A Multi-graph Fusion Based Spatiotemporal Dynamic Learning Framework

Xu Wang
wx309@mail.ustc.edu.cn
University of Science and Technology of China
Hefei, China

Lianliang Chen
cll006@mail.ustc.edu.cn
University of Science and Technology of China
Hefei, China

Hongbo Zhang
zhb9988@mail.ustc.edu.cn
University of Science and Technology of China
Hefei, China

Pengkun Wang
pengkun@mail.ustc.edu.cn
University of Science and Technology of China
Hefei, China

Zhengyang Zhou
zzy0929@mail.ustc.edu.cn
University of Science and Technology of China
Hefei, China

Yang Wang
angyan@ustc.edu.cn
University of Science and Technology of China
Hefei, China

ABSTRACT

Spatiotemporal data forecasting is a fundamental task in the field of graph data mining. Typical spatiotemporal data prediction methods usually capture spatial dependencies by directly aggregating features of local neighboring vertices in a fixed graph. However, this kind of aggregators can only capture localized correlations between vertices, and while been stacked for larger receptive field, they fall into the dilemma of over-smoothing. Additional, in temporal perspective, traditional methods focus on fixed graphs, while the correlations among vertexes can be dynamic. And time series components integrated strategies in traditional spatiotemporal learning methods can hardly handle frequently and drastically changed sequences. To overcome those limitations of existing works, in this paper, we propose a novel multi-graph based dynamic learning framework. First, a novel Dynamic Neighbor Search (DNS) mechanism is introduced to model global dynamic correlations between vertices by constructing a feature graph (FG), where the adjacency matrix is dynamically determined by DNS. Then we further alleviate the over-smoothing issue with our newly designed Adaptive Heterogeneous Representation (AHR) module. Both FG and origin graph (OG) are fed into the AHR modules and fused in our proposed Multi-graph Fusion block. Additionally, we design a Differential Vertex Representation (DVR) module which takes advantage of differential information to model temporal trends. Extensive experiments illustrate the superior forecasting performances of our proposed multi-graph based dynamic learning framework on six real-world spatiotemporal datasets from different cities and domains, and this corroborates the solid effectiveness of our proposed framework and its superior generalization ability.

Prof. Yang Wang is the corresponding author.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
WSDM '23, February 27–March 3, 2023, Singapore, Singapore.
© 2023 Association for Computing Machinery.
ACM ISBN 978-1-4503-9407-9/23/02…$15.00
https://doi.org/10.1145/3539597.3570396

CCS CONCEPTS

• **Information systems** → **Spatial-temporal systems**; **Data mining**.

KEYWORDS

spatiotemporal data, traffic prediction, data mining

ACM Reference Format:
Xu Wang, Lianliang Chen, Hongbo Zhang, Pengkun Wang, Zhengyang Zhou, and Yang Wang. 2023. A Multi-graph Fusion Based Spatiotemporal Dynamic Learning Framework. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining (WSDM '23), February 27–March 3, 2023, Singapore, Singapore.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3539597.3570396

1 INTRODUCTION

Spatiotemporal forecasting, which is an important research branch for analyzing the spatial correlations and temporal tendencies of spatiotemporal data, has attracted more and more attentions. Extensive efforts [22–24, 26, 27, 35] have been devoted to applications of complex systems such as urban traffic volume prediction, human mobility recognition, and air pollution propagation analysis.

Essentially, the future status of a vertex in a graph is simultaneously correlated with the status of some other vertices as well as the historical statuses of itself. Correspondingly, existing spatiotemporal data forecasting methods are typically composed of two components, which are applied to extract spatial and temporal patterns respectively. Early works mostly focus on approximating the complex spatiotemporal patterns with Convolution Neural Network (CNN) and Recurrent Neural Network (RNN) integrated framework [3, 5, 30]. Nevertheless, all these methods, which can only reconstruct spatial data into image-like manners, cannot abstract the non-Euclidean correlations in spatial perspective. To this end, researchers consider to model these Non-Euclidean structured characteristics with graph structure. [21] captures spatial dependencies with bidirectional random walks on the graph, and captures temporal dependencies with the encoder-decoder framework and scheduled sampling; [42] extends traditional CNN to graph-structured data to build an end-to-end traffic prediction network. Next, with the development of GCNs, researchers attempt

to extract the complex spatial correlations with GCN based methods. For instance, [39] propose a novel Spatiotemporal-GCN model which combines graph convolution with standard 1D convolution together, and the computation efficiency of this proposed model is significantly better than the computation efficiencies of traditional RNNs; [44] devises a novel Temporal-GCN model, which combines GCN with GRU to exploit the spatiotemporal correlations of urban traffics. However, these above-mentioned graph modeling approaches are all based on predefined fixed graph structures which ignore the dynamic spatiotemporal correlations among different vertices. To address the dynamic issue, some state-of-the-art solutions start to take initial steps on extracting the dynamic spatiotemporal correlations within graph-structured data. For instance, [11] uses spatiotemporal attention mechanism to capture the dynamic spatiotemporal correlations during different time intervals, and [48] proposes a novel Attention Temporal Graph Convolutional Network (A3T-GCN) for the application of traffic flow forecasting, and this proposed network can simultaneously capture both spatial and dynamic temporal correlations. Most recent works, [31] and [18], construct temporal graphs based on the similarities between time series of vertices, and the constructed temporal graphs can facilitate the caption of global correlations between vertices. Nevertheless, all these dynamic-issue oriented methods use invariable spatial processing in different time intervals, and can hardly handle drastic and frequent changing temporal series. *In summary, even though the effectiveness of spatiotemporal data prediction on simultaneously learning spatial and temporal correlations with existing methods has been prominently verified, there still exist some problems with existing methods in both spatial and temporal perspectives.*

