# Dynamic Gated Spatial Temporal Graph Neural Networks for Traffic Forecasting

Ziyan Gui[1], Changhui Liu[1*], Li Xiong[1], Zuoquan Xie[2] and Liang Wu[3]

[1]*Wuhan Institute of Technology, Wuhan, China*
[2]*Central China Normal University, Wuhan, China*
[3]*Southwestern University of Finance and Economics, Chengdu, China*

**Abstract**

Traffic forecasting is crucial to intelligent transportation system, and very challenging due to the uncertainty and complexity of spatial-temporal dependencies in real-world traffic network. Many existing approaches use the pre-defined graph to model spatial correlations, but they fail to capture the latent spatial evolution. Then some dynamic graph-based methods are proposed to address this issue, however they separately model spatial and temporal dependencies without internal connection. In this paper, we propose a novel Dynamic gated Spatial Temporal Graph Neural Network (DSTGNN) for traffic forecasting, which can capture time-varying spatial correlations and temporal dependencies jointly. Besides, we apply gate mechanism into residual connection between extracted spatial and temporal features. Experimental results on two real-world traffic datasets have demonstrated the effectiveness of DSTGNN, and that the proposed DSTGNN can compete with state-of-the-art baselines.

**Keywords**

Traffic forecasting, DSTGNN, Spatial-Temporal correlation

## 1. Introduction

As a core component of Intelligent Transportation System (ITS)[4], real-time and accurate traffic forecasting is curial for road resource planning and public traffic safety. The key of traffic forecasting is to capture dynamic and uncertain spatial-temporal dependencies from historical data.

In early deep learning approaches, convolutional neural networks (CNNs) are used to extract spatial correlations and RNNs are used to model temporal dependencies. However, CNNs are only suitable for capturing spatial features in grid-data and perform poorly in non-Euclidean space. Recently, graph neural networks (GNNs)[1,12,13] have been generalized as convolution on graph-based data, which can extract the intrinsic spatial topological information of graphs.

To model both temporal and spatial dependencies, many early approaches [3]combined GNNs with RNN-based sequence models and achieved improvement. However, most of them model spatial correlations based on predefined static graph and cannot capture spatial dynamics. With the advent of self-attention in Transformer[13]spatial and temporal attention are adopted in these methods[6,9,11]to model dynamic spatial-temporal correlations and improved a lot. However, spatial-temporal correlations are not captured interactively, resulting in learning irrelevant or redundant information.

In this paper, we proposed an end-to-end model named Dynamic gated Spatial Temporal Graph Neural Network (DSTGNN), which models spatial-temporal dependencies jointly to address the above issues. Specifically, we stack multiple the proposed dynamic gated spatial-temporal(DGST) blocks to extract spatial-temporal features from historical traffic data. Each DGST block consists of a dynamic graph convolution module for extracting dynamic and static spatial correlations, a temporal attention

ICBASE2022@3rd International Conference on Big Data & Artificial Intelligence & Software Engineering, October 21-23, 2022, Guangzhou, China

EMAIL: 2927990583@qq.com (Ziyan Gui); lch52012@163.com (Changhui Liu)*; 2273167975@qq.com (Li Xiong); 1584103174@qq.com (Zuoquan Xie)
ORCID: 0000-0002-4530-5604(ZiyanGui);

module for modeling time dynamics and a gated residual connection for interaction between extracted spatial and temporal features. The contribution of this paper can be summarized as:

- We propose the dynamic gated spatial-temporal block that jointly models the spatial and temporal dependencies of traffic data, and then make prediction in a non-autoregressive way.
- Experiments on two real-world traffic datasets are conducted to evaluate the effectiveness of our model, and results show that our model can compete with the state-of-the-arts.

