Dual Dynamic Spatial-Temporal Graph Convolution Network for Traffic Prediction

Yanfeng Sun, *Member, IEEE*, Xiangheng Jiang, Yongli Hu, *Member, IEEE*, Fuqing Duan, *Member, IEEE*, Kan Guo, Boyue Wang, Junbin Gao, *Member, IEEE*, and Baocai Yin, *Member, IEEE*

***Abstract*— Recently, Graph Convolution Network (GCN) and Temporal Convolution Network (TCN) are introduced into traffic prediction and achieve state-of-the-art performance due to their good ability for modeling the spatial and temporal property of traffic data. In spite of having good performance, the current methods generally focus on the traffic measurement of road segments, i.e. the nodes of traffic flow graph, while the edges of the graph, which represent the correlation of traffic data of different road segments and form the affinity matrix for GCN, are usually constructed according to the structure of road network, but the spatial and temporal properties are not well exploited in their theories. In this paper, we propose a Dual Dynamic Spatial-Temporal Graph Convolution Network (DDSTGCN), which not only models the dynamic property of the nodes of the traffic flow graph but also captures the dynamic spatial-temporal feature of the edges by transforming the traffic flow graph into its dual hypergraph. The traffic prediction is enhanced by the collaborative convolutions on the traffic flow graph and its dual hypergraph. The proposed method is evaluated by extensive traffic prediction experiments on six real road datasets and the results show that it outperforms state-of-the-art related methods. Source codes are available at *https://github.com/j1o2h3n/DDSTGCN*.**

***Index Terms*— Graph convolution network, traffic prediction, hypergraph, intelligent transportation systems.**

I. Introduction

WITH the rapidly increasing population and vehicles of big cities, the transportation system is becoming complicated and is suffering from many traffic issues like congestion. Thus, it is urgent to construct Intelligent Transportation Systems (ITS) for the modern city. In this situation, as one of the most important tasks in ITS, traffic prediction raises much attention and becomes a hot topic in this field. The main challenge of traffic prediction is how to model the dynamic spatial and temporal property of larger-scale traffic data.

In the past decades, many traffic prediction methods have been proposed [1]. In the early stage, the linear methods were firstly proposed, such as Historical Average (HA) method [2] and Moving Average (MA) method, which averaged historical data to simulate the future traffic trends. Later, some statistical and data-driven methods were proposed, such as K-Nearest Neighbor (KNN) [3] and other methods based on clustering models [4], the Auto Regressive Integrated Moving Average (ARIMA) method [5], [6], [7] and its extensions [8], [9], [10], which integrated Auto Regressive (AR) and MA model and solved the defect of non-stationary hypothesis by difference, and the Vector Auto Regressive (VAR) method [11], which regressed the lagged variables on the basis of AR and generalizes to multivariate series. The above methods are simple and effective traffic prediction methods. They run fast but with low prediction accuracy. Afterwards, some methods with good generalization ability and small error have been proposed, such as Kalman Filters (KFs) and its extensions, the Extended Kalman Filters (EKFs) methods [12], [13], [14], [15], [16], [17], [18], [19], the variant models of Support Vector Regression (SVR) [20], [21], [22], [23], [24] of Support Vector Machine (SVM) [25], the Random Forest Regression (RFR) methods [26], [27], [28], the Fuzzy Logic Regression (FLR) methods [29], [30], and the hybrid models [31], [32], etc. However, these methods do not perform very well from the current point of view.

Compared with the above conventional methods, the neural network based methods have been proposed and widely used in traffic prediction due to neural network has good representation for the nonlinear relation among traffic data, such as the Neural Networks (NNs) methods [33], [34], the Bayesian Networks (BNs) methods [35], [36], the Recurrent Neural Network (RNN) method [37], etc. With the great progress

Manuscript received 18 August 2021; revised 26 May 2022 and 17 August 2022; accepted 13 September 2022. Date of publication 5 October 2022; date of current version 5 December 2022. This work was supported in part by the Beijing Natural Science Foundation under Grant 4222021; in part by the National Natural Science Foundation of China under Grant U21B2038, Grant U19B2039, Grant U1811463, and Grant 62172023; in part by the Research and Deveopment Program of Beijing Municipal Education Commission under Grant KZ202210005008; and in part by the Beijing Talents Project under Grant 2017A24. The Associate Editor for this article was N. Bekiaris-Liberis. *(Corresponding author: Yongli Hu.)*

Yanfeng Sun, Xiangheng Jiang, Yongli Hu, Kan Guo, and Boyue Wang are with the Beijing Key Laboratory of Multimedia and Intelligent Software Technology, Faculty of Information Technology, Beijing Institute of Artificial Intelligence, Beijing University of Technology, Beijing 100124, China (e-mail: yfsun@bjut.edu.cn; jiangxiangheng@gmail.com; huyongli@bjut.edu.cn; guokan.cn@gmail.com; wby@bjut.edu.cn).

Fuqing Duan is with the School of Artificial Intelligence, Beijing Normal University, Beijing 100875, China (e-mail: fqduan@bnu.edu.cn).

Junbin Gao is with the Discipline of Business Analytics, The University of Sydney Business School, The University of Sydney, Camperdown, NSW 2006, Australia (e-mail: junbin.gao@sydney.edu.au).

Baocai Yin is with the Beijing Key Laboratory of Multimedia and Intelligent Software Technology, Faculty of Information Technology, Beijing Institute of Artificial Intelligence, Beijing University of Technology, Beijing 100124, China, and also with the Faulty of Electronic Information and Electrical Engineering, Dalian University of Technology, Dalian 116024, China (e-mail: ybc@bjut.edu.cn).

Digital Object Identifier 10.1109/TITS.2022.3208943

achieved in the deep learning field [38], many deep learning based traffic prediction methods [39], [40], [41], [42] have been proposed and have obtained state-of-the-art performance due to deep neural networks can capture more complex data features. In these methods, to learn the temporal features of traffic data, the temporal neural networks such as LSTM [39], [40], [43] and GRU [44], [45] are generally adopted, but the sequential learning models are difficult to train with high calculation. To learn the spatial features of traffic data, some CNN [43], [46] based methods like ST-ResNet [41] are introduced, but these methods represent the city's road network as images for convolution operator, which will lose the natural graph structure of the traffic network.

Recently, Graph Convolutional Network (GCN) [47], [48] and Temporal Convolutional Network (TCN) [49], [50] have been introduced into traffic prediction. As GCN can retain the graph structure information of the traffic network and TCN can capture temporal features of traffic flow, their integration results in a set of GCN based traffic prediction methods with high performance, such as STGCN [51], Graph WaveNet [52] and DGCNN [53]. The GCN based traffic prediction methods depend on how to construct the graph or the adjacent matrix, which represents the traffic measurements of the road segments as the nodes and represents the relations of the traffic measurements of the road segments as the edges. Simply, researchers justly construct the graph by the direct connections of different road segments [54]. To achieve a more precise description, the distance of the road segments is considered in the relations of the traffic measurements of the road segments [55], and the similarity of geographic functions are also taken into account [56]. As the traffic flow measurements on the road segments and their correlations will change dynamically in space and time, these static graphs can not model its dynamic property. For example, traffic accidents and congestion often happen in the real world, which will affect the dynamic traffic flow and also change the correlation of traffic flow on the relevant road segments. For this purpose, some latest methods try to construct a dynamic graph or the adjacent matrix for GCN by the traffic data observed in real time [53], [57]. It is proved that the dynamic graph based on GCN methods have obtained considerable improvement compared with these static models [58], [59], [60].

The dynamic graph based on GCN methods give a perspective way for traffic forecasting by modeling the dynamic property of the traffic flow graph, but this problem is far from being completely resolved. For the complicated transportation system and its dynamic traffic data, the current models based on dynamic graph only utilize the intuitive traffic data dependency between nodes of road segments and represent them as dynamic edges. The high-order dependency of the dynamic traffic graph is not well exploited, such as the dependency between these dynamic edges. How to reveal and utilize the latent and high-order dependency in the dynamic traffic graph to improve the traffic prediction performance is a new and significant problem. For the dependency between dynamic edges, compared with the nodes which have definite values of traffic measurements, the edges have no directly obtainable features except for the direction and structure of traffic flow graph. Therefore, how to establish dynamic edge features is another important problem.

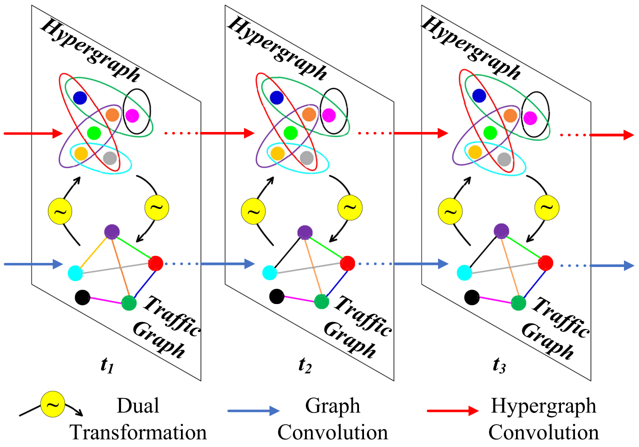


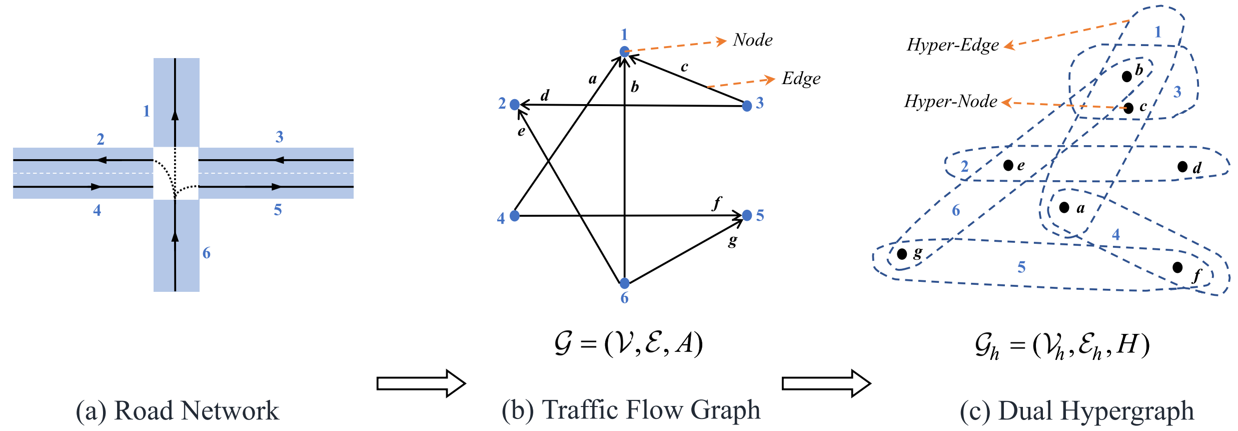
Fig. 1. The traffic flow graph are transformed into its dual hypergraph and two streams of dynamic GCNs are implemented on the sequences of the graph and hypergraph, which can fully explore the spatial and temporal property of traffic data.

