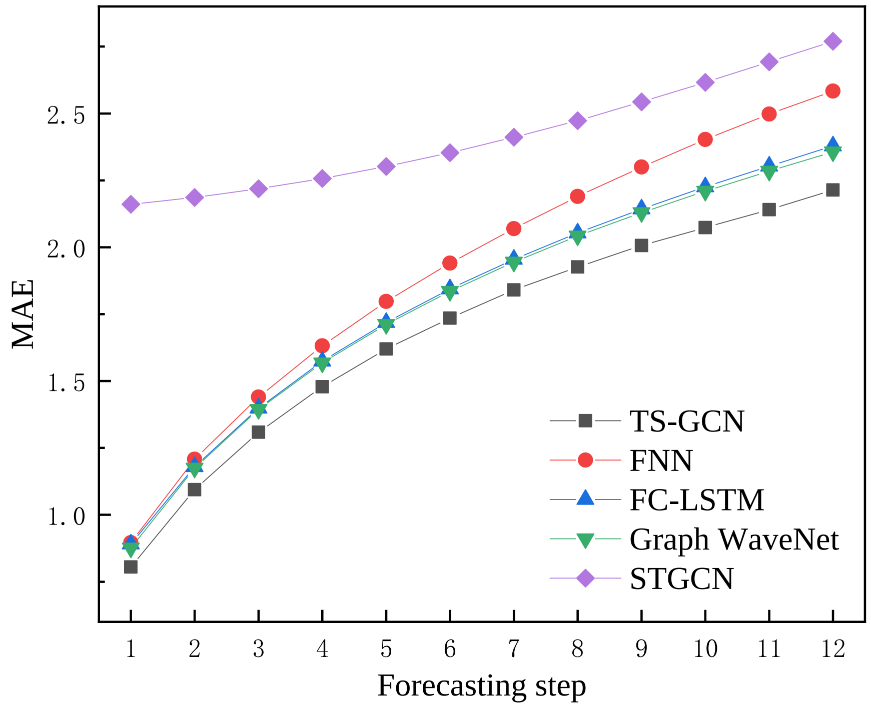

**Figure 6:** Visualization of MAE experiment results

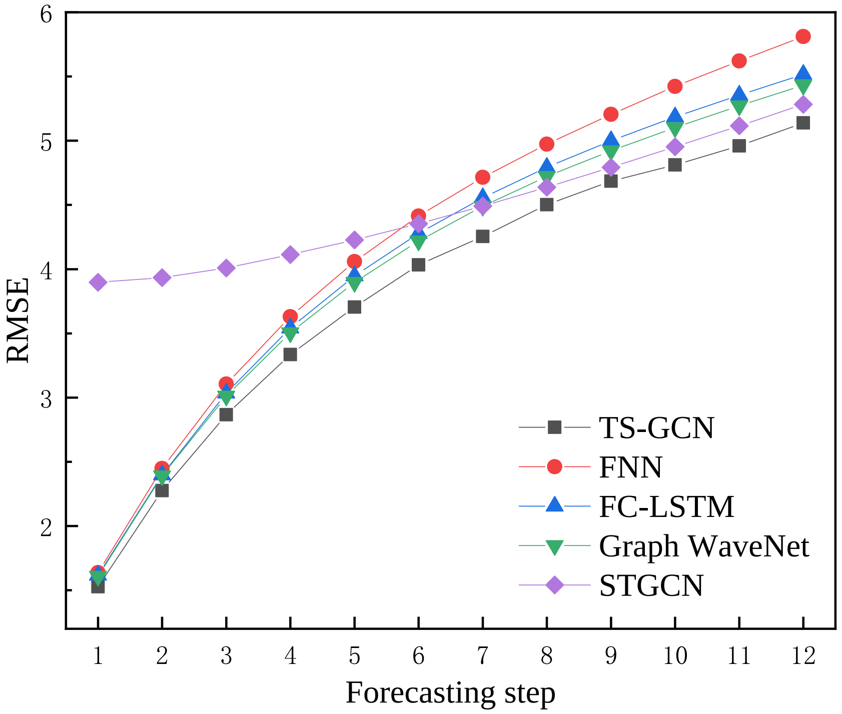

**Figure 7:** Visualization of RMSE experiment results

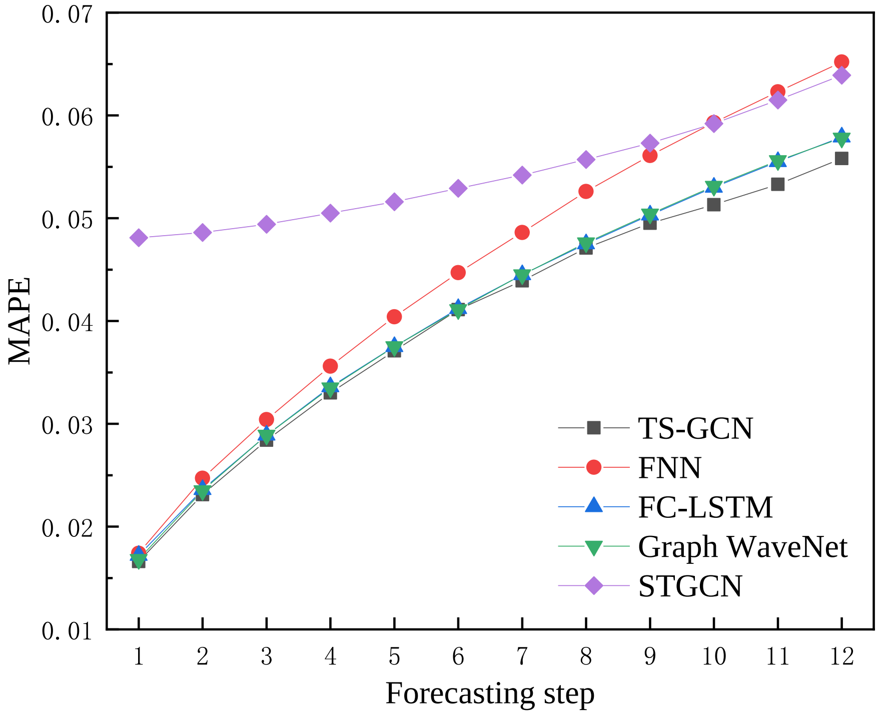

**Figure 8:** Visualization of MAPE experiment results

In short, under different forecasting time steps, the TS-GCN model can always achieve the best results, which show that the TS-GCN model can have a good forecasting performance in traffic forecasting tasks. The model can also capture the changing trend of traffic speed and identify the start time and end time of the peak period of traffic flow. The TS-GCN model can accurately forecast traffic congestion. Thus, proving the effectiveness of the TS-GCN model in real-time traffic forecasting.

## V. CONCLUSION

Fully capturing the temporal and spatial characteristics of traffic flow and considering this complex characteristic in the modeling process can effectively improve the forecasting accuracy of traffic flow. In this paper, a novel spatio-temporal graph modeling model, which it is called TS-GCN, is proposed. It combines temporal convolutional network (TCN) and graph convolutional network (GCN), effectively capture the temporal and spatial dependence of traffic flow data. This paper proposes an effective method to automatically learn the hidden spatial features from the data. That is, to learn the hidden spatial correlation by constructing an adaptive adjacency matrix. In this paper, a spatial attention mechanism and a temporal attention mechanism are