# 2. Related works

## 2.1. Graph Convolutional Network

As an efficient variant of CNNs on graph-structured data, graph convolutional networks (GCNs) have been applied into various areas and achieved state-of-the-art results. In the spectral perspective [1], GCNs need to compute the eigen-decomposition of the Laplacian matrix, which leads to a huge consumption of computation resources. Subsequently, methods[12,14]based on Chebyshev polynomial approximation are proposed to improve the computing efficiency. In addition, GCNs based on the spatial perspective[15,16]not only avoid the eigen-decomposition of Laplacian matrix but also can learn the vertices representations inductively.[5]proposes STGCN combined GCN with standard temporal 1D CNN to tackle the traffic time series prediction. [10]proposes Graph WaveNet which learns static adjacency matrices for spatial-temporal modeling but failed to capture dynamic spatial correlations.

## 2.2. Attention based Traffic forecasting

Attention mechanism has been extensively utilized in many domains such as natural language processing and graph learning (GAT[16]). Recently, researchers apply attention mechanism to model spatial-temporal dependencies for traffic forecasting. GMAN[6] proposes the ST-Attention block, which adds a transform attention layer between the encoder and decoder and models dynamic spatial-temporal correlations in both the encoder and decoder. For learning spatial-temporal features, ASTGCN[9]employs attention mechanisms in the spatial and temporal dimensions, respectively. A general dynamic graph neural network is created by STTN[11]to model the spatial dependencies that change over time. LSGCN[2]combines GCN with graph attention for long short-term traffic prediction.

# 3. Methodology

## 3.1. Problem Definition

In this study, a traffic network is denoted as a directed weighted graph $G=(V,E,\mathrm{A})$, where $V$ is a set of $N=|V|$ nodes representing sensors in traffic network; $E$ is a set of edges indicating the connectivity among the nodes; and $\mathrm{A}\in\mathbb{R}^{N\times N}$ is the weighted adjacency matrix of graph $G$ representing the proximity measured by the Euclidean distances between sensors via Gaussian kernel.

At each time step $t$ , the traffic conditions can be represented as a graph signal $X_t\in\mathbb{R}^{N\times C}$ on graph $G$ , where $C$ is the number of traffic conditions observed such as traffic speed, traffic density and so on.

Given a traffic network graph $G$ and traffic conditions of historical $P$ time steps $\left[X_{t-P+1},\cdots,X_t\right]\in\mathbb{R}^{P\times N\times C}$ observed by the $N$ nodes, we aim to learn a function $f$ to forecast the traffic conditions of the next $F$ time steps over all nodes. The process can be formulated as:

$$\hat{X}_{t+1},\cdots,\hat{X}_{t+F}=f\left(X_{t-P+1},\cdots,X_t;G\right), \tag{1}$$

where $\left[\hat{X}_{t+1},\cdots,\hat{X}_{t+F}\right]\in\mathbb{R}^{F\times N\times C}$.

## 3.2. Overall Architecture

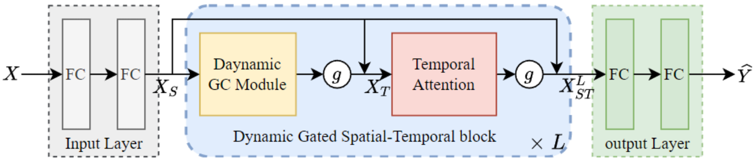


**Figure 1**: The overview of DSTGNN

The overall architecture of the proposed DSTGNN is shown in Figure 1, which consists of three main components including input layer, multiple stacked dynamic gated spatial-temporal (DGST) blocks and output layer. We first use two-layer fully-connected network before traffic data enters the model to project the data to high-dimension space, then the multiple stacked DGST blocks extract spatial and temporal dependencies jointly from input. Finally, the output layer transforms the features with spatial-temporal information from high-dimension space to traffic speed. Besides, DSTGNN predicts future traffic conditions in a multi-step manner. The detailed modules will be introduced later.