To solve the above problems, we propose a novel Dual Dynamic Spatial-Temporal Graph Convolution Network (DDSTGCN) for traffic prediction. As shown in Fig.1, the main idea of DDSTGCN is to construct a dual hypergraph of traffic flow graph by a dual transformation to capture the correlations of the edges of traffic flow graph, i.e. the nodes of the hypergraph are actually the edges of the original traffic flow graph, and vice versa. Moreover, to reveal the complicated spatial-temporal property of the dynamic transportation system, we implement GCNs on both of the sequences of hypergraph and graph, in which a Dynamic Interaction Module (DIM) is designed to propagate features between the dual dynamic GCNs. Due to the above features of the representation of edges correlations and the collaborative GCNs on the two graphs with merged spatial-temporal features, our model gains considerable improvement compared with current GCN-based methods, which is validated by the extensive traffic prediction experiments in the following section of experiments. The main contributions of this work are listed as follows,

- A novel framework of Dual Dynamic Spatial-Temporal Graph Convolution Network (DDSTGCN) is proposed for traffic prediction, which is featured by the two streams of GCNs on the sequences of traffic flow graph and its dual hypergraph with interaction.
- Different from the current GCN based traffic prediction methods, the proposed DDSTGCN exploits the dynamic correlations of the edges of traffic flow graph by its dual hypergraph, which helps to reveal more complicated latent relations of the dynamic transportation system.
- The proposed DDSTGCN is extensively evaluated on six real traffic datasets and the results show that DDSTGCN outperforms state-of-the-art related works.

II. Related Works and Preliminary

In this section, we firstly review the GCN based traffic prediction methods, then give the preliminary knowledge of


1
2
3
4
5
6
Node
Edge
a
b
c
d
e
f
g
Hyper-Edge
Hyper-Node
$\mathcal{G} = (\mathcal{V}, \mathcal{E}, A)$
$\mathcal{G}_h = (\mathcal{V}_h, \mathcal{E}_h, H)$

(a) Road Network (b) Traffic Flow Graph (c) Dual Hypergraph

Fig. 2. The road network, the traffic flow graph and the dual hypergraph.

traffic prediction and Graph/Hypergraph Convolution Networks, and finally briefly review the spatial and temporal feature representation for traffic prediction.

A. GCN Based Traffic Prediction

Traffic prediction is a classic problem in ITS. As mentioned above, there are many different types of traffic prediction methods proposed in the past decades. Considering that there are many comprehensive reviews on the topic [61], [62], [63], [64], [65], [66] and our method is built on the latest GCN technique, we justly introduce some latest and representative GCN based traffic prediction methods as follows,

STGCN [51]: STGCN proposes a pure convolution model by combining GCN and TCN, which can well extract the spatial-temporal features of traffic data.

Graph WaveNet [52]: Graph WaveNet proposes an adaptive adjacency matrix to capture hidden spatial correlation of traffic data, and adopts a 1D dilated convolution to capture temporal correlation.

ASTGCN [57]: ASTGCN uses a self-attention mechanism to construct a dynamic mask matrix to capture the dynamic characteristics of traffic data.

DGCN [58]: DGCN proposes a latent network to estimate the dynamic Laplace matrix of the nodes of traffic flow graph, which constructs the traffic flow graph adaptively from the dynamic traffic data.

STSGCN [67]: STSGCN proposes a spatial-temporal synchronously modeling mechanism to capture the spatial-temporal correlation and heterogeneity of traffic data effectively.

MRA-BGCN [68]: MRA-BGCN proposes to model the spatial properties of the nodes and edges of traffic flow graph by bicomponent graph convolutional network, and then uses GRU to model the temporal correlation of the nodes of traffic flow graph.

B. Problem Definition of Traffic Prediction

The traffic road network can be defined as a weighted directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, A)$, where the road segments represent nodes, and the relation or correlation of the traffic flow on the nodes of road segments represent edges. The corresponding relationship is shown in Fig.2 (a) and (b), where $\mathcal{V}$ is a set of nodes of road segments with the cardinality $|\mathcal{V}| = N$, and $\mathcal{E}$ is a set of edges and $A \in \mathbb{R}^{N \times N}$ is the adjacent matrix with elements representing the weights of the edges, which can be obtained from the links of the road network or the distances between road segments. The traffic data (e.g. flow or speed) of a road network at time t can be represented as $\mathbf{X}_t = \left(x_t^1, x_t^2, \dots, x_t^N\right)^\top \in \mathbb{R}^{N \times F}$, where $x_t^i \in \mathbb{R}^F$ is the measurement of ith node at time t, and F is the feature size. For a period $[1, \dots, \tau]$, the sequence of the traffic data can be represented as $\mathcal{X}^{1:\tau} = (\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_\tau)^\top \in \mathbb{R}^{N \times F \times \tau}$. Thus, the traffic prediction problem can be interpreted as, given a traffic flow graph $\mathcal{G}$ and the historical traffic data of T' slots before the time t, a function $\mathcal{F}$ is learned to predict the traffic data of T slots after time t, which can be formulated as follows,

$$\mathcal{F}(\mathcal{X}^{(t-T'+1):t}, \mathcal{G}) = \mathcal{X}^{(t+1):(t+T)} \tag{1}$$

where $\mathcal{X}^{(t-T'+1):t} \in \mathbb{R}^{N \times F \times T'}$, $\mathcal{X}^{(t+1):(t+T)} \in \mathbb{R}^{N \times F \times T}$.

C. Graph and Hypergraph Convolution Networks

GCN [47], [69] is a convolution network on graph, which updates nodes feature by aggregating and transforming neighbor nodes features. Kipf *et al.* [48] proposed a first-order Chebyshev polynomial approximation form of GCN as follows,

$$GCN(\mathbf{X}) = \hat{A}\mathbf{X}\Theta \tag{2}$$

where $\hat{A} \in \mathbb{R}^{N \times N}$ is the normalized adjacency matrix, $\mathbf{X} \in \mathbb{R}^{N \times F_1}$ is the input feature matrix, and $\Theta \in \mathbb{R}^{F_1 \times F_2}$ is the parameter to learn.

Hypergraph Convolution Network (HGCN) [70] can be regarded as the extension of GCN from graph to hypergraph. Feng *et al.* [71] proposed a hypergraph convolution as follows,

$$HGCN(\mathbf{X}_h) = D_{hv}^{-\frac{1}{2}} H W D_{he}^{-\frac{1}{2}} H^\top D_{hv}^{-\frac{1}{2}} \mathbf{X}_h \Theta \tag{3}$$

where $D_{he} \in \mathbb{R}^{N\times N}$ and $D_{hv} \in \mathbb{R}^{E\times E}$ denote the diagonal matrices of the hyper-edge degrees and the hyper-node degrees, respectively. Here, we call the node of hypergraph as hyper-node for distinguishing from that of the general graph. $H \in \mathbb{R}^{E\times N}$ is the incidence matrix, $W \in \mathbb{R}^{N\times N}$ is the weighted diagonal matrix of the hyper-edge, generally the identity matrix. $\mathbf{X}_h \in \mathbb{R}^{E\times F_1}$ is the input feature matrix, and $\Theta \in \mathbb{R}^{F_1\times F_2}$ is the learnable parameter.

D. Spatial and Temporal Feature Representation for Traffic Prediction

The core of traffic prediction is to properly model the spatial and temporal properties of the dynamic traffic data. The vanilla GCN in (2) cannot effectively simulate the stochastic nature of the traffic dynamics in space. Therefore, Li *et al.* [55] proposed a graph signal diffusion process with N-order finite step size to simulate the traffic space modeling as follows,

$$GCN^d(\mathbf{X}) = \sum_{n=0}^{N-1} \hat{A}^n \mathbf{X}\Theta_n \tag{4}$$

where $\hat{A}^n \in \mathbb{R}^{N\times N}$ is the n-th power of the normalized adjacency matrix $\hat{A}$, N is the diffusion coefficient, and $\Theta = [\Theta_0, \dots, \Theta_{N-1}] \in \mathbb{R}^{N\times F_1\times F_2}$ is the learnable parameter. In order to further capture both upstream and downstream effects of traffic data in the directed traffic graph, the diffusion process is extended to the bidirectional model [55] as follows,

$$GCN^d(\mathbf{X}) = \sum_{n=0}^{N-1} A_f^n \mathbf{X}\Theta_{n,f} + A_b^n \mathbf{X}\Theta_{n,b} \tag{5}$$

where A_f and A_b are the normalization of the forward adjacency matrix A and the backward adjacency matrix A^T respectively, $\Theta_f = [\Theta_{0,f}, \dots, \Theta_{N-1,f}]$ and $\Theta_b = [\Theta_{0,b}, \dots, \Theta_{N-1,b}]$ are learnable parameters.

RNNs are typical temporal modeling methods, which are widely used to extract temporal correlations of data in traffic prediction tasks. However, the RNNs based methods suffer from low processing efficiency for long sequences and cannot process data in parallel. For this reason, Bai *et al.* [49] proposed TCNs combining the one-dimensional causal convolution and the dilated convolution as follows,

$$Conv_g(\mathbf{x})[t] = \sum_{i=0}^{T_0-1} g(i)\mathbf{x}(t - m \times i) \tag{6}$$

where $\mathbf{x} \in \mathbb{R}^T$ is the input sequence, $Conv_g(\cdot)$ is the dilated convolution operation with convolution kernel of $g \in \mathbb{R}^{T_0}$ and dilation factor of m, and $Conv_g(\mathbf{x})[t] \in \mathbb{R}^1$ is the output at step t. TCNs is not only with lower computational cost, but also has a larger receptive field of time dimension. Later, inspired by the gating mechanism for sequence modeling [72], Wu *et al.* [52] proposed a Gate Temporal Convolutional Network (Gate-TCN), which attempts to model more complex temporal dependencies in traffic data by restricting the flow of feature streams.

III. Methodology

The framework of the proposed DDSTGCN is shown in Fig.3. The input traffic data is first processed by a linear transformation module and then fed into the stacked Dual Spatial-Temporal Blocks (DST-Blocks), which are the core components to implement our dual dynamic graph convolution on the traffic flow graph and its dual hypergraph. Then the learned features are fused by a skip connection. Finally, by the Leaky ReLU and linear transformation layers, the final traffic prediction results are obtained. We will introduce the details of these components in the following.

A. Dual Spatial-Temporal Block

The architecture of DST-Block dynamically captures the spatial-temporal features of the nodes and edges of traffic flow graph, in which the edges features are represented as the dual hypergraph. The residual connections are made between different stacked DST-Blocks. Each DST-Block contains the dual transformations between traffic flow graph and its hypergraph, and three modules of the dynamic traffic graph convolution, the dynamic hypergraph convolution and the Dynamic Interaction Module (DIM). The dual transformations realize the mapping from the traffic flow graph to its dual hypergraph and the inverse mapping. The dynamic graph and hypergraph convolution modules implement convolution on the traffic flow graph and its dual hypergraph to capture the features of the nodes and edges of the traffic data. The DIM module interchanges feature between the two streams of GCNs to obtain more effective information for traffic prediction. The details of these modules are given in the following.

