



STLGCN: Spatial-Temporal Graph Convolutional Network for Long Term Traffic Forecasting

Xuewen Chen, Peng Peng, and Haina Tang^(✉)

School of Artificial Intelligence, University of Chinese Academy of Sciences,
No. 19A, Yuquan Road, Beijing, China
{chenxuewen20, pengpeng21}@mailsucas.ac.cn, hntang@ucas.ac.cn

Abstract. As an essential part of intelligent transportation, accurate traffic forecasting helps city managers make better arrangements and allows users to make reasonable travel plans. Current mainstream traffic forecasting models are developed based on spatial-temporal graph convolutional neural networks, in which appropriate graph structures must be generated in advance. However, most existing graph generation approaches learn graph structures based on local neighborhood relationships of urban nodes, which cannot capture complex dependencies over long spatial ranges. To solve the above problems, we propose Spatial-Temporal Graph Convolutional Neural Network (STLGCN) for long-term traffic forecasting, in which a novel graph generation method is developed by measuring multi-scale correlations among vertices. Meanwhile, a new graph convolution method is proposed for extracting valuable features and filtering out the irrelevant ones, which significantly optimizes the process of spatial information aggregation. Extensive experimental results on two real public traffic datasets, METR-LA and PEMS-BAY, demonstrate the superior performance of our algorithm.

Keywords: traffic forecasting · spatial-temporal graph neural network · long-term forecasting · graph generation

1 Introduction

Intelligent Transportation System (i.e., ITS) effectively applies computer technology and artificial intelligence in transportation and service control, strengthening the connection among vehicles, roads, and users, thus forming a safe, efficient, and accurate comprehensive transportation system. As a critical task in intelligent transportation, traffic prediction significantly impacts traffic management, vehicle allocation, travel time prediction, and other downstream tasks and has become an important research direction in ITS.

The existing traffic forecasting methods can be divided into three categories: statistical methods, traditional machine learning methods, and deep learning

methods. Statistical methods predict future traffic conditions through historical averages or probability modeling. The former corresponds to the historical average model, which assumes that traffic at the exact location has similar daily patterns; Therefore, the historical average can be used as the prediction result. The latter’s representative is the Auto-Regressive Integrated Moving Average model (ARIMA) [1,2], which uses ARIMA to model the time series of traffic volume. However, unexpected situations such as traffic congestion and accidents often occur in the transportation system. In this case, statistical methods may be unable to make accurate predictions.

Machine learning-based methods, which generally rely on support vector machines (SVM) [3] and hidden Markov models for modeling, can better handle unexpected conditions than statistical methods. However, the performance of machine learning-based methods is heavily determined by the effectiveness of feature extraction. Thus, with the popularization of the Global Positioning System (GPS) and the development of traffic sensors, traffic data has accumulated rapidly. These methods are no longer suitable for processing increasingly expanding datasets. In this case, deep learning has gradually gained attention and become the mainstream research direction.

Deep learning methods mainly use Convolutional Neural Networks (CNN) [4,5] or Graph Convolutional Networks (GCN) [6–8] to extract spatial features of nodes and use Recurrent Neural Networks (RNN) [10], CNN, and Attention mechanisms to extract temporal features of nodes. Compared to CNN-based models that only apply to grid format data, Graph Convolutional Networks (GCNs) can extract spatial features from data with non-Euclidean structures. However, its effectiveness highly depends on the quality of the graph and requires prior knowledge of physical topology or traffic networks. In addition, graph structures generated by adjacent relationships typically reflect local correlations between nodes, which is insufficient to capture complex dependency over a long spatial range. In practice, the state of a node may be affected by other non-adjacent nodes. For example, an accident vehicle may cause congestion in the entire transportation network. Therefore, the global correlation between non-adjacent nodes is crucial for long-term prediction, which has been largely ignored in previous models.

To solve the above issues, we propose a new spatio-temporal Graph Neural Network (STLGCN) for long-term traffic prediction. Different from existing methods that require the pre-generation of graph structures, our model adaptively generates multi-scale feature maps by calculating node correlations based on time information. To improve the effectiveness of graph generation, we have designed a self-generated position encoding method. In addition, we propose a new graph convolution method to optimize the use of traffic information. Our contributions are summarized as follows:

- An adaptive graph generation method is proposed for learning a multi-scale feature graph based on temporal information. This method uses self-generated position encoding to improve the effectiveness of the generated graph.