## 3.3. Dynamic Graph Convolution Module

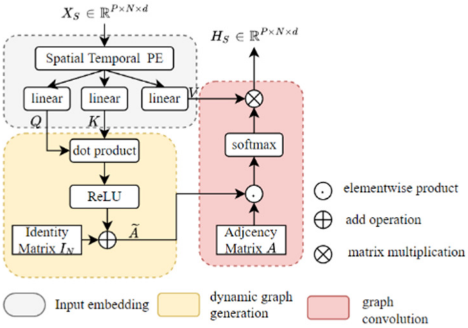


**Figure 2**: Illustration of the proposed dynamic graph convolution (DGC) module

Traffic conditions at a location are influenced not only by predefined static graph relations, but also by dynamic spatial correlations at that time step, with varying degrees of influence at each time step. According to CDGNet[12]is vital to reflect the sparsity of spatial correlations in real traffic network, instead of calculating a dense adjacency matrix at each time step via the dot-product operation. In the proposed dynamic graph convolution (DGC) module, the sparse dynamic adjacency matrix is combined with predefined graph to extract dynamic and static spatial correlations jointly at each time slice.

As shown in Figure 2, the output tensor $X_S \in \mathbb{R}^{P\times N\times d}$ of input layer is embedded by spatial temporal positional embedding module to get $X_S' \in \mathbb{R}^{P\times N\times d}$ according to STTN, before it entering the first DGC module (for DGC module in the first DGST block, $X_{ST}^{(0)} = X_S'$ ). We first view the temporal dimension of $X_S'$ as a batch dimension, and then $X_S'$ is transformed to the query, key and value subspaces by three different linear projection, obtaining $Q_S, K_S, V_S \in \mathbb{R}^{P\times N\times d}$ ， formally:

$$Q_S = X_{ST}^{(l-1)} W_q, K_S = X_{ST}^{(l-1)} W_k, V_S = X_{ST}^{(l-1)} W_v, \tag{2}$$

where $W_q, W_k, W_v \in \mathbb{R}^{d\times d}$ are all learnable weight matrix, $X_{ST}^{(l-1)} \in \mathbb{R}^{P\times N\times d}$ is input of DGC module in $l^{th}$ DGST block, and $d$ is the dimension of feature space.

Next, we use $ReLU(\cdot)$ to sparse the dense adjacency matrix derived from dot product as follows:

$$\tilde{A} = \mathrm{Re}LU\left(Q_S K_S^T\right) + I_N, \tag{3}$$

where $\tilde{A} \in \mathbb{R}^{N \times N}$ is the sparse spatial correlation matrix, $I_N$ is an identity matrix, and add it to enhance the self-connection ability. Furthermore, to capture both dynamic and static spatial correlations, we combine $\tilde{A}$ and the predefined adjacency matrix $A$ by element-wise product with broadcasting mechanism. A $softmax(\cdot)$ function is then adopted to normalize the combined adjacency matrix to avoid gradient disappearing or explosion, the final graph convolution operation can be expressed formally as:

$$H_S^{(l)} = softmax(\tilde{A} \odot A)V_S \ , \tag{4}$$

where $H_S^{(l)} \in \mathbb{R}^{P \times N \times d}$ denotes the output of the DGC module in the $l^{th}$ DGST block, and $\odot$ stands for element-wise product.

## 3.4. Gated Residual Connection

As shown in Figure 1, we apply a gate function to regulate the flow of residual information, drawing inspiration from the gate mechanism in the Gated Recurrent Unit (GRU). How much the previous residual information can influence the following module is regulated by this gate mechanism. Specifically, a gated residual shortcut path is added to DGC module, fusing the spatial correlation of the output $H_S^{(l)}$ and the input's spatial-temporal features $X_{ST}^{(l-1)} \in \mathbb{R}^{P \times N \times d}$ in the $l^{th}$ DGST block with $X_{ST}^{(0)} = X_S^{'}$, which can be formulated as:

$$\begin{gathered} X_T^{(l)} = g_s \odot H_S^{(l)} + X_{ST}^{(l-1)} W_{res}, \\ g_s = \sigma\left(X_{ST}^{(l-1)} U_g\right), \end{gathered} \tag{5}$$

where $X_T^{(l)} \in \mathbb{R}^{P \times N \times d}$ is the following temporal attention module's input, $g_s \in \mathbb{R}^{P \times N \times d}$ denotes the gate, $W_{res} \in \mathbb{R}^{d \times d}$ denotes the linear projection weight, and $U_g \in \mathbb{R}^{d \times d}$ denotes the state-to-state weight matrix, $\odot$ is the element-wise product, and $\sigma$ stands for the $sigmoid(\cdot)$ non-linear activation function.