1) Dual Transformation: The dual transformation of a graph is defined as a mapping that maps the nodes of the graph to edges of the target and maps the edges of the graph to the nodes of the target. The result of the dual transformation of a graph is usually a hypergraph as a node in the original graph may have more than 2 edges, which means the corresponding edge of the dual graph will link more than 2 nodes. Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, A)$ with N nodes and E edges, we denote the dual hypergraph of $\mathcal{G}$ by $\mathcal{G}_h = (\mathcal{V}_h, \mathcal{E}_h, H)$, as shown in Fig.2 (b) and (c), it is obvious that $|\mathcal{V}_h| = E$ and $|\mathcal{E}_h| = N$. $H \in \mathbb{R}^{E\times N}$ is the incidence matrix of $\mathcal{G}_h$ defined as follows,

$$[H]_{ij} = \begin{cases} 1 & if\ v_i \in e_j; \\ 0 & otherwise. \end{cases} \tag{7}$$

where $v_i \in \mathcal{V}_h$ and $e_j \in \mathcal{E}_h$. The incidence matrix H can simultaneously describe the structural information of the original graph and the dual hypergraph. Considering that the traffic flow graph generally is a directed graph as the accessibility of the road network. Thus, we decompose H into two parts, i.e. $H = H_{src}+H_{dst}$, where $H_{src}, H_{dst} \in \mathbb{R}^{E\times N}$ are the matrices with no-zero elements represent the source nodes and the destination nodes of the directed edges in the original graph, respectively. From H_{src}, H_{dst}, we have the following dual transformations.

- Graph to Hypergraph Transformation

The feature of the hyper-node of the dual hypergraph corresponds to the edge feature of the traffic flow graph,

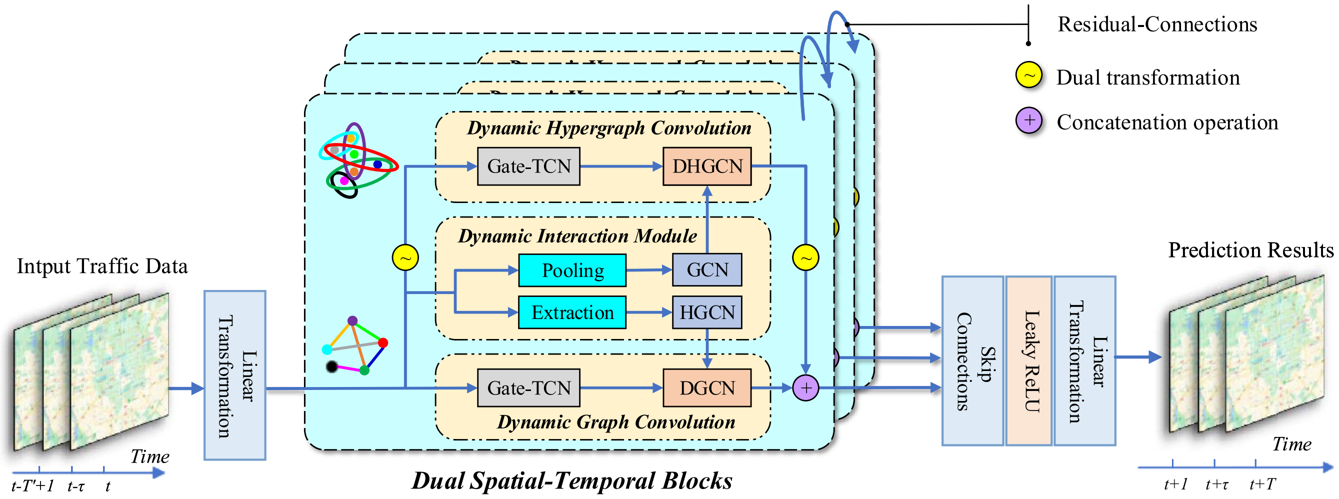


Fig. 3. The framework of DDSTGCN. It consists of multiple stacked Dual Spatial-Temporal Blocks (DST-Blocks), input layer and output layer. Each DST-Block has residual connections, and sum up the output of the block by skip-connection and input it into the output layer.

which is generally defined by the connections or the distance between the nodes of road segments. In fact, the edges feature contains the correlations of dynamic traffic data of nodes of road segments and the static information of the road network, so the feature of the hyper-node should have new definition. Here, we define the hyper-node feature $\mathcal{X}_h$ of the hypergraph by the edge and node features of the traffic flow graph as follows,

$$\mathcal{X}_h = [(W_1 \odot H_{src})\mathcal{X}; (W_2 \odot H_{dst})\mathcal{X}; \mathcal{X}_{dis}] \tag{8}$$

where $\mathcal{X}_h \in \mathbb{R}^{E\times(2F+1)\times T}$, $\odot$ means Hadamard product, $[\cdot;\cdot;\cdot]$ is the concatenation operation, $\mathcal{X} \in \mathbb{R}^{N\times F\times T}$ is the nodes features, $\mathcal{X}_{dis} \in \mathbb{R}^{E}$ is the distance matrix of road network, and $W_1, W_2 \in \mathbb{R}^{E\times N}$ are the learning parameters. The first two parts in (8) represent the dynamic features of directed edges, and the last part is the static features of the edges. The correlation between edges and nodes of the traffic flow graph can be captured by the weight matrices W_1, W_2.

- Hypergraph to Graph Transformation

The dual transformation from the hypergraph to the graph is to transform the feature of hypergraph back to the traffic flow graph. Here we map the feature of the hyper-nodes of hypergraph back to the feature of the nodes of the traffic flow graph by the following formula,

$$\mathcal{X}' = (W_3 \odot H)^{\mathsf{T}}\mathcal{X}'_h \tag{9}$$

where $\mathcal{X}' \in \mathbb{R}^{N\times F'\times T}$ and $\mathcal{X}'_h \in \mathbb{R}^{E\times F'\times T}$. Besides, $W_3 \in \mathbb{R}^{E\times N}$ is the learnable parameter for capturing the correlation between the edges and the nodes of the dual hypergraph.

In the traffic flow graph, the number of edges is usually greater than the number of nodes, which will bring a large number of nodes for the hypergraph and so add too much complexity for GCNs on the hypergraph. In addition, too many redundant edges maybe introduce noise or interference for the hypergraph. For this purpose, it is necessary to select a more representative part of edges in the traffic flow graph for reducing the number of nodes in the dual hypergraph. Here, we simply adopt the Top-k sampling method, i.e. only the top-k edges with the largest weight of a node are selected for dual transformation, which will ensure the main information of the traffic flow graph is transformed into its dual hypergraph.

2) Dynamic Interaction Module (DIM): To capture more complicated and dynamic correlations of traffic data and exchange information between the traffic flow graph and its dual hypergraph, a dynamic interaction module is designed to produce dynamic edges or adjacent matrices used in the convolution of the graph and hypergraph. As shown in Fig.3, there are two branches in the dynamic interaction module, which generate dynamic edges or adjacent matrices for the graph and hypergraph, respectively.

- Dynamic hyper-edge generating for hypergraph

We utilize the dual relation between the traffic flow graph and its hypergraph to generate dynamic hyper-edges for hypergraph, i.e. utilizing GCN on the traffic flow graph to update the feature of nodes and so update the corresponding edge of hypergraph. Thus, we first perform the average pooling on the node features $\mathcal{X}$ of the traffic flow graph in the time dimension:

$$\mathbf{X}^d = AvgPool(\mathcal{X}) \in \mathbb{R}^{N\times F} \tag{10}$$

Then, we adopt the diffusion GCN [55] in (5) to capture the spatial correlation of these changes by

$$GCN^d(\mathbf{X}^d) = \sum_{n=0}^{N-1} A_f^n\mathbf{X}^d\Theta_{n,f} + A_b^n\mathbf{X}^d\Theta_{n,b} \tag{11}$$

where $A_f = A/rowsum(A)$ and $A_b = A^{\mathsf{T}}/rowsum(A^{\mathsf{T}})$, A_f^n and A_b^n are the n-order power matrices of A_f and A_b respectively. $\Theta_f = [\Theta_{0,f}, \ldots, \Theta_{N-1,f}] \in \mathbb{R}^{N\times F\times 1}$ and $\Theta_b = [\Theta_{0,b}, \ldots, \Theta_{N-1,b}] \in \mathbb{R}^{N\times F\times 1}$ are learnable parameters. The output $GCN^d(\mathbf{X}^d) \in \mathbb{R}^N$ will be used as the dynamic hyper-edges in the following DHGCN block of the hypergraph.

- Dynamic edge generating for the traffic flow graph

Similarly, the dynamic edges of the traffic flow graph can be generated from the feature of the corresponding nodes in the hypergraph. For this purpose, we firstly extract the correlation between the two end nodes of the edge in the traffic flow graph as the initial edge feature as follows,

$$\mathbf{X}_h = [(W_1'\mathcal{X})[ind_{src}, :]; (W_2'\mathcal{X})[ind_{dst}, :]] \in \mathbb{R}^{E \times 2F'} \quad (12)$$

where $W_1', W_2' \in \mathbb{R}^{T \times F \times F'}$ are learnable parameters of the linear transformation of $\mathcal{X}$, and $W_1'\mathcal{X}, W_2'\mathcal{X} \in \mathbb{R}^{N \times F'}$. $(\cdot)[ind, :]$ is the tensor composed with selected slices of ind_{src} and ind_{dst} which represent the source nodes and the destination nodes of the directed edges in the traffic flow graph. $[\cdot; \cdot]$ is the concatenation operation. $\mathbf{X}_h$ is considered containing the nodes correlation information of the directed edges of traffic flow graph. Then, we further fuse the correlation of nodes by the following convolution,

$$\mathbf{X}_h^d = Conv_{1\times 1}(\mathbf{X}_h) \in \mathbb{R}^{E \times F''} \quad (13)$$

where $Conv_{1\times 1}(\cdot)$ is the convolution operation of with 1×1 convolution kernels, which realizes feature fusion and obtain the updated features of the edges.

From the updated features $\mathbf{X}_h^d$ of the edges, which is also the features of the hyper-nodes in the hypergraph, we adopt the following diffusion HGCN conducted from (3) and (11) to achieve the dynamic edge feature,

$$\begin{aligned} W_{adp} &= diag(L_{adp}) \in \mathbb{R}^{N \times N} \\ W_h &= D_{hv}^{-\frac{1}{2}} H W_{adp} D_{he}^{-\frac{1}{2}} H^\top D_{hv}^{-\frac{1}{2}} \in \mathbb{R}^{E \times E} \\ HGCN_\Theta^d(\mathbf{X}_h^d) &= \sum_{n=0}^{N-1} W_h^n \mathbf{X}_h^d \Theta_n \in \mathbb{R}^E \end{aligned} \quad (14)$$

where $\Theta = [\Theta_0, \ldots, \Theta_{N-1}] \in \mathbb{R}^{N \times F'' \times 1}$ is the parameter to learn, and $L_{adp} \in \mathbb{R}^N$ is the weight vector of the hyper-edge of self-adaptive learning. The output $HGCN_\Theta^d(\mathbf{X}_h^d)$ will be used as the input to the following dynamic GCN of the traffic flow graph.