- A novel graph convolution method is proposed to optimize the utilization of related traffic information. This method can better extract valuable features and filter out irrelevant features.
- Extensive experiments conducted on real-world datasets prove that our proposed method achieves higher precision than other baseline methods.

2 Related Work

This section briefly introduces methods used in graph convolution, graph generation, and traffic forecasting.

2.1 Graph Convolution

Graph convolution is a commonly used technology in applications, including natural language processing and social network analysis. It is specifically designed to aggregate information from neighboring nodes through a convolution kernel for processing graph structured data. Based on the way of processing the adjacency matrix, there are two main graph convolution methods: spectral-based and spatial-based [9].

The spectral-based method originates from graph spectral theory. This method defines the graph Fourier transform based on the concept of the graph Laplacian and then establishes the graph filter following the traditional signal processing method. For example, Bruna et al. [8] first defined spectral-based graph convolution based on graph theory. However, this method involves the eigendecomposition of the Laplacian matrix and multiple matrix multiplications. Thus its computational cost is enormous. At the same time, the quantity of learnable parameters defined by its convolutional kernel is equivalent to the number of graph nodes, which might lead to high computational complexity and make the method less effective. Therefore, until the proposal of ChebNet [9], graph convolution had not received attention and development. ChebNet uses Chebyshev polynomials to parameterize convolutional kernels, which significantly reduces time and spatial complexity and gives the characteristics of Spatial Localization.

Similar to traditional CNN convolution on images, spatial-based graph convolution is defined based on the spatial relationships of graph nodes. This method, whose essence is the transmission of node information along the graph edges, integrates the information of the central node and its neighboring nodes to update the feature representation of the central node. There are several main methods in this category. Micheli et al. [9] defined spatial convolution operations through message passing. Atwood et al. [12] proposed that graph convolution is a diffusion process of information between different nodes, and they defined a convolution on a graph based on diffusion theory. And, Velickovic et al. [6] introduced an attention mechanism that uses attention weights to aggregate information about neighboring nodes.

2.2 Graph Generation

Graph generation methods can be divided into two categories [18]. The first generates an adjacency matrix based on the spatial features, while the second is based on the temporal features. In general, most methods are based on the former method regarding the spatial distance or the status of road connections as spatial features. Such spatial graphs generated in this way can effectively capture local correlation in transportation networks. However, it often performs poorly in capturing the global correlation, which is even more crucial for long-term prediction.

There are three main methods for generating graph structures based on temporal features: 1. Metric Learning [22], 2. Probabilistic Modeling [19], and 3. Directly Optimizing [20].

Among them, metric learning aims to learn a metric function that calculates the correlation between nodes. Common methods for metric learning include kernel-based methods and attention-based methods. The former method commonly uses Gaussian or polynomial kernel functions and neural networks. However, attention-based methods use attention mechanisms to calculate correlations, which are more dynamic than metric-based methods.

Probabilistic modeling aims to learn the probability distribution of edges. This method assumes that graphs can be generated by sampling edges, whose probabilities can be modeled by learnable parameters. Probabilistic modeling is often combined with the Bayesian theorem to filter out irrelevant edges.

Direct optimization methods are based on the prior knowledge of the graph and handle the adjacency matrix directly. This method assumes that similar nodes are connected and generates the graph based on this assumption. Direct optimization methods are often combined with GCN for graph generation and use regularization for optimization.

2.3 Traffic Forecasting

Traffic prediction problems aim to predict future traffic status given historical traffic information. The information here is usually provided by sensor networks on the road, and the states between sensor nodes are generally strongly correlated in time and space. Thus, how to capture the implicit spatial and temporal dependencies in data is a critical issue in this field.

In recent years, deep learning-based methods have become the focus of this field. In the early stages, CNN-based methods typically converted urban traffic data into grid format to meet the requirements of image convolution. For example, Guo [21] designed a 3D CNN for capturing spatial-temporal correlations. Yu [15] combined CNN with LSTM for traffic forecasting. Although these methods have achieved improvements over traditional machine learning methods, the transformation process results in the loss of topological information about roads.