introduced to learn the dynamic spatial correlation and nonlinear time dependence of traffic flow, respectively. In addition, this paper designs a gated fusion mechanism to fuse the information extracted through the temporal and spatial attention mechanism, fully excavate the complex spatio-temporal relationships of traffic flow, thereby improving the model's ability to characterize temporal and spatial characteristics of traffic flow, so as to improve the accuracy of traffic flow forecasting. A large number of experiments were performed on two traffic data sets to verify the forecasting performance of the TS-GCN model and compare it with some baseline models. The experimental results show that the forecasting accuracy of the TS-GCN model in different data sets and different forecasting time periods is the best among the baseline models, which proves the effectiveness and accuracy of the model in traffic flow forecasting.

In future work, we will further study the application of the model proposed in this paper on large-scale traffic flow data sets and the scalable method to continue to improve its forecasting accuracy and the time efficiency.

## ACKNOWLEDGMENT

This work was supported by Natural Science Foundation of Gansu Province, China (Grant No.20JR5RA450); National Key Research and Development Plan (Grant No.2019YFB1707303).

## REFERENCES

[1]. J. Zhang, F.-Y. Wang, K. Wang et al., "Data-Driven Intelligent Transportation Systems: A Survey," IEEE Transactions on Intelligent Transportation Systems, vol. 12, no. 4, pp. 1624-1639, 2011.