3) Dynamic Traffic Graph Convolution: We adopt the Gate-TCN [52] in (6) to learn the time correlation of traffic data by the dilated convolution in the time dimension. For the input features sequence $\mathcal{X} \in \mathbb{R}^{N \times F \times T}$ of the nodes of traffic flow graph, Gate-TCN is formulated as follows,

$$\begin{aligned} TCN_\phi(\mathcal{X}) &= Conv_g(\mathcal{X}) \in \mathbb{R}^{N \times F \times (T - m(T_0 - 1))} \\ GateTCN(\mathcal{X}) &= tanh(TCN_{\phi_1}(\mathcal{X})) \odot sigmoid(TCN_{\phi_2}(\mathcal{X})) \end{aligned} \quad (15)$$

where $GateTCN(\mathcal{X}) \in \mathbb{R}^{N \times F \times (T - m(T_0 - 1))}$, $Conv_g(\cdot)$ is the dilated convolution in the time dimension with the convolution kernel $g \in \mathbb{R}^{T_0}$ and the dilation factor m, $TCN_{\phi_1}(\cdot)$ and $TCN_{\phi_2}(\cdot)$ are two models of $TCN_\phi(\cdot)$ with different parameters ϕ_1, ϕ_2 of ϕ. $tanh(\cdot)$ and $sigmoid(\cdot)$ are activation functions.

Algorithm 1 DST-Block of DDSTGCN

```
Require: The node features $\mathcal{X} \in \mathbb{R}^{N \times F_1 \times T_1}$ of the traffic flow graph.
Calculate $\mathcal{X}_h$ by (8);
Calculate $\mathbf{X}^d$ by (10);
Update $\mathbf{X}^d \leftarrow GCN^d(\mathbf{X}^d)$ by (11);
Calculate $\mathbf{X}_h^d$ by (12),(13);
Update $\mathbf{X}_h^d \leftarrow HGCN_\Theta^d(\mathbf{X}_h^d)$ by (14);
Update $\mathcal{X} \leftarrow GateTCN(\mathcal{X})$ by (15);
Update $\mathcal{X} \leftarrow DGCN(\mathcal{X})$ by (16);
Update $\mathcal{X}_h \leftarrow GateTCN(\mathcal{X}_h)$ by (15);
Update $\mathcal{X}_h \leftarrow DHGCN(\mathcal{X}_h)$ by (17);
Calculate $\mathcal{X}'$ by (9);
$\mathcal{X}_{new} = [\mathcal{X}; \mathcal{X}']$;
Output: The new node features $\mathcal{X}_{new} \in \mathbb{R}^{N \times F_2 \times T_2}$ of the traffic flow graph.
```

Then, we propose a Dynamic Graph Convolution Network (DGCN) block to capture the spatial and temporal features of the traffic data as follows, following the idea of Graph WaveNet [52],

$$\begin{aligned} D_f &= Reshape(HGCN_{\Theta_f''}^d(\mathbf{X}_h^d)) \in \mathbb{R}^{N \times N} \\ D_b &= Reshape(HGCN_{\Theta_b''}^d(\mathbf{X}_h^d))^\top \in \mathbb{R}^{N \times N} \\ DGCN(\mathcal{X}) &= \sum_{n=0}^{N-1} (A_f \odot D_f)^n \mathcal{X} {\Theta'}_{n,f} \\ &\quad + (A_b \odot D_b)^n \mathcal{X} {\Theta'}_{n,b} + A_{adp}^n \mathcal{X} {\Theta'}_{n,adp} \end{aligned} \quad (16)$$

where $A_{adp} = Softmax(Relu(E_1 E_2^T)) \in \mathbb{R}^{N \times N}$ is a self-adaptive adjacency matrix in which $E_1, E_2 \in \mathbb{R}^{N \times c}$ are learnable parameters. ${\Theta'}_f = [{\Theta'}_{0,f}, \ldots, {\Theta'}_{N-1,f}] \in \mathbb{R}^{N \times F \times 1}$, ${\Theta'}_b = [{\Theta'}_{0,b}, \ldots, {\Theta'}_{N-1,b}] \in \mathbb{R}^{N \times F \times 1}$ and ${\Theta'}_{adp} = [{\Theta'}_{0,adp}, \ldots, {\Theta'}_{N-1,adp}] \in \mathbb{R}^{N \times F \times 1}$ are learnable parameters. The formula $Reshape(V)$ is the operation of reshaping the vector V into a sparse matrix D, so that the matrix element $D[ind_{src}, ind_{dst}]$ is equal to V.

4) Dynamic Hypergraph Convolution: According to the dual transformation, for the hyper-nodes feature $\mathcal{X}_h$ obtained by (8), we apply the similar Gate-TCN block on it by (15). Then, we give the Dynamic Hypergraph Convolutional Network (DHGCN) to capture the spatial and temporal feature of the hypergraph as follows,

$$\begin{aligned} D_w &= diag(GCN^d(\mathbf{X}^d)) \in \mathbb{R}^{N \times N} \\ W_{hd} &= D_{hv}^{-\frac{1}{2}} H D_w D_{he}^{-\frac{1}{2}} H^\top D_{hv}^{-\frac{1}{2}} \in \mathbb{R}^{E \times E} \\ DHGCN(\mathcal{X}_h) &= \sum_{n=0}^{N-1} W_{hd}^n \mathcal{X}_h {\Theta'}_n \end{aligned} \quad (17)$$

where $\Theta' = [{\Theta'}_0, \ldots, {\Theta'}_{N-1}] \in \mathbb{R}^{N \times F'' \times 1}$ is the parameter to learn. $GCN^d(\mathbf{X}^d)$ provides the dynamic features of the nodes of the traffic flow graph, so that brings the dynamic weights of the hyper-edges of the hypergraph.

Finally, we apply the dual transformation in (9) to the features of the hyper-nodes of hypergraph in (17) and concatenate the results with the node features of the traffic flow graph in (16), and get the output of DST-Block. The above DST-Block can be summarized as Algorithm 1.

B. Input Layer and Output Layer

As shown in Fig.3, the input layer is composed of one Linear Transformation, which mainly transforms the input traffic data to a higher-dimensional space to improve the expressive ability of the network.

The output layer first performs a skip-connection operation on the output of all DST-Blocks, then fuses these spatial-temporal features and feeds them into the layers of Leaky ReLU and Linear Transformations to obtain the final prediction results $\hat{\mathcal{X}}^{(t+1):(t+T)} = \left(\hat{\mathbf{X}}_{t+1}, \hat{\mathbf{X}}_{t+2}, \ldots, \hat{\mathbf{X}}_{t+T}\right)^{\mathsf{T}}$.

In this paper, we use the Mean Absolute Error (MAE) to form the loss function, i.e. for the ground truth $\mathcal{X}^{(t+1):(t+T)} = (\mathbf{X}_{t+1}, \mathbf{X}_{t+2}, \ldots, \mathbf{X}_{t+T})^{\mathsf{T}}$, the loss is defined as follows,

$$\begin{aligned} loss &= MAE(\hat{\mathcal{X}}^{(t+1):(t+T)}, \mathcal{X}^{(t+1):(t+T)}) \\ &= \frac{\sum_{i=1}^{N}\sum_{j=t+1}^{t+T}\left|\hat{x}_j^i - x_j^i\right|}{N \times T} \end{aligned} \tag{18}$$

IV. Experiments

To evaluate the proposed DDSTGCN, we implement traffic prediction experiments on six real traffic datasets compared with state-of-the-art related methods.

A. Experimental Settings

1) Datasets: Six traffic datasets are used in our experiments: METR-LA, PEMS-BAY, PEMS03, PEMS04, PEMS07 and PEMS08, all of which are collected by loop detectors on highways to obtain traffic data on corresponding road segments. METR-LA records traffic speed data of the Los Angeles freeway [55], [73], and PEMS-BAY records traffic speed data of San Francisco Bay Area [55], [74]. PEMS03, PEMS04, PEMS07 and PEMS08 are the traffic flow datasets recorded in four areas of California respectively [67], [74]. The detail of these datasets are shown in Table I. We adopt the same data pre-processing procedures in [55] by taking the sampling rate of one sample every 5 minutes and applying Z-Score normalization. For the METR-LA and PEMS-BAY datasets, the initial traffic flow graph is constructed based on the distance of the road segments [55], while the initial traffic flow graphs of the PEMS03, PEMS04, PEMS07 and PEMS08 datasets are constructed based on the road segment connections [67]. The adjacency matrix of these traffic flow graphs are constructed by the thresholded Gaussian kernel mapping [55]. These datasets are divided into training set, validation set and test set in chronological order with the ratios 7:1:2 for METR-LA and PEMS-BAY, 6:2:2 for PEMS03, PEMS04, PEMS07 and PEMS08.

2) Comparison Methods: We compare the proposed DDSTGCN with a set of traffic prediction methods, including ARIMA (Auto Regressive Integrated Moving Average) [7], VAR (Vector Auto Regression) [11], SVR (Support Vector Regression) [20], LSTM (Long Short-Term Memory) [75], GCRN (Graph Convolutional Recurrent Network) [76], STGCN (Spatio-Temporal Graph Convolutional Network) [51], DCRNN (Diffusion Convolutional Recurrent Neural Network) [55], ASTGCN(r) (Attention based Spatial-Temporal Graph Convolutional Network with recent component) [57], Graph WaveNet [52], DGCN(r) (Dynamic Graph Convolution Network with recent component) [58], STSGCN (Spatial-Temporal Synchronous Graph Convolutional Network) [67], MRA-BGCN (Multi-Range Attentive Bicomponent Graph Convolutional Network) [68] and OGCRNN (Optimized Graph Convolution Recurrent Neural Network) [54]. The performance of all methods is measured by three indicators: Mean Absolute Errors (MAE), Mean Absolute Percentage Errors (MAPE) and Root Mean Squared Errors (RMSE).

TABLE I
THE DATASETS USED IN THE EXPERIMENTS

Dataset	Nodes	Edges	Time Steps	Missingness
METR-LA	207	1515	34272	8.11%
PEMS-BAY	325	2369	52116	0.003%
PEMS03	358	547	26208	0.67%
PEMS04	307	340	16992	3.18%
PEMS07	883	866	28224	0.45%
PEMS08	170	295	17856	0.69%

TABLE II
THE HYPER-PARAMETES USED IN THE EXPERIMENT

Category	Hyper-Parameter	Setup
Train	Batch size	64
	Epochs	200
Optimizer	Initial learning rate	0.001
	Learning rate decay	0.97
	Weight decay	0.0001
	Gradient clipping	3
Model	DST-Blocks b	3
	Intermediate filters F	40
	Diffusion coefficient N	2
	Dilation factor m	1, 2
	Node embedded size c	10
	Dropout p	0.3
	Top-k	4

3) Parameters Setting: Following the previous works [52], [77], the length of the input time series is equal to the length of output time series, i.e. $T' = T = 12$, which means the data in the next hour is predicted by the data in the previous hour. A total of $b = 3$ DST-Blocks are stacked. All adaptive matrices are embedded with uniformly distributed initialization variables in the size of $c = 10$. The dilation factor m of Gate-TCN is alternated between 1 and 2. The diffusion coefficient of all spatial convolutions $N = 2$. Both add dropout with $p = 0.3$. The number of intermediate filters is $F = 40$. The value of top-k sampling is $k = 4$. The model uses a single Nvidia GeForce RTX 3090 GPU and adopts Adam optimizer to train the model. The batch size is 64, the initial learning rate is 0.001, and the learning rate decay per epoch is 0.97. All experiments are repeated ten times. More details of the experimental setup can be found in Table II and the source code.