Considering the importance of spatio-temporal dependence in traffic prediction problems, spatio-temporal graph convolution is a more suitable choice for processing traffic data. These methods use GCN to capture spatial correlations

between nodes and use CNN, RNN, or Attention Mechanisms to capture temporal correlations, effectively utilizing the topological information of the road. One representative is the DCRNN model proposed by Li [14], which embeds graph convolution in GRU to solve the task of spatio-temporal prediction. And Yu et al. obtained spatio-temporal correlation by stacking spatio-temporal modules constructed by TCN [10] and GCN. In addition, there are methods to directly use graph convolution to get spatio-temporal correlation by generating a spatio-temporal graph [16].

3 Methods

In this section, after introducing some basic concepts, we provide a detailed introduction to the proposed model’s network framework, focusing on the graph generator and spatio-temporal blocks.

3.1 Preliminary

One of the most important goals of traffic forecasting is to predict the future traffic condition given historical features. These traffic features (e.g., velocity, flow, volume) can basically reflect the real-time traffic conditions of road segments in a city.

In this article, we denote the sensor networks as a weighted undirected graph $G = (V, E, W)$, where V is the set of sensor nodes with a number of elements $|V| = N$, while W is the weights of Nodes, and E denotes the set of edges. Based on the above assumptions, the traffic features observed at time t can be denoted by $X_t^P = (x_{i,t}), i = 1, \dots, N$, where i represents the i -th node. The primary purpose of traffic forecasting is to learn a function $f(\cdot)$ which establishes a mapping from T historical signals to T' future signals, i.e.:

$$f : x_{future} = f(x_{historical}), \quad (1)$$

where:

$$\begin{aligned} x_{future} &= [x_{t+1}, \dots, x_{t+T'}] \in R^{T' \times N \times d}, \\ x_{historical} &= [x_{t-T+1}, \dots, x_t] \in R^{T \times N \times d}, \end{aligned}$$

and d is the dimension of the feature.

3.2 Framework

The proposed Spatial-Temporal graph convolutional network framework is shown in Fig. 1, which consists of a Graph Generating block, a Spatial-Temporal block, and a Prediction block.

Among them, the graph generation block generates multi-scale graphs by redefining the neighbors of each node and generating a multi-order neighborhood graph. Additionally, self-generated position encoding is used in the process

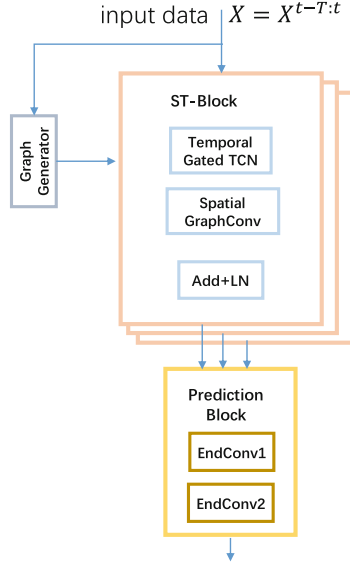


Fig. 1. Framework of STLGCN, consisting of Graph Generator block, Spatial-Temporal blocks and Prediction block.

of graph generating. In the Spatial-Temporal blocks, dilated causal convolutions are utilized to extract the temporal correlation. And a novel graph convolution method is used to capture the spatial dependency of graphs generated in the Graph Generator block. And in the prediction block, two temporal convolutions are used for prediction. The graph-generating module and spatio-temporal module will be introduced in detail in the following text.

3.3 Graph Generation Block

Figure 2 shows the framework of the graphic generation block. The definition of multi-level neighbors, self-generated position encoding methods, and graph generating methods involved in this module will be introduced in this section.

Definition of Multi-order Neighbors. For each pair of nodes $v, u \in V$ in a given graph $G = (E, V)$, if there is a path:

$$p_{v,u}^{(S)} = (e_{v,s_1}, e_{s_1,s_2}, \dots, e_{s_{k-2},s_{k-1}}, e_{s_{k-1},u}), \quad (2)$$

such that connects v, u , then v, u are called multi-order neighbors on graph G ; where $S = \{s_1, \dots, s_{k-1}\}$ represents intermediate nodes that the path $p_{v,u}^{(S)}$ passes through, and $e_{i,j} \in E, \{i, j\} \subset \{v, u, s_1, \dots, s_{k-1}\}$. In addition, the length of the shortest path:

$$k = \min_S |p_{v,u}^{(S)}|, \quad (3)$$

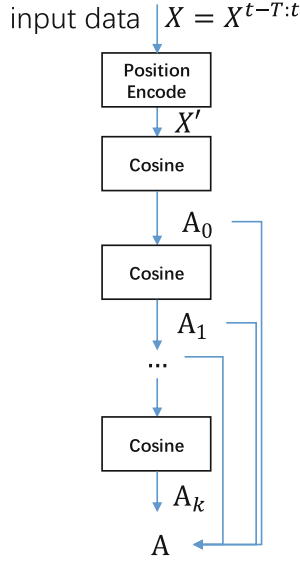


Fig. 2. Graph Generator Block.