[2]. Y. Wang, N. Geroliminis, and L. Leclercq, "Recent advances in ITS, traffic flow theory, and network operations," Transportation Research Part C: Emerging Technologies, vol. 68, pp. 507-508, 2016.

[3]. Y. Lv, Y. Duan, W. Kang et al., "Traffic Flow Prediction With Big Data: A Deep Learning Approach," IEEE Transactions on Intelligent Transportation Systems, pp. 1-9, 2014.

[4]. Y. Jia, J. Wu, and Y. Du, ''Traffic speed prediction using deep learning method,'' in Proc. IEEE 19th Int. Conf. Intell. Transp. Syst. (ITSC), Nov. 2016, pp. 1217–1222.

[5]. Li Y, Yu R, Shahabi C, et al. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. In ICLR, 2018.

[6]. Yao H , Fei W , Ke J , et al. Deep multi-view spatial-temporal network for taxi demand prediction. In AAAI, pages 2588–2595, 2018.

[7]. Yan S, Xiong Y, Lin D. Spatial temporal graph convolutional networks for skeleton-based action recognition. In AAAI, pages 3482–3489, 2018.

[8]. Yu B, Yin H, Zhu Z. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. In IJCAI, pages 3634–3640, 2018.

[9]. Zhang J, Shi X, Xie J, et al. Gaan: Gated attention networks for learning on large and spatiotemporal graphs. arXiv preprint arXiv:1803.07294, 2018.

[10]. A. Krizhevsky, I. Sutskever, and G. E. Hinton, ''ImageNet classification with deep convolutional neural networks,'' in Proc. Adv. Neural Inf. Process. Syst., 2012, pp. 1097–1105.

[11]. D. Y u, Y . Liu, and X. Y u, ''A data grouping CNN algorithm for short-term traffic flow forecasting,'' in Proc. Asia–Pacific Web Conf. Springer, 2016,pp. 92–103.

[12]. G. Dai, C. Ma, and X. Xu, "Short-Term Traffic Flow Prediction Method for Urban Road Sections Based on Space–Time Analysis and GRU," IEEE Access, vol. 7, pp. 143025-143035, 2019.

[13]. Z. Zhao, W. Chen, X. Wu et al., "LSTM network: a deep learning approach for short- term traffic forecast," IET Intelligent Transport Systems, vol. 11, no. 2, pp. 68-75, 2017.

[14]. Y. Wu and H. Tan, "Short-term traffic flow forecasting with spatial-temporal correlation in a hybrid deep learning framework," 2016, arXiv:1612.01022.

[15]. S. Xingjian, 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 Proc. Adv. Neural Inf. Process. Syst., 2015,pp. 802–810.

[16]. S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan, "Attention based spatial-temporal graph convolutional networks for traffic flow forecasting," in Proc. AAAI Conf. Artif. Intell., vol. 33, 2019, pp. 922–929.

[17]. X. Feng, J. Guo, B. Qin, T. Liu, and Y. Liu, "Effective deep memory networks for distant supervised relation extraction," in Proc. IJCAI, Aug. 2017, pp. 4002–4008.

[18]. D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," 2014, arXiv:1409.0473.

[19]. P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, ''Graph attention networks,'' 2017, arXiv:1710.10903.

[20]. Q. Liu, B. Wang, and Y . Zhu, ''Short-term traffic speed forecasting based on attention convolutional neural network for arterials,'' Comput.-Aided Civil Infrastruct. Eng., vol. 33, no. 11, pp. 999–1016, Nov. 2018.

[21]. Hansen B E. TIME SERIES ANALYSIS James D. Hamilton Princeton University Press, 1994[J]. Econometric Theory, 1995, 11(5):1148-1171.

[22]. Oord A, Dieleman S, Zen H, et al. WaveNet: A generative model for raw audio. arXiv preprint arXiv:1609.03499, 2016.

[23]. Wu, Z.; Pan, S.; Long, G.; Jiang, J.; and Zhang, C. 2019b. Graph WaveNet for deep spatial-temporal graph modeling. In IJCAI.

[24]. Song C, Lin Y, Guo S, et al. Spatial-Temporal Synchronous Graph Convolutional Networks: A New Framework for Spatial-Temporal Network Data Forecasting[J]. Proceedings of the AAAI Conference on Artificial Intelligence, 2020, 34(1):914-921.