TABLE III
THE TRAFFIC SPEED PREDICTION RESULTS OF DIFFERENT METHODS ON METR-LA AND PEMS-BAY

Dataset	Models	15 min			30 min			60 min		
		MAE	RMSE	MAPE(%)	MAE	RMSE	MAPE(%)	MAE	RMSE	MAPE(%)
METR-LA	ARIMA	3.99	8.21	9.60	5.15	10.45	12.70	6.90	13.23	17.40
	VAR	4.22	7.89	10.20	5.41	9.13	12.70	6.52	10.11	15.80
	SVR	3.99	7.94	9.90	4.23	8.17	12.90	4.49	8.69	14.00
	LSTM	3.44	6.30	9.60	3.77	7.23	10.90	4.37	8.69	13.20
	GCRN	3.03	5.75	8.26	3.54	6.92	10.11	4.32	8.48	13.05
	STGCN	2.88	5.74	7.62	3.47	7.24	9.57	4.59	9.40	12.70
	DCRNN	2.77	5.38	7.30	3.15	6.45	8.80	3.60	7.60	10.50
	ASTGCN(r)	2.98	5.60	7.93	3.36	6.54	9.24	3.83	7.75	10.78
	OGCRNN	3.04	5.72	8.20	3.45	6.66	9.56	3.98	7.80	11.35
	STSGCN	3.43	6.57	9.73	3.60	6.96	10.35	3.95	7.77	11.65
	DGCN(r)	2.90	5.48	7.60	3.23	6.41	8.79	3.63	7.37	10.31
	Graph WaveNet	2.69	5.15	6.90	3.07	6.22	8.37	3.53	7.37	10.01
	MRA-BGCN	2.67	5.12	6.80	3.06	6.17	8.30	3.49	7.30	10.00
	DDSTGCN(ours)	**2.64**	**5.01**	**6.76**	**3.00**	**6.02**	**8.12**	**3.44**	**7.13**	**9.74**
PEMS-BAY	ARIMA	1.62	3.30	3.50	2.33	4.76	5.40	3.38	6.50	8.30
	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
	LSTM	2.05	4.19	4.80	2.20	4.55	5.20	2.37	4.96	5.57
	GCRN	1.46	3.06	3.22	1.88	4.17	4.34	2.40	5.36	5.89
	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
	ASTGCN(r)	1.48	3.02	3.27	1.80	3.88	4.17	2.15	4.81	5.11
	OGCRNN	1.48	3.06	3.28	1.83	3.96	4.26	2.16	4.77	5.12
	STSGCN	2.54	4.47	5.88	2.60	4.93	6.03	2.71	5.28	6.39
	DGCN(r)	1.43	2.93	3.10	1.71	3.85	3.97	2.00	4.57	4.70
	Graph WaveNet	1.30	2.74	2.73	1.63	3.70	3.67	1.95	4.52	4.63
	MRA-BGCN	**1.29**	2.72	2.90	1.61	3.67	3.80	1.91	4.46	4.60
	DDSTGCN(ours)	**1.29**	**2.71**	**2.69**	**1.60**	**3.64**	**3.62**	**1.89**	**4.37**	**4.46**

B. Experimental Results

1) Traffic Prediction Results: We perform traffic prediction experiments on two different types of datasets of traffic speed and traffic flow. In order to facilitate the performance comparison with other methods, we adopt the same predictive evaluation settings as that of [55] and [67], i.e. for the traffic speed datasets of METR-LA and PEMS-BAY, 15 minutes', 30 minutes' and 60 minutes' prediction results are reported, and for the traffic flow datasets of PEMS03, PEMS04, PEMS07 and PEMS08, the average traffic flow within an hour is predicted. Table III shows the results of different methods for the traffic speed prediction. Table IV shows the results of different methods for predicting the traffic flow. The best results are highlighted in bold font, and the second results are highlighted with underlines. It is shown that the proposed DDSTGCN method outperforms the other algorithms in all indicators in all cases. Compared with the second results, DDSTGCN gains average improvements of 1.16%, 1.69%, and 1.87% under MAE, RMSE and MAPE metrics on the traffic speed datasets of METR-LA and PEMS-BAY with the three prediction intervals. For the four traffic flow datasets of PEMS03, PEMS04, PEMS07 and PEMS08, the improvements are more significant. There are average increments of 9.78%, 7.80%, 5.96% under MAE, RMSE and MAPE metrics compared with the second results.

From the results in Table III and Table IV, it is observed that most methods usually perform well on certain types of traffic datasets. For example, MRA-BGCN [68] and Graph WaveNet [52] have excellent performances on the two speed datasets, but have significantly decreased performances on the four flow datasets. In contrast, DGCN [58] and STSGCN [67] perform well on the four flow datasets, but have poor performances on the two speed datasets. However, our DDSTGCN model achieves the best results whether on the traffic speed datasets or the traffic flow datasets. This excellent generalization of our DDSTGCN model is explained that the dual dynamic spatial-temporal GCNs can capture more complex spatial-temporal properties of traffic data. Additionally, from the results in Table IV, it is indicated that our DDSTGCN model has good scalability for large dataset. In the experiment, PEMS07 is a large-scale traffic dataset with 883 nodes which is larger than other datasets. DGCN [58] and MRA-BGCN [68] models perform very well on the other datasets but fail on PEMS07. These poor scalabilities result from the models' over complex structure with high memory demanding. For the large dataset, our DDSTGCN model still gets the best results with good scalability.

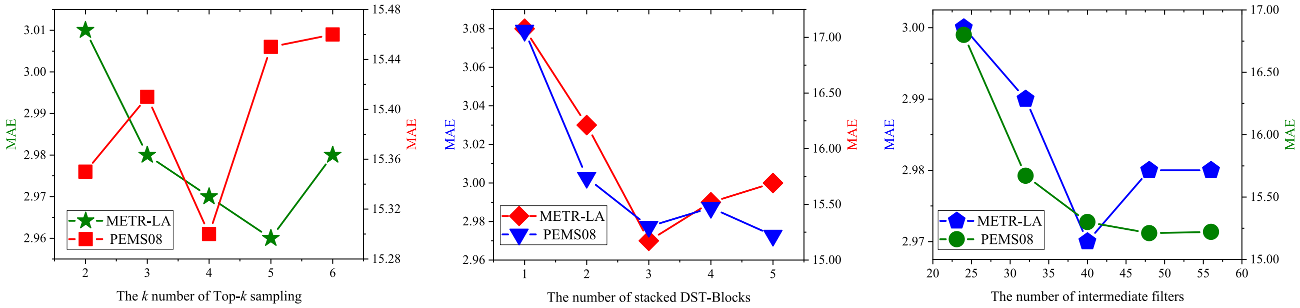
2) Ablation Experiments: To further verify the effectiveness of different parts of DDSTGCN, we implement ablation experiments by combing the components of the proposed model with different forms, including dynamic GCNs on the traffic flow graph and its dual hypergraph, and the DIM module, as shown in Fig.3. The results of one hour traffic prediction on METR-LA and PEMS08 are shown in Table V. It can be concluded that the proposed DDSTGCN model has the best performance compared with the models removing certain components. Concretely, the model that only uses GCN on the traffic flow graph almost achieves the same performance

TABLE IV

THE TRAFFIC FLOW PREDICTION RESULTS OF DIFFERENT METHODS ON PEMS03, PEMS04, PEMS07 AND PEMS08

Models / Dataset	PEMS03			PEMS04		
Metrics	MAE	RMSE	MAPE(%)	MAE	RMSE	MAPE(%)
ARIMA	35.41	47.59	33.78	33.73	48.80	24.18
VAR	23.65	38.26	24.51	23.75	36.66	18.09
SVR	21.97	35.29	21.51	28.70	44.56	19.20
LSTM	21.33±0.24	35.11±0.50	23.33±4.23	27.14±0.20	41.59±0.21	18.20±0.40
GCRN	19.88±0.04	33.20±0.12	19.71±0.38	26.73±0.06	41.56±0.12	19.20±0.77
STGCN	17.49±0.46	30.12±0.70	17.15±0.45	22.70±0.64	35.55±0.75	14.59±0.21
DCRNN	18.18±0.15	30.31±0.25	18.91±0.82	24.70±0.22	38.12±0.26	17.12±0.37
ASTGCN(r)	17.69±1.43	29.66±1.68	19.40±2.24	22.93±1.29	35.22±1.90	16.56±1.36
OGCRNN	17.16±0.27	29.40±0.34	16.48±0.63	22.16±0.20	35.15±0.42	15.72±0.70
STSGCN	17.48±0.15	29.21±0.56	16.78±0.20	21.19±0.10	33.65±0.20	13.90±0.05
DGCN(r)	16.36±0.28	26.11±0.41	17.29±0.67	21.11±0.06	32.46±0.18	14.46±0.13
Graph WaveNet	19.85±0.03	32.94±0.18	19.31±0.49	25.45±0.03	37.70±0.04	17.29±0.24
MRA-BGCN	20.40±0.25	32.94±0.43	24.62±0.95	23.38±0.37	36.02±0.63	16.43±0.23
DDSTGCN(ours)	**14.77±0.02**	**24.57±0.43**	**14.76±0.05**	**19.61±0.11**	**31.05±0.05**	**13.70±0.03**
Models Dataset	PEMS07			PEMS08		
ARIMA	38.17	59.27	19.46	31.09	44.32	22.73
VAR	75.63	115.24	32.22	23.46	36.33	15.42
SVR	32.49	50.22	14.26	23.25	36.16	14.64
LSTM	29.98±0.42	45.84±0.57	13.20±0.53	22.20±0.18	34.06±0.32	14.20±0.59
GCRN	31.03±0.40	48.70±0.83	15.67±0.81	21.28±0.38	33.46±0.67	14.15±1.15
STGCN	25.38±0.49	38.78±0.58	11.08±0.18	18.02±0.14	27.83±0.20	11.40±0.10
DCRNN	25.30±0.52	38.58±0.70	11.66±0.33	17.86±0.03	27.83±0.05	11.45±0.03
ASTGCN(r)	28.05±2.34	42.57±3.31	13.92±1.65	18.61±0.40	28.16±0.48	13.08±1.00
OGCRNN	24.45±0.18	39.80±0.29	10.22±0.12	18.50±0.23	29.00±0.25	11.74±0.33
STSGCN	24.26±0.14	39.03±0.27	10.21±0.05	17.13±0.09	26.80±0.18	10.96±0.07
DGCN(r)	-	-	-	17.19±0.21	26.56±0.26	11.93±0.50
Graph WaveNet	26.85±0.05	42.78±0.07	12.12±0.41	19.13±0.08	31.05±0.07	12.68±0.57
MRA-BGCN	-	-	-	18.97±0.18	28.69±0.25	13.04±0.27
DDSTGCN(ours)	**21.44±0.03**	**34.38±0.12**	**9.39±0.09**	**15.30±0.12**	**24.16±0.03**	**10.53±0.17**

'-' indicates the method is failure because of out-of-memory.



(a) The Top-k edges of the traffic flow graph sampled for dual transformation.