connecting v and u on graph G is called the order of the neighbor v, u ; Meanwhile, v, u are called k -hops neighbor to each other.

Self-generated Position Encoding. The success of the Transformer can be attributed, in part, to its ingenious design of position encoding, which enables the model to distinguish different nodes and obtain unique relationships between them. Inspired by this, we propose a self-generated position encoding method, which can simultaneously determine the temporal and spatial patterns of the data and enables the encoding process to be learnable. By applying this method in the graph representation, the model proposed in this article can more effectively capture the temporal and spatial correlations between nodes.

Specifically, let $X_i^{(T)} \in R^{d \times T}$ be the feature of node i at time slot T , where d denotes the dimension of node features. To capture time patterns, a convolutional layer with a kernel size of 1×1 is used for temporal encoding. In terms of spatial encoding, a learnable parameter P is used to distinguish the differences between nodes, which will be adaptively optimized during the training process. Overall, The entire process can be represented as:

$$X' = Conv(X) + P, \quad (4)$$

where $Conv(\cdot)$ represents convolutional layer, and P represents learnable position encoding.

Graph Generation Algorithm. The graph-generation algorithm adopts a metric learning method, using cosine similarity as the kernel function. The original adjacency matrix cap A . sub 1 of graph script cap G is calculated first in graph generating.

$$A_{i,j}^{(1)} = \text{cosine}(X_i, X_j) = \frac{(X_i \cdot X_j)}{\|X_i\| \|X_j\|}, \quad (5)$$

Then, for each pair of nodes v, u , their k -order correlation is generated by all $(k - 1)$ -order correlations of v, u , i.e.:

$$A^{(k)}(v, u) = A_{i,j}^{(k)} = \text{cosine}(A_{i,:}^{(k-1)}, A_{j,:}^{(k-1)}), \quad (6)$$

where $A_{i,:}^{(k-1)}$ represents the corresponding rows of the i -th nodes v in matrix $A^{(k)}$, that is, all its $(k - 1)$ -order correlations. By iteratively solving the above recursive equation, any multi-order graph $A_k, k = 1, 2, \dots$ can be obtained. Finally, the final set of multi-order graphs is obtained by combining all the generated adjacency matrix, i.e.:

$$A = [A^{(1)}, \dots, A^{(k)}], \quad (7)$$

3.4 Spatial-Temporal Blocks

This block sequentially uses dilated convolution and a new graph convolution method to capture temporal and spatial correlations. The framework of the convolution module proposed in this article for capturing spatial correlation is shown in Fig. 3.

Temporal Correlation Capturing. Compared to RNN, CNN-based methods have fewer parameters and are easier to optimize. Therefore, we utilize gated causal convolution with dilation to capture a temporal correlation. The dilated causal convolution increases the Receptive field by increasing expansion, which improves efficiency. In addition, by stacking dilated causal convolutional layers with different kernel sizes, various scale temporal correlations can be effectively captured. This process can be represented as:

$$x \star f(t) = \sum_{s=0}^{K-1} f(s)x(t - d \times s), \quad (8)$$

where d is the size of dilation. In addition, the gating mechanism is adopted to control the transmission of information. Specifically, assuming that the input is $X \in R^{N \times d \times S}$, this final output can be represented as:

$$h = \tau(\theta_1 \star X + b) \odot \sigma(\theta_2 \star X + c), \quad (9)$$

where $\tau(\cdot), \sigma(\cdot)$ represent *Tanh* and *Sigmoid* activation function.