Similar to DGC module, we also added a gated residual shortcut path to the following temporal attention module, which combines the spatial correlation of input $X_T^{(l)}$ with the temporal dependence of output $H_T^{(l)}$, building an interaction between the extracted spatial and temporal information that can be learned to extract spatial-temporal representations $X_{ST}^{(l)} \in \mathbb{R}^{P \times N \times d}$ . This process can be written as：

$$\begin{gathered} X_{ST}^{(l)} = g_t \odot H_T^{(l)} + X_T^{(l)} W_{res}, \\ g_t = \sigma\left(X_T^{(l)} U_g\right), \end{gathered} \tag{6}$$

where $H_T^{(l)} \in \mathbb{R}^{P \times N \times d}$ is output of the following temporal attention module, and $g_t$ is the gate function.

## 3.5. Temporal Attention Module

The temporal attention module takes the output $X_T^{(l)} \in \mathbb{R}^{P \times N \times d}$ of gated residual connection in DGC module as input and views the spatial dimension of $X_T^{(l)}$ as a batch dimension. Then it uses the transformer-based encoder to model temporal dependence from $X_T^{(l)}$ with spatial correlation. It mainly consists of attention aggregation and FFN refining, with the former highlighting relative temporal cues and the latter updating refined features. The output tensor $H_T^{(l)} \in \mathbb{R}^{P \times N \times d}$ of the temporal attention module can be computed as follows:

$$\begin{gathered} Q_T = X_T^{(l)} W_q^{'}, K_T = X_T^{(l)} W_k^{'}, V_T = X_T^{(l)} W_v^{'}, \\ V_T^{'} = \mathrm{LN}\left( softmax\left( \frac{Q_T K_T^T}{\sqrt{d}} \right) V_T + X_T \right), \\ H_T^{(l)} = \mathrm{LN}\left( \mathrm{FFN}\left(V_T^{'}\right) + V_T^{'} \right), \end{gathered} \tag{7}$$

where $W_q^{'}, \; W_k^{'}, \; W_v^{'} \in \mathbb{R}^{d \times d}$ are all weight matrix, LN indicates layer normalization, and FFN is the feed forward network. The above calculation can also be extended in a multi-head manner.

## 3.6. Loss Function

The output layer regards the last DGST block's output $X_{ST}^{(l)} \in \mathbb{R}^{P \times N \times d}$ as input, then transforms it to final prediction results $\hat{Y} \in \mathbb{R}^{F \times N \times C}$ . The proposed DSTGNN can be trained in an end-to-end style via back-propagation by minimizing the mean absolute error between predicted values and ground truths:

$$\mathcal{L}(\Theta) = \frac{1}{F \times N} \sum_{t=1}^{F} \sum_{n=1}^{N} \left| \hat{Y}_{t,n} - Y_{t,n} \right| \tag{8}$$

where $\Theta$ denotes all the learnable parameters in our model, and $Y$ denotes the ground truth.

# 4. Experiments

## 4.1. Experimental Settings

### 4.1.1. Datasets

As detailed below, we evaluate our DSTGNN on two public real-world traffic datasets, namely PeMSD7 and PEMS-BAY. PeMSD7 collects 2-month traffic data on 228 sensors during the weekdays of May and June 2012 in California's District 7. PEMS-BAY contains six months of traffic data from 325 sensors in the Bay area from January 1st to May 31st, 2017. Following STGCN[5], PeMSD7 uses the first 34 days as a training set and the remaining days as a validation and test set. As with DCRNN[3], PEMS-BAY is divided into three sets: training (70%), validation (10%), and test (20%).