(b) The number of b DST-Blocks stacking.

(c) The number of intermediate filters F.

Fig. 4. Hyper-parameter study on METR-LA and PEMS08.

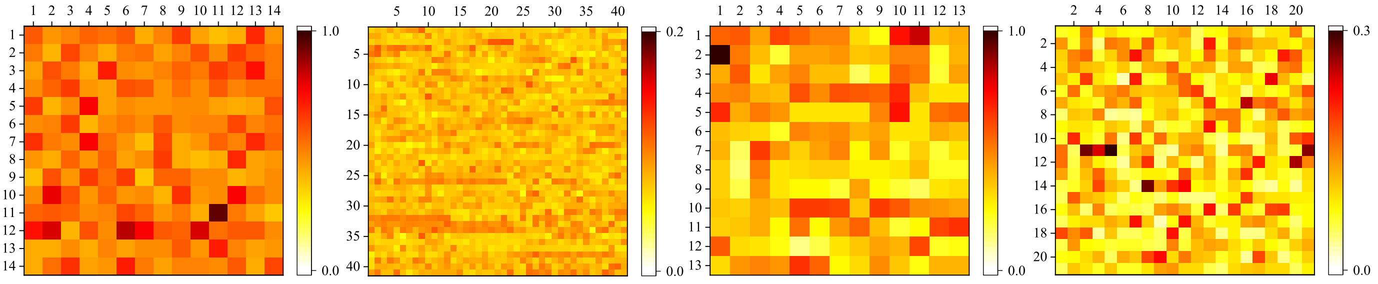
as that only uses GCN on the hypergraph, but there will be an improvement if both GCNs are used on the traffic flow graph and its hypergraph, which shows that the dual structure we proposed can better capture the intrinsic spatial-temporal feature of traffic data. On the basis of similar structure, the model with dynamic mechanism added to GCNs of graph or hypergraph has better performance than the model without it, which means the proposed dynamic mechanism will bring gain for our model by modeling the dynamic characteristics of the traffic data. Moreover, the DIM module also plays a positive role as the model with DIM performs better than that without it.

3) Parameter Sensitivity: The main hyper-parameters of the proposed DDSTGCN model include the Top-k edges of the graph sampled for dual transformation, the number of b DST-Blocks Stacking and the number of intermediate filters F. We analyze their sensitivity by traffic prediction experiments on METR-LA and PEMS08 with different settings. Here, MAE is used as the evaluation metric. The detailed results are shown in Fig.4.

- Top-k edges of the traffic flow graph sampled for dual transformation: k determines the number of hyper-nodes in hypergraph. As shown in Fig.4 (a), with k increasing, the dual hypergraph will be more complex, which will improve the expressiveness of the hypergraph and thus bring performance gain to the model. But further increases will lead to performance degradation as too much noise will be introduced into the model, which will mislead the model

TABLE V
ABLATION EXPERIMENT RESULTS ON METR-LA AND PEMS08

Selected components					Dataset					
Traffic graph		DIM	Hypergraph		METR-LA			PEMS08		
GCN	DGCN		HGCN	DHGCN	MAE	RMSE	MAPE(%)	MAE	RMSE	MAPE(%)
✓					3.08	6.09	8.71	16.09	25.16	15.16
			✓		3.04	6.09	8.37	17.40	26.77	11.19
	✓				3.06	6.10	8.37	15.75	24.77	11.97
				✓	3.02	6.05	8.26	16.59	25.58	10.92
✓			✓		3.00	5.99	8.12	15.60	24.45	10.73
✓		✓		✓	2.98	5.94	8.07	15.63	24.52	10.73
	✓	✓	✓		2.98	5.96	8.06	15.39	24.24	10.37
	✓	✓		✓	**2.97**	**5.90**	**8.01**	**15.30**	**24.15**	**10.36**



(a) $GCN^d(\mathbf{X}^d)$ of METR-LA. (b) $HGCN^d_\Theta(\mathbf{X}^d_h)$ of METR-LA. (c) $GCN^d(\mathbf{X}^d)$ of PEMS08. (d) $HGCN^d_\Theta(\mathbf{X}^d_h)$ of PEMS08.

Fig. 5. Dynamic mechanism visualization. (a) and (c) are the standard deviation visualization matrices of the dynamic value $HGCN^d_\Theta(\mathbf{X}^d_h)$ of DGCN, (b) and (d) are the standard deviation visualization matrices of the dynamic value $GCN^d(\mathbf{X}^d)$ of DHGCN.

TABLE VI
THE COMPUTATION COST OF DDSTGCN COMPARED WITH OTHER GCN BASED METHODS

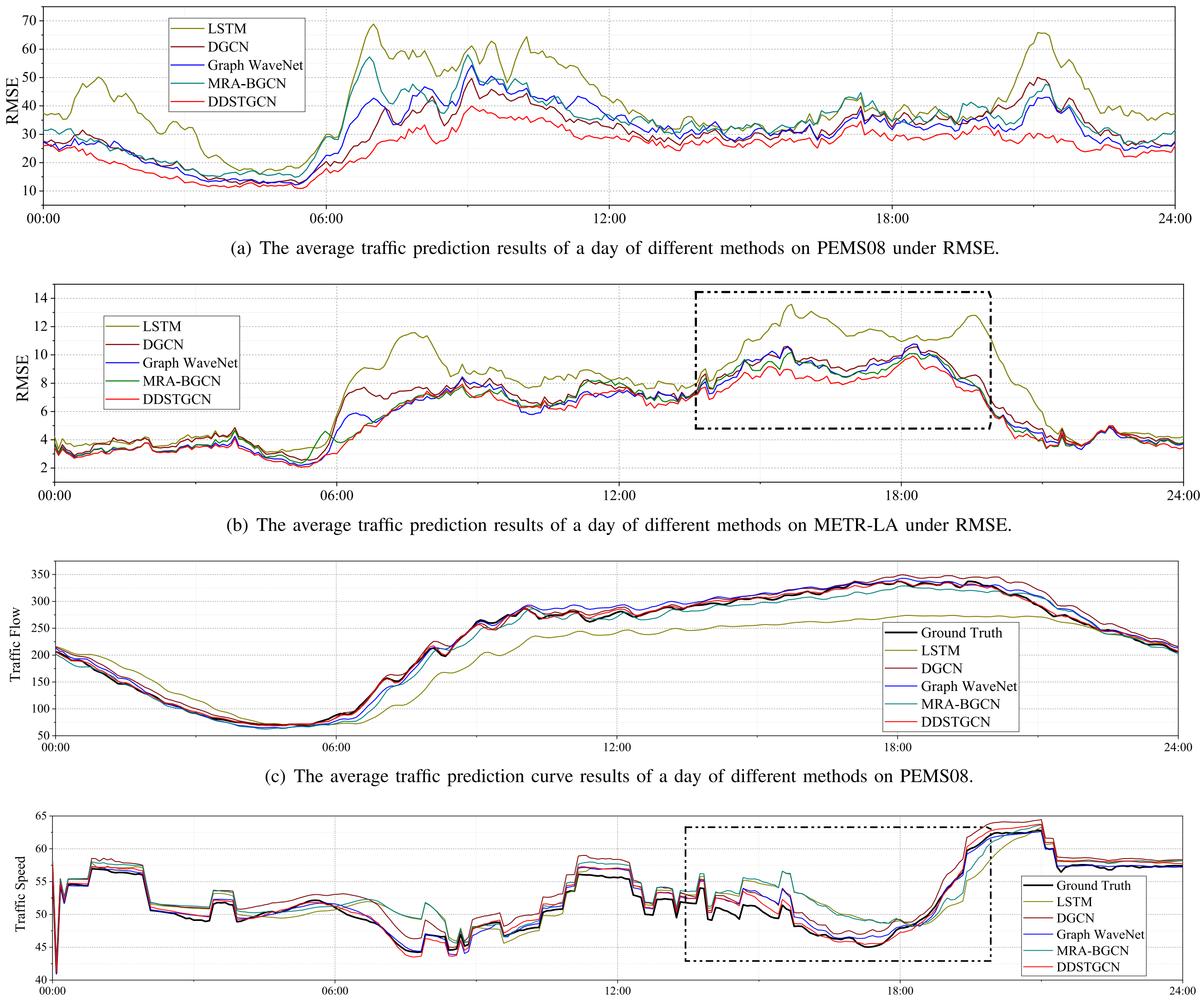
Datasets	Computation Time (Training(*sec/epoch*) / Inference(*sec*))				
	Graph WaveNet	DGCN	MRA-BGCN	STSGCN	DDSTGCN
METR-LA	46.84 / 2.13	85.54 / 5.08	1206.49 / 29.91	297.10 / 4.62	80.15 / 3.50
PEMS-BAY	139.46 / 6.27	251.57 / 13.54	3238.24 / 87.42	1006.54 / 11.52	227.38 / 10.70
PEMS03	36.82 / 3.85	115.91 / 16.73	496.97 / 24.87	319.83 / 7.93	56.72 / 5.94
PEMS04	32.18 / 3.45	62.02 / 8.17	211.62 / 15.86	181.90 / 4.64	28.06 / 2.94
PEMS07	124.08 / 12.83	- / -	- / -	975.04 / 30.20	201.82 / 27.55
PEMS08	24.61 / 2.30	30.27 / 4.35	155.77 / 17.41	109.36 / 4.77	17.82 / 1.58

'-' indicates the method fails because of out-of-memory.

bringing too much computational burden. Thus, we set $k = 4$ for our model.

- The number of b DST-Blocks stacking: As shown in Fig.4 (b), the number of DST-Blocks will increase the depth of the network and so enhance the expression ability of the model to improve performance, but with the number increasing, the performance will gradually stabilize. From the results, we stack $b = 3$ DST-Blocks in our model.
- The number of intermediate filters F: The number of intermediate filters is the dimension of the feature channels of nodes or hyper-nodes of graph and hypergraph. As shown in Fig.4 (c), the performance first declines and then stabilizes with the number increasing, but large number will make the model too complicated, leading to overfitting and degrading performance. Thus, we set the value of $F = 40$ in our model.

4) Visualization of Experiment Results: To intuitively demonstrate the experiment results, we visualize the average traffic prediction results of a day of different methods on METR-LA and PEMS08, and the average RMSE and the prediction results compared with ground truth are shown in Fig.6. It is indicated that our DDSTGCN method has better performance at all times compared with other methods. In Fig.6 (a), our DDSTGCN method (red curve) is always clearly below the other curves, which indicates that DDSTGCN is significantly superior to other methods on the PEMS08 dataset. However, on the METR-LA dataset, the advantages of DDSTGCN are not so obvious. As shown in Fig.6 (b) and (d), they leads other methods with weak gap at most of time. It can be explained that the METR-LA dataset is a relatively simple dataset with less number of nodes and fewer various traffic data, compared with the PEMS08 dataset. So many



(a) The average traffic prediction results of a day of different methods on PEMS08 under RMSE.

(b) The average traffic prediction results of a day of different methods on METR-LA under RMSE.

(c) The average traffic prediction curve results of a day of different methods on PEMS08.

(d) The average traffic prediction curve results of a day of different methods on METR-LA.