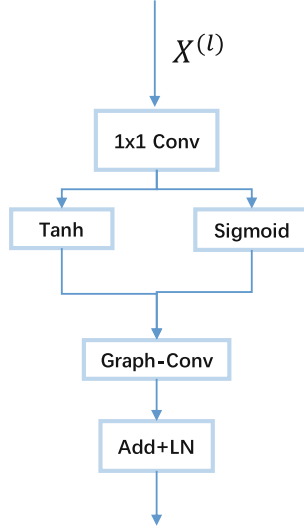


Fig. 3. Spatial-temporal Block.

Spatial Correlation Capturing. Global features, usually implicit, refer to the overall dependency information between nodes, which is crucial for revealing the global correlation between data. Existing GCN methods typically use a spatial graph to aggregate information, which may result in neglecting these global features. For example, when dealing with a fully connected graph, previous methods usually choose a threshold to filter out the irrelevant neighbor information, such as the degree of relevance or the number of neighbors. However, if the threshold value is set inappropriately, the irrelevant information may be included or the valuable information may be ignored while aggregating information from neighbors.

To solve this problem, we propose a novel graph convolution method. When sampling the neighborhoods of nodes, we assume that highly correlated nodes should exist in all graphs with different sampling rates, while unrelated nodes may only exist in those graphs with high sampling rates. Based on this assumption, we use different sampling rates on the sub-graphs generated by the graph generator module to filter out irrelevant influences and enhance the effect of highly correlated neighbors. In addition, the sampling strategy is the same in all sub-graphs. The sampling frequency is a_1, a_2, \dots, N , where $a_1 < a_2 < \dots < N$ and N is the total number of neighbors. We then use these sampled subgraphs to aggregate information from neighbors. As a result, STLGCN can aggregate more useful information from highly related neighbors and ignore the irrelevant information more efficiently.

4 Experiments

4.1 Datasets

Experiments in this paper are conducted on two public transportation datasets: the PEMS-BAY dataset and the METR-LA dataset. The PEMS-BAY dataset records the speed data of 325 road nodes in the California highway network, and the METE-LA dataset contains the traffic speed data of 207 nodes on the Los Angeles expressway. In terms of data processing, the time window size adopted in this paper is 5 min. The data in the dataset is divided into a training set, a validation set, and a test at the ratio of 6:2:2. The specific data are shown in Table 1.

Table 1. The statics of METR-LA and PEMS-BAY

Data	Nodes	edges	Time Window
META-LA	207	1515	34272
PEMS-BAY	325	2369	52116

4.2 Baseline Algorithms

The dynamic spatial-temporal graph convolution is compared with the following models.

- T-GCN [15], which integrates GCN into GRU, is used for traffic forecasting.
- STGCN [13], which combines GCN with one-dimensional causal convolution and adopts a sandwich structure for spatial-temporal relationship acquisition.
- DCRNN [12], which uses diffusion graph convolution and integrates it with an RNN. It adopts an encoder-decoder structure for traffic forecasting.
- STSGCN [14], which generates the spatial-temporal adjacency matrix of nodes and obtains the spatial-temporal relationship through graph convolution.
- AGCRN [11], which proposes a learnable node encoding method.
- Graph WaveNet [2], which adopts improved diffusion graph convolution to obtain spatial relationships, uses dilated graph convolution to obtain temporal relationships, and extracts spatial-temporal relationships through iterative temporal relationship acquisition and spatial relationship acquisition.

4.3 Experiment Settings

All experiments are conducted on a hardware platform equipped with an Intel(R) Xeon(R) Gold 6138 CPU @ 2.00 GHz and an NVIDIA GeForce RTX 2080 Ti. The dilation sizes are defined as 1, 2, 1, 2, 1, 2, 1, 2. There are a total of four spatial-temporal convolution blocks and two diffusion steps. The graph generator block generates zero-order and first-order neighbors. The sampling frequency is

set to 50 nodes, 100 nodes, and all nodes, respectively. The Dropout is set to 0.3, and the Adam optimizer is used. One hundred training rounds are conducted, and the decay index is 0.0001. Besides, the loss function is MAE. The evaluation indexes include MAE, RMSE, and MAPE, where MAE and RMSE reflect the fitting of the model to the extreme values, and MAPE reflects the average prediction of the model.