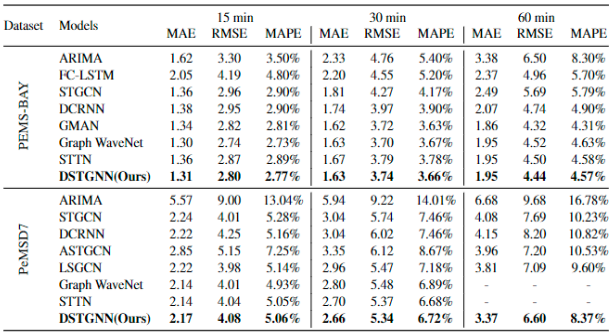
### 4.1.2. Evaluation Metric

**Metrices.** To evaluate the performance of our model, we employ three metrics: Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Mean Absolute Percentage Error (MAPE).

**Baselines.** Our DSTGNN is compared to the following baselines: ARIMA[7], FC-LSTM[8], DCRNN[3], STGCN[5], GMAN[6], ASTGCN[9], STTN[11], Graph WaveNet[10], LSGCN[2].

## 4.2. Experimental Results and Analysis

**Table 1**
Traffic forecasting performance comparison of DSTGNN and other baselines.



| Dataset | Models | 15 min |  |  | 30 min |  |  | 60 min |  |  |
|---|---|---|---|---|---|---|---|---|---|---|
|  |  | MAE | RMSE | MAPE | MAE | RMSE | MAPE | MAE | RMSE | MAPE |
| PEMS-BAY | 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% |
|  | STGCN | 1.36 | 2.96 | 2.90% | 1.81 | 4.27 | 4.17% | 2.49 | 5.69 | 5.79% |
|  | DCRNN | 1.38 | 2.95 | 2.90% | 1.74 | 3.97 | 3.90% | 2.07 | 4.74 | 4.90% |
|  | GMAN | 1.34 | 2.82 | 2.81% | 1.62 | 3.72 | 3.63% | 1.86 | 4.32 | 4.31% |
|  | Graph WaveNet | 1.30 | 2.74 | 2.73% | 1.63 | 3.70 | 3.67% | 1.95 | 4.52 | 4.63% |
|  | STTN | 1.36 | 2.87 | 2.89% | 1.67 | 3.79 | 3.78% | 1.95 | 4.50 | 4.58% |
|  | **DSTGNN(Ours)** | **1.31** | **2.80** | **2.77%** | **1.63** | **3.74** | **3.66%** | **1.95** | **4.44** | **4.57%** |
| PeMSD7 | ARIMA | 5.57 | 9.00 | 13.04% | 5.94 | 9.22 | 14.01% | 6.68 | 9.68 | 16.78% |
|  | STGCN | 2.24 | 4.01 | 5.28% | 3.04 | 5.74 | 7.46% | 4.08 | 7.69 | 10.23% |
|  | DCRNN | 2.22 | 4.25 | 5.16% | 3.04 | 6.02 | 7.46% | 4.15 | 8.20 | 10.82% |
|  | ASTGCN | 2.85 | 5.15 | 7.25% | 3.35 | 6.12 | 8.67% | 3.96 | 7.20 | 10.53% |
|  | LSGCN | 2.22 | 3.98 | 5.14% | 2.96 | 5.47 | 7.18% | 3.81 | 7.09 | 9.60% |
|  | Graph WaveNet | 2.14 | 4.01 | 4.93% | 2.80 | 5.48 | 6.89% | - | - | - |
|  | STTN | 2.14 | 4.04 | 5.05% | 2.70 | 5.37 | 6.68% | - | - | - |
|  | **DSTGNN(Ours)** | **2.17** | **4.08** | **5.06%** | **2.66** | **5.34** | **6.72%** | **3.37** | **6.60** | **8.37%** |

Table 1 shows the experimental performance of DSTGNN and baselines for 15, 30 and 60 minutes ahead prediction on the PeMSD7 and PEMS-BAY datasets. The prediction performance comparison results show that DSTGNN can compete with the state-of-the-art methods in both long-term and short-term predictions on both datasets, while outperforming predefined graph-based spatial-temporal models, namely STGCN[5]and DCRNN[3].