Fig. 6. The average traffic prediction results of a day of different methods on METR-LA and PEMS08.

methods perform well on the METR-LA dataset. However, in the period of evening rush hour, indicated by the black dashed box in Fig.6 (b) and (d), the proposed DDSTGCN method performs significantly better than all other methods, which means that DDSTGCN is good at dealing with complex and changeable traffic data. This also verifies our thought that modeling the dynamic spatial and temporal features of the nodes and edges of traffic flow graph at the same time can reveal the intrinsic property of the transportation system and so enhance the robustness of the model under complex traffic conditions.

To further show the advantages of the dynamic mechanism of the edges features, we visualize the standard deviation of the dynamic values of the edge in DGCN and the hyper-edge in DHGCN at different prediction moments of our model on METR-LA and PEMS08. To facilitate visualization, we arrange the weight standard deviation values of different edges into a square matrix, as shown in Fig.5. It is shown that the dynamic edge values of the traffic flow graph which can be calculated by $HGCN^d_\Theta(\mathbf{X}^d_h)$ in (14) and the dynamic hyper-edge values of the hypergraph which can be calculated by $GCN^d(\mathbf{X}^d)$ in (11) have large scale of variation range, where darker pixels represent larger standard deviation values. Additionally, different edges and hyper-edges have different standard deviations, which means during dynamic modeling, some areas are prone to severe traffic fluctuations and have large standard deviations, while the other areas have relatively stable traffic conditions and so small standard deviations. This point further verifies that our dynamic mechanism of the edges features is effective and necessary for traffic prediction under complex traffic conditions.

5) Computation Time: In Table VI, we report the computation efficiency of DDSTGCN and other GCN-based methods. It is indicated that our model has acceptable training and inference time on most datasets. Although our method has more training and inference time than Graph WaveNet, our method has less training and inference time than the other three methods of DGCN, MRA-BGCN and STSGCN, which are considered as the representative traffic prediction methods with good performance. Considered the obvious performance improvement of the proposed DDSTGCN method, the computation time of DDSTGCN is acceptable. Even in real-time applications, the difference in the inference time of these methods is almost imperceptible.

V. Conclusion

In this paper, we propose a novel dual dynamic GCN traffic prediction method. Different from the current methods that only consider the changes and correlations of the nodes of traffic flow graph, our method transforms the edges of traffic flow graph to its dual hypergraph and exploits their spatial and temporal property by GCN on hypergraph. The proposed method is evaluated on six real-world traffic datasets and proved to have state-of-the-art performance. However, considering the high complexity of the proposed model, exploring an efficient sampling method for the edges of traffic flow graph and constructing a more flexible dual dynamic GCN traffic prediction method is worth studying in future work.

References

[1] E. I. Vlahogianni, M. G. Karlaftis, and J. C. Golias, "Short-term traffic forecasting: Where we are and where we're going," *Transp. Res. C, Emerg. Technol.*, vol. 43, pp. 3–19, Jun. 2014.

[2] J. Liu and W. Guan, "A summary of traffic flow forecasting methods," *J. Highway Transp. Res. Develop.*, vol. 21, no. 3, pp. 82–85, Mar. 2004.

[3] P. Cai, Y. Wang, G. Lu, P. Chen, C. Ding, and J. Sun, "A spatiotemporal correlative k-nearest neighbor model for short-term traffic multistep forecasting," *Transp. Res. C, Emerg. Technol.*, vol. 62, pp. 21–34, Jan. 2016.

[4] C. Antoniou, H. N. Koutsopoulos, and G. Yannis, "Dynamic data-driven local traffic state estimation and prediction," *Transp. Res. C, Emerg. Technol.*, vol. 34, pp. 89–107, Sep. 2013.

[5] G. E. P. Box and D. A. Pierce, "Distribution of residual autocorrelations in autoregressive-integrated moving average time series models," *J. Amer. Statist. Assoc.*, vol. 65, no. 332, pp. 1509–1526, Apr. 1970.

[6] M. S. Ahmed and A. R. Cook, "Analysis of freeway traffic time-series data by using Box–Jenkins techniques," *Transp. Res. Rec.*, vol. 722, no. 722, pp. 1–9, 1979.

[7] S. Makridakis and M. Hibon, "ARMA models and the box-jenkins methodology," *J. Forecasting*, vol. 16, no. 3, pp. 147–163, May 1997.

[8] M. Van Der Voort, M. Dougherty, and S. Watson, "Combining Kohonen maps with ARIMA time series models to forecast traffic flow," *Transp. Res. C, Emerg. Technol.*, vol. 4, no. 5, pp. 307–318, Oct. 1996.

[9] B. M. Williams and L. A. Hoel, "Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: Theoretical basis and empirical results," *J. Transp. Eng.*, vol. 129, no. 6, pp. 664–672, Nov. 2003.

[10] B. L. Smith, B. M. Williams, and R. K. Oswald, "Comparison of parametric and nonparametric models for traffic flow forecasting," *Transp. Res. C, Emerg. Technol.*, vol. 10, no. 4, pp. 303–321, Aug. 2002.

[11] D. Qin, "Rise of VAR modelling approach," *J. Econ. Surv.*, vol. 25, no. 1, pp. 156–174, 2011.

[12] Y. Xie, Y. Zhang, and Z. Ye, "Short-term traffic volume forecasting using Kalman filter with discrete wavelet decomposition," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 22, no. 5, pp. 326–334, 2007.

[13] Y. Wang and M. Papageorgiou, "Real-time freeway traffic state estimation based on extended Kalman filter: A general approach," *Transp. Res. B, Methodol.*, vol. 39, no. 2, pp. 141–167, 2005.

[14] J. Guo, W. Huang, and B. M. Williams, "Adaptive Kalman filter approach for stochastic short-term traffic flow rate prediction and uncertainty quantification," *Transp. Res. C, Emerg. Technol.*, vol. 43, pp. 50–64, Jun. 2014.

[15] Y. Wang, M. Papageorgiou, and A. Messmer, "RENAISSANCE—A unified macroscopic model-based approach to real-time freeway network traffic surveillance," *Transp. Res. C, Emerg. Technol.*, vol. 14, no. 3, pp. 190–212, Jun. 2006.

[16] C. M. J. Tampere and L. H. Immers, "An extended Kalman filter application for traffic state estimation using CTM with implicit mode switching and dynamic parameters," in *Proc. IEEE Intell. Transp. Syst. Conf.*, Sep. 2007, pp. 209–216.

[17] C. P. Van Hinsbergen, T. Schreiter, F. S. Zuurbier, J. Van Lint, and H. J. Van Zuylen, "Localized extended Kalman filter for scalable real-time traffic state estimation," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 1, pp. 385–394, Mar. 2011.

[18] A. Nantes, D. Ngoduy, A. Bhaskar, M. Miska, and E. Chung, "Real-time traffic state estimation in urban corridors from heterogeneous data," *Transp. Res. C, Emerg. Technol.*, vol. 66, pp. 99–118, May 2016.

[19] Y. Yuan, J. W. C. van Lint, R. E. Wilson, F. van Wageningen-Kessels, and S. P. Hoogendoorn, "Real-time Lagrangian traffic state estimator for freeways," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 1, pp. 59–70, Mar. 2012.

[20] H. Drucker *et al.*, "Support vector regression machines," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 9, 1997, pp. 155–161.

[21] C.-H. Wu, C.-C. Wei, and D.-C. Su, "Travel-time prediction with support vector regression," in *Proc. IEEE Int. Conf. Intell. Transp. Syst.*, Dec. 2004, vol. 5, no. 4, pp. 276–281.

[22] R. Chen, C.-Y. Liang, W.-C. Hong, and D.-X. Gu, "Forecasting holiday daily tourist flow based on seasonal support vector regression with adaptive genetic algorithm," *Appl. Soft Comput.*, vol. 26, pp. 435–443, Jan. 2015.

[23] N. Mitrovic, M. T. Asif, J. Dauwels, and P. Jaillet, "Low-dimensional models for compressed sensing and prediction of large-scale traffic data," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 5, pp. 2949–2954, Oct. 2015.

[24] H. Drucker, C. J. Burges, L. Kaufman, A. Smola, and V. Vapnik, "Support vector regression machines," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 9, 1996, pp. 1–7.

[25] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining Knowl. Discovery*, vol. 2, no. 2, pp. 121–167, Jan. 1998.

[26] G. Leshem and Y. Ritov, "Traffic flow prediction using AdaBoost algorithm with random forests as a weak learner," in *Proc. World Acad. Sci., Eng. Technol.*, vol. 19, pp. 193–198, Jan. 2007.

[27] U. Johansson, H. Boström, T. Löfström, and H. Linusson, "Regression conformal prediction with random forests," *Mach. Learn.*, vol. 97, nos. 1–2, pp. 155–176, 2014.

[28] S. Yang and S. Qian, "Understanding and predicting travel time with spatio-temporal features of network traffic flow, weather and incidents," *IEEE Intell. Transp. Syst. Mag.*, vol. 11, no. 3, pp. 12–28, Jun. 2019.

[29] H. Xiao, H. Sun, B. Ran, and Y. Oh, "Fuzzy-neural network traffic prediction framework with wavelet decomposition," *Transp. Res. Rec.*, vol. 1836, no. 1, pp. 16–20, 2003.

[30] H. Kanoh, T. Furukawa, S. Tsukahara, K. Hara, H. Nishi, and H. Kurokawa, "Short-term traffic prediction using fuzzy C-means and cellular automata in a wide-area road network," in *Proc. IEEE Intell. Transp. Syst.*, Sep. 2005, pp. 381–385.

[31] H. Wang, L. Liu, S. Dong, Z. Qian, and H. Wei, "A novel work zone short-term vehicle-type specific traffic speed prediction model through the hybrid EMD–ARIMA framework," *Transportmetrica B, Transp. Dyn.*, vol. 4, no. 3, pp. 159–186, 2016.

[32] A. Hofleitner, R. Herring, and A. Bayen, "Arterial travel time forecast with streaming data: A hybrid approach of flow modeling and machine learning," *Transp. Res. B, Methodol.*, vol. 46, no. 9, pp. 1097–1122, 2012.

[33] E. S. Yu and C. Y. R. Chen, "Traffic prediction using neural networks," in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*, Nov. 1993, pp. 991–995.

[34] L. Florio and L. Mussone, "Neural-network models for classification and forecasting of freeway traffic flow stability," *Control Eng. Pract.*, vol. 4, no. 2, pp. 153–164, 1996.

[35] J. Van Lint, S. Hoogendoorn, and H. J. van Zuylen, "Accurate freeway travel time prediction with state-space neural networks under missing data," *Transp. Res. C, Emerg. Technol.*, vol. 13, nos. 5–6, pp. 347–369, 2005.

[36] G. Fusco, C. Colombaroni, and N. Isaenko, "Comparative analysis of implicit models for real-time short-term traffic predictions," *IET Intell. Transp. Syst.*, vol. 10, no. 4, pp. 270–278, 2016.

[37] C. Zhou and P. C. Nelson, "Predicting traffic congestion using recurrent neural networks," in *Proc. World Congr. Intell. Transp. Syst.*, 2002.

[38] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015.

[39] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[40] Z. Zhao *et al.*, “LSTM network: A deep learning approach for short-term traffic forecast,” *IET Intell. Transp. Syst.*, vol. 11, no. 2, pp. 68–75, 2017.