4.4 Experimental Results

Table 2. Performance of STLGCN and other baselines.

data	model	30 min			45 min			60 min		
		MAE	MAPE	RMSE	MAE	MAPR	RMSE	MAE	MAPE	RMSE
METR-LA	TGCN	5.67	14.09%	9.89	6.21	15.81%	10.81	6.68	17.37%	11.53
	STGCN	3.62	10.43%	7.27	4.05	12.04%	8.24	4.44	13.53%	9.00
	DCRNN	5.11	12.85%	9.89	5.92	15.01%	11.20	6.66	16.94%	12.22
	STSGCN	4.07	10.37%	9.64	4.68	12.15%	11.00	5.18	13.62%	12.07
	AGCRN	4.79	10.56%	12.08	5.58	11.92%	13.80	6.21	13.08%	14.96
	Graph WaveNet	3.16	8.55%	6.31	3.45	9.54%	7.03	3.67	10.28%	7.53
	STLGCN	3.14	8.51%	6.31	3.40	9.41%	6.91	3.58	10.17%	7.30
PEMS-BAY	TGCN	2.48	5.58%	4.53	2.77	6.33%	5.04	3.06	7.04%	5.50
	STGCN	1.94	4.41%	4.27	2.25	5.31%	5.02	2.50	6.06%	5.63
	DCRNN	1.80	4.07%	3.91	2.05	4.79%	4.52	2.24	5.31%	4.92
	STSGCN	1.89	4.28%	4.19	2.15	5.05%	4.85	2.36	5.60%	5.28
	AGCRN	1.74	4.01%	3.90	1.92	4.53%	4.37	2.07	4.93%	4.70
	Graph WaveNet	1.77	3.92%	3.72	1.97	4.50%	4.20	2.12	4.90%	4.78
		STLGCN	1.68	3.77%	3.78	1.87	4.39%	4.24	2.02	4.84%

This paper mainly compares the long-term forecasting ability of the models. To this end, the time window sizes of 30 min, 45 min, and 60 min are adopted in the selection of the forecast horizon. The experiments of all models are conducted in the same environment, and the training parameters of the models are kept consistent. The prediction results of the models are shown in Table 2. It can be seen that STLGCN shows the best result in all indexes. With the increase in the prediction time, the improvement of MAPE is more significant. On the METR-LA dataset, compared with the optimal model, the MAPE of STLGCN at 30 min, 45 min, and 60 min is improved by 0.04%, 0.13%, and 0.11%, respectively. On the PEMS-BAY dataset, compared with the optimal model, the MAPE of STLGCN at 30 min, 45 min, and 60 min is improved by 0.08%, 0.06%, and 0.03%, respectively. In the baseline models, the performance of the models using the RNN is poor. For example, the three models (TGCN, DCRNN, and AGCRN) perform worse than the TCN-based model. Among the baseline models, the best-performing model is Graph WaveNet, and this is highly related to the diffusion graph convolution it used. STSGCN builds a spatial-temporal graph and exploits the spatial-temporal correlations more powerfully, so its performance is relatively better.

5 Conclusions

In this study, we propose a novel spatial-temporal graph convolutional network for long-term traffic forecasting (STLGCN). Our model adaptively generates a multi-order graph using a graph generator that incorporates self-generated position encoding to enhance the effectiveness of the generated graph. Additionally, we propose a graph convolution method to extract useful traffic information from the generated graph while filtering out irrelevant data, improving the model's ability to capture spatial-temporal correlations.

Acknowledgement. This work is supported by the National Natural Science Foundation of China (NSFC) (Grant No. 52071312), and the Open Program of Zhejiang Lab (Grant No. 2019KE0AB03).