In terms of short-term prediction (<= 30 min), DSTGNN outperforms STTN[11]and GMAN[6]on PEMS-BAY but falls short of Graph WaveNet[10]. It is superior than Graph WaveNet in long-term prediction (60 min), and competitive with STTN, weaker to GMAN. On PeMSD7, DSTGNN displays similar results. This fact specifies that DSTGNN performs better in long-term forecasting due to its use of gated residual connections in DGST block, which sufficiently incorporates related spatial-temporal information and alleviates the accumulation of errors over time.

## 5. Conclusion

In this paper, we propose a novel framework named dynamic gated spatial temporal graph neural network (DSTGNN) for long and short-term traffic forecasting. In DSTGNN, we adopt the DGC module to precisely integrate both static and dynamic spatial correlations and at the same use temporal attention module to capture evolution cues in time series. Besides, we add the gated residual connection to the proposed DGST block for fusing extracted spatial and temporal features. The experiments on two real traffic datasets verifies the effectiveness of DSTGNN on modeling spatial-temporal correlations from time series. In the future, we will consider our DSTGNN for more general spatial-temporal structural graph sequence forecasting tasks, such as preference prediction in recommendation systems.

## 6. Acknowledgements

This work was supported by Wuhan Institute of Technology under Grant No.CX2021277.

## 7. References

[1] Bruna, W. Zaremba, A. Szlam, and Y. Lecun. Spectral networks and locally connected networks on graphs. Computer Science, 2013.

[2] Rongzhou Huang, Chuyin Huang, Yubao Liu, Genan Dai,and Weiyang Kong. Lsgcn: Long short-term traffic prediction with graph convolutional networks. In IJCAI, pages 2355–2361, 2020.

[3] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. arXiv preprint arXiv:1707.01926, 2017.

[4] Usue Mori, Alexander Mendiburu, Maite ́Alvarez, and Jose A Lozano. A review of travel time estimation and forecasting for advanced traveller information systems. Transportmetrica A: Transport Science, 11(2):119–157, 2015.

[5] Bing Yu, Haoteng Yin, and Zhanxing Zhu. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. arXiv preprint arXiv:1709.04875, 2017.

[6] Chuanpan Zheng, Xiaoliang Fan, Cheng Wang, and Jianzhong Qi. Gman: A graph multi-attention network for traffic prediction. In Proceedings of the AAAI conference on artificial intelligence, volume 34, pages 1234–1241, 2020.

[7] Spyros Makridakis and Michele Hibon. Arma models and the box–jenkins methodology. Journal of forecasting, 16(3):147–163, 1997.

[8] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. Advances in neural information processing systems, 27, 2014.

[9] Shengnan Guo, Youfang Lin, Ning Feng, Chao Song, and Huaiyu Wan. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In Proceedings of the AAAI conference on artificial intelligence, volume 33, pages 922–929, 2019.

[10] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, and Chengqi Zhang. Graph wavenet for deep spatial-temporal graph modeling. arXiv preprint arXiv:1906.00121, 2019.

[11] Mingxing Xu, Wenrui Dai, Chunmiao Liu, Xing Gao, Weiyao Lin, Guo-Jun Qi, and Hongkai Xiong. Spatial- temporal transformer networks for traffic flow forecasting. arXiv preprint arXiv:2001.02908, 2020.

[12] Micha¨el Defferrard, Xavier Bresson, and Pierre Van dergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. Advances in neural information processing systems, 29, 2016.

[13] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. Advances in neural information processing systems, 30, 2017.

[14] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907, 2016.

[15] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. Advances in neural information processing systems, 30, 2017.

[16] Petar Veliˇckovi´c, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. arXiv preprint arXiv:1710.10903, 2017.