[41] J. Zhang, Y. Zheng, and D. Qi, “Deep spatio-temporal residual networks for citywide crowd flows prediction,” in *Proc. AAAI Conf. Artif. Intell.*, 2017, pp. 1655–1661.

[42] Y. Lv, Y. Duan, W. Kang, Z. Li, and F.-Y. Wang, “Traffic flow prediction with big data: A deep learning approach,” *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 865–873, Apr. 2014.

[43] H. Yu, Z. Wu, S. Wang, Y. Wang, and X. Ma, “Spatiotemporal recurrent convolutional networks for traffic prediction in transportation networks,” *Sensors*, vol. 17, no. 7, p. 1501, Jun. 2017.

[44] K. Cho *et al.*, “Learning phrase representations using RNN encoder–decoder for statistical machine translation,” 2014, *arXiv:1406.1078*.

[45] R. Fu, Z. Zhang, and L. Li, “Using LSTM and GRU neural network methods for traffic flow prediction,” in *Proc. 31st Youth Academic Annu. Conf. Chin. Assoc. Autom. (YAC)*, Nov. 2016, pp. 324–328.

[46] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, vol. 25, Dec. 2012, pp. 1097–1105.

[47] J. Bruna, W. Zaremba, A. Szalm, and Y. LeCun, “Spectral networks and deep locally connected networks on graphs,” in *Proc. Int. Conf. Learn. Represent.*, 2014, pp. 1–14.

[48] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” in *Proc. Int. Conf. Learn. Represent.*, 2017, pp. 1–14.

[49] S. Bai, J. Zico Kolter, and V. Koltun, “An empirical evaluation of generic convolutional and recurrent networks for sequence modeling,” 2018, *arXiv:1803.01271*.

[50] C. Lea, M. D. Flynn, R. Vidal, A. Reiter, and G. D. Hager, “Temporal convolutional networks for action segmentation and detection,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 156–165.

[51] B. Yu, H. Yin, and Z. Zhu, “Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting,” in *Proc. 27th Int. Joint Conf. Artif. Intell.*, Jul. 2018, pp. 1–7.

[52] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, “Graph WaveNet for deep spatial–temporal graph modeling,” in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 1907–1913.

[53] Z. Diao, G. Wang, D. Zhang, Y. Liu, K. Xie, and S. He, “Dynamic spatial–temporal graph convolutional neural networks for traffic forecasting,” in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, Jul. 2019, pp. 890–897.

[54] K. Guo *et al.*, “Optimized graph convolution recurrent neural network for traffic prediction,” *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 2, pp. 1138–1149, Feb. 2021.

[55] Y. Li, R. Yu, C. Shahabi, and Y. Liu, “Diffusion convolutional recurrent neural network: Data-driven traffic forecasting,” in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–16.

[56] X. Geng *et al.*, “Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting,” in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, no. 1, 2019, pp. 3656–3663.

[57] 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.*, 2019, pp. 922–929.

[58] K. Guo, Y. Hu, Z. Qian, Y. Sun, J. Gao, and B. Yin, “Dynamic graph convolution network for traffic forecasting based on latent network of Laplace matrix estimation,” *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 2, pp. 1–10, Feb. 2020.

[59] J. Li, Y. Liu, and L. Zou, “DynGCN: A dynamic graph convolutional network based on spatial–temporal modeling,” in *Proc. Int. Conf. Web Inf. Syst. Eng.* Amsterdam, The Netherlands: Springer, 2020, pp. 83–95.

[60] F. Li, J. Feng, H. Yan, G. Jin, D. Jin, and Y. Li, “Dynamic graph convolutional recurrent network for traffic prediction: Benchmark and solution,” 2021, *arXiv:2104.14917*.

[61] J. Barros, M. Araujo, and R. J. Rossetti, “Short-term real-time traffic prediction methods: A survey,” in *Proc. Int. Conf. Models Technol. Intell. Transp. Syst.*, 2015, pp. 132–139.

[62] S. George and A. K. Santra, “Traffic prediction using multifaceted techniques: A survey,” *Wirelesss Pers. Commun.*, vol. 115, no. 2, pp. 1047–1106, Nov. 2020.

[63] J. Ye, J. Zhao, K. Ye, and C. Xu, “How to build a graph-based deep learning architecture in traffic domain: A survey,” *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 5, pp. 3904–3924, May 2020.

[64] X. Yin, G. Wu, J. Wei, Y. Shen, H. Qi, and B. Yin, “Deep learning on traffic prediction: Methods, analysis, and future directions,” *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 6, pp. 4927–4943, Jun. 2021.

[65] W. Jiang and J. Luo, “Graph neural network for traffic forecasting: A survey,” 2021, *arXiv:2101.11174*.

[66] H. Yuan and G. Li, “A survey of traffic prediction: From spatio-temporal data to intelligent transportation,” *Data Sci. Eng.*, vol. 6, no. 1, pp. 63–85, Mar. 2021.

[67] C. Song, Y. Lin, S. Guo, and H. Wan, “Spatial-temporal synchronous graph convolutional networks: A new framework for spatial–temporal network data forecasting,” in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, no. 1, pp. 914–921.

[68] W. Chen, L. Chen, Y. Xie, W. Cao, and X. Feng, “Multi-range attentive bicomponent graph convolutional network for traffic forecasting,” in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, no. 4, pp. 3529–3536.

[69] M. Defferrard, X. Bresson, and P. Vandergheynst, “Convolution neural networks on graphs with fast localized spectral filtering,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 1–9.

[70] N. Yadati, M. Nimishakavi, P. Yadav, V. Nitin, A. Louis, and P. Talukdar, “HyperGCN: A new method for training graph convolutional networks on hypergraphs,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 1511–1522.

[71] Y. Feng, H. You, Z. Zhang, R. Ji, and Y. Gao, “Hypergraph neural networks,” in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 3558–3565.

[72] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier, “Language modeling with gated convolutional networks,” in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 933–941.

[73] H. V. Jagadish *et al.*, “Big data and its technical challenges,” *Commun. ACM*, vol. 57, no. 7, pp. 86–94, Jul. 2014.

[74] C. Chen, K. Petty, A. Skabardonis, P. Varaiya, and Z. Jia, “Freeway performance measurement system: Mining loop detector data,” *Transp. Res. Rec., J. Transp. Res. Board*, vol. 1748, no. 1, pp. 96–102, Jan. 2001.

[75] I. Sutskever, O. Vinyals, and Q. V. Le, “Sequence to sequence learning with neural networks,” in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, vol. 27, Montreal, QC, Canada, Dec. 2014, pp. 3104–3112.

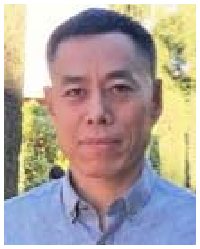
[76] Y. Seo, M. Defferrard, P. Vandergheynst, and X. Bresson, “Structured sequence modeling with graph convolutional recurrent networks,” in *Proc. Int. Conf. Neural Inf. Process.* Siem Reap, Cambodia: Springer, 2018, pp. 362–373.

[77] S. Shleifer, C. McCreery, and V. Chitters, “Incrementally improving graph WaveNet performance on traffic prediction,” 2019, *arXiv:1912.07390*.

Yanfeng Sun (Member, IEEE) received the Ph.D. degree from the Dalian University of Technology in 1993. She is currently a Professor with the Faculty of Information Technology, Beijing University of Technology, Beijing, China. She is also a Researcher with the Beijing Key Laboratory of Multimedia and Intelligent Software Technology, and the Beijing Artificial Intelligence Institute. Her research interests are machine learning and image processing. She is the member of China Computer Federation.

Xiangheng Jiang received the B.Eng. degree in electrical engineering from the Institute of Disaster Prevention, in 2019. He is currently pursuing the M.S. degree in control science and engineering with the Beijing Key Laboratory of Multimedia and Intelligent Software Technology, Faculty of Information Technology, Beijing University of Technology, Beijing. His research interests include intelligent transportation systems, deep learning, and artificial intelligence.

Yongli Hu (Member, IEEE) received the Ph.D. degree from the Beijing University of Technology, China, in 2005. He is currently a Professor with the Faculty of Information Technology, Beijing University of Technology. He is also a Researcher with the Beijing Key Laboratory of Multimedia and Intelligent Software Technology, and the Beijing Artificial Intelligence Institute. His research interests include computer graphics, pattern recognition, and multimedia technology.



Fuqing Duan (Member, IEEE) received the B.S. and M.S. degrees in mathematics from Northwest University, Xi'an, China, in 1995 and 1998, respectively, and the Ph.D. degree in pattern recognition from the National Laboratory of Pattern Recognition, Beijing, China, in 2006. He is currently a Professor with the School of Artificial Intelligence, Beijing Normal University, Beijing. His current research interests include 3D face reconstruction, skull identification, and machine learning and applications. He has authored more than 80 conference and journal articles on related topics.

Kan Guo received the bachelor's degree in mathematics and physics from the School of Beijing University of Posts and Telecommunications of China, in 2015. He is currently pursuing the Ph.D. degree in control science and engineering with the Beijing Key Laboratory of Multimedia and Intelligent Software Technology, the Faculty of Information Technology, Beijing University of Technology, Beijing, China. His research interests include intelligent transportation systems, deep learning, and artificial intelligence.

Boyue Wang received the B.Sc. degree in computer science from the Hebei University of Technology, China, in 2012, and the Ph.D. degree from the Beijing University of Technology, China, in 2018. He is currently a Post-Doctoral Researcher with the Beijing Municipal Key Laboratory of Multimedia and Intelligent Software Technology, Beijing University of Technology, Beijing. His current research interests include computer vision, pattern recognition, manifold learning, and kernel methods.

Junbin Gao (Member, IEEE) graduated from the Huazhong University of Science and Technology (HUST), China, in 1982, and the B.Sc. degree in computational mathematics and the Ph.D. degree from the Dalian University of Technology, China, in 1991. He was a Professor in computer science at the School of Computing and Mathematics, Charles Sturt University, Australia. From 1982 to 2001, he was an Associate Lecturer, a Lecturer, an Associate Professor, and a Professor at the Department of Mathematics, HUST. From 2001 to 2005, he was a Senior Lecturer and a Lecturer in computer science, at University of New England, Australia. He is currently a Professor of big data analytics with The University of Sydney Business School, The University of Sydney. His main research interests include machine learning, data analytics, Bayesian learning and inference, and image analysis.

Baocai Yin (Member, IEEE) received the M.S. and Ph.D. degrees in computational mathematics from the Dalian University of Technology, Dalian, China, in 1988 and 1993, respectively. He is currently the Director with the Beijing Key Laboratory of Multimedia and Intelligent Software Technology and the Beijing Artificial Intelligence Institute. He has authored or coauthored more than 200 academic papers in prestigious international journals including IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, IEEE TRANSACTIONS ON MULTIMEDIA, IEEE TRANSACTIONS ON IMAGE PROCESSING, IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, IEEE TRANSACTIONS ON CYBERNETICS, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, and top-level conferences such as CVPR, AAAI, INFOCOM, IJCAI, and ACM SIGGRAPH. His research interests include multimedia, image processing, computer vision, and pattern recognition.