References

1. Fattah, J., Ezzine, L., Aman, Z., Moussami, H., Lachhab, A.: Forecasting of demand using ARIMA model. *Int. J. Eng. Bus. Manag.* **10** (2018). <https://doi.org/10.1177/1847979018808673>
2. Xie, Y., Zhang, P., Chen, Y.: A fuzzy ARIMA correction model for transport volume forecast. *Math. Problems Eng.* **2021**, 6655102, 10 p. (2021). <https://doi.org/10.1155/2021/6655102>
3. Kim, K.: Financial time series forecasting using support vector machines. *Neurocomputing* **551**, 307–319 (2003). [https://doi.org/10.1016/S0925-2312\(03\)00372-2](https://doi.org/10.1016/S0925-2312(03)00372-2)
4. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA (2016). <https://doi.org/10.1109/CVPR.2016.90>
5. Szegedy, C., et al.: Going deeper with convolutions. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA, pp. 1–9 (2015). <https://doi.org/10.1109/CVPR.2015.7298594>
6. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y.: Graph attention networks. In: *International Conference on Learning Representations* (2018). <http://openreview.net/forum?id=rJXMpikCZ>
7. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: *International Conference on Learning Representations* (2016). <http://openreview.net/forum?id=SJU4ayYgl>
8. Bruna, J., Zaremba, W., Szlam, A., LeCun, Y.: Spectral networks and deep locally connected networks on graphs. In: *2nd International Conference on Learning Representations, ICLR, Banff, Canada* (2014)
9. Michael, D., Xavier, B., Pierre, V.: Convolutional neural networks on graphs with fast localized spectral filtering. In: *International Conference on Neural Information Processing Systems (NIPS)*, pp. 3844–3852 (2016)
10. Shi, X., Chen, Z., Wang, H., Yeung, D., Wong, W., Woo, W.: Convolutional LSTM network: a machine learning approach for precipitation nowcasting. In: *International Conference on Neural Information Processing Systems (NIPS)*, 802–810 (2015)

11. Wu, Z., Pan, S., Long, G., Jiang, J., Zhang, C.: Graph WaveNet for deep spatial-temporal graph modeling. In: International Joint Conference on Artificial Intelligence, pp. 1907–1913 (2019). <https://doi.org/10.5555/3367243.3367303>
12. Atwood, J., Towsley, D.: Diffusion-convolutional neural networks. In: Neural Information Processing Systems, pp. 2001–2009, Barcelona, Spain (2016). <https://doi.org/10.5555/3157096.3157320>
13. Bai, L., Yao, L., Li, C., Wang, X., Wang, C.: Adaptive graph convolutional recurrent network for traffic forecasting. In: Neural Information Processing Systems (2020). <https://doi.org/10.5555/3495724.3497218>
14. Li, Y., Yu, R., Shahabi, C., Liu, Y.: Diffusion convolutional recurrent neural network: data-driven traffic forecasting. In: 6th International Conference on Learning Representations (ICLR) (2018). <http://openreview.net/forum?id=SJiHXGWAZ>
15. Yu, B., Yin, H., Zhu, Z.: Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting. In: International Joint Conference on Artificial Intelligence (IJCAI) Stockholm, Sweden (2018). <https://doi.org/10.5555/3304222.3304273>
16. Song, C., Lin, Y., Guo, S., Wan, H.: Spatial-temporal synchronous graph convolutional networks: a new framework for spatial-temporal network data forecasting. In: Proceedings of the AAAI Conference on Artificial Intelligence **34**(01), 914–921 (2022). <https://doi.org/10.1609/aaai.v34i01.5438>
17. Zhao, L., et al.: T-GCN: a temporal graph convolutional network for traffic prediction. *IEEE Trans. Intell. Transport. Syst.* **21**(9), 3848–3858 (2020). <https://doi.org/10.1109/TITS.2019.2935152>
18. Zhu, Y., et al.: A survey on graph structure learning: progress and opportunities. In: International Joint Conference on Artificial Intelligence (2021)
19. Franceschi, L., Niepert, M., Pontil, M., He, X.: Learning discrete structures for graph neural networks. In: International Conference on Machine Learning (ICML) pp. 1972–1978 (2019). <https://proceedings.mlr.press/v97/franceschi19a.html>
20. Gao, X., Hu, W., Guo, Z.: Exploring structure-adaptive graph learning for robust semi-supervised classification. In: 2020 IEEE International Conference on Multimedia and Expo (ICME), pp. 1–6, London, UK (2020). <https://doi.org/10.1109/ICME46284.2020.9102726>
21. Guo, S., Lin, Y., Li, S., Chen, Z., Wan, H.: Deep spatial-temporal 3D convolutional neural networks for traffic data forecasting. *IEEE Trans. Intell. Transp. Syst.* **20**(10), 3913–3926 (2019). <https://doi.org/10.1109/TITS.2019.2906365>
22. Shao, Z., et al.: Decoupled dynamic spatial-temporal graph neural network for traffic forecasting. *Proc. VLDB Endow.* **16**(16), 2733–2746 (2022). <https://doi.org/10.14778/3551793.3551827>
23. Jiang, J., Han, C., Zhao, W., Wang, J.: PDFormer: propagation delay-aware dynamic long-range transformer for traffic flow prediction. In: AAAI (2023)