Spatial challenges. Even though these GNN based spatiotemporal methods can effectively extract neighboring correlations within graphs, they still cannot exploit long-range correlations within graphs, hence falling into the conflict between over-smoothing and localized receptive field. Specifically, there do exist some long-range correlations within graphs, and these information is essential for accurately approximating spatiotemporal patterns. In some real applications, two neighboring vertices in graph may be not definitely correlated, two remote vertices may be strongly correlated. For example, a residential area and a commercial area are far apart, but connected by a subway line. The flow of people in the two areas has a great correlation, but they are usually far apart on a graph, that is, they are multi-hop neighbors. Given the short-sighted nature of single aggregator, existing GNNs can hardly capture such long-range correlations between vertices, hence the localized receptive field of spatiotemporal learning. To enlarge receptive field, existing GNNs usually stack multilayer aggregators [16, 17, 19]. However, limited by the existing GNN aggregation strategy, the embedded features of all vertices may converge to a subspace unrelated to themselves during message transmission process, and eventually lead to feature failure [29]. This phenomenon is called over-smoothing where the representations of neighboring vertices may become more similar when the stack of multilayer aggregators grows [19, 28, 29].

Temporal challenges. First, the spatial correlations among vertices in different time stamps can be dynamic. For instance, the correlations between the monitored pollution concentrations of two monitoring stations in a city are not only related to the geographical deployments of these two stations, but also highly impacted by the globally dynamic meteorology features of the city such as the speed and direction of wind. However, existing methods apply constant spatial networks on different time stamps. This may consequently lead to static spatial correlations within different time stamps, i.e., temporal stationary. Even though some recent works [28, 31, 37] have taken initial steps on efficiently capturing dynamic correlations by dynamically rebuilding the adjacency matrices of graphs, with no exception, all these methods may potentially lose the inherent structures of graphs. Second, commonly used temporal series prediction models, such as RNN [10, 12, 20] and CNN [38, 39], can not handle drastic and frequent changes, thus performing poorly on short-term prediction on those time stamps.

In summary, these issues are non-negligible obstacles in understanding complex and dynamic spatial correlations and temporal tendencies in spatiotemporal data prediction. To address these challenges, in this paper, we propose a multi-graph fusion based graph neural network for spatiotemporal dynamic learning. In particular, we first propose a multi-graph mechanism which respectively employs feature graph to learn global dynamic correlations and origin structure graph to learn local static correlations from spatiotemporal data. To alleviate the problem of spatial over-smoothing issue, we devise a dynamic learning component, Adaptive Heterogeneous Representation (AHR) to filter irrelevant embedded features of neighboring vertices to reduce redundancy in spatial perspective. For constructing feature graph to address the issues of localized receptive field and temporal stationary in both spatial and temporal perspectives, we propose a novel method, Dynamic Neighbor Search (DNS), to dynamically select partially most correlated neighbors in each time interval for each vertex, hence capturing global dynamic correlations. The feature graph constructed by DNS and the origin structure graph are fed into AHR to generate embedded features for vertices and the generated results of both graphs are fused by our multi-graph fusion block. Also, we propose Differential Vertex Representation (DVR), which utilizes differential information to help to handle complex inputs with drastic and frequent changes, and to aggregate the current embedded information of each vertex with the difference between the current and last time step's embedded information of each vertex to generate a new heterogeneous representation for each vertex in temporal perspective. The main contributions of this paper are as follows.

- To tackle the conflict between over-smoothing and localized receptive field which is inherent for existing spatiotemporal learning methods, we propose an AHR module which takes advantage of attention and gated mechanisms for alleviating over-smoothing issue and a DNS method to construct feature graph for capturing global dynamic correlations.
- To dynamically learn global and local correlations within graph-structured data, we propose a novel multi-graph based spatiotemporal learning framework, where both constructed feature graph and origin structure graph are fused. Additionally, a novel DVR module is designed to handle drastic and frequent changing temporal series by considering interval-wise differential information.
- We conduct extensive experiments on cross-domain real-world datasets, and the results demonstrate our technical proposal is superior even when compared to state-of-the-art

solutions. And the visualizations of dynamic vertex-wise correlations derived with DNS and the evaluations of AHR's impacts on addressing spatial smoothness show that our model can effectively capture global dynamic correlations among vertices and alleviate the issue of over-smoothing, and this verifies the interpretability of our proposed approach.

2 RELATED WORKS

Great efforts have been devoted to addressing the spatiotemporal prediction issue in different application domains [9, 34, 36, 42, 43]. Previous works usually mesh object area into regular grids, and employ CNN based model to capture the spatial correlations of the entire area. For instance, ConvLSTM [30] uses a CNN and LSTM integrated model to capture both spatial and temporal correlations; ST-ResNet [40] predicts urban human mobility with a deep ResNet. However, all those grid-partition based methods may lose the inherent Non-Euclidean structured characteristics of graph-structured data during the phase of grid partition, and fail to effectively capture the deep spatial correlations.

To this end, in recent years, researchers consider to model these Non-Euclidean structured characteristics with graph structure. DCRNN [21] captures spatial dependencies with bidirectional random walks on the graph, and captures temporal dependencies with the encoder-decoder framework and scheduled sampling; SLCNN [42] extends traditional CNN to graph-structured data to build an end-to-end traffic prediction network. Recently, with the development of GCNs, researchers attempt to extract the complex spatial correlations with GCN based methods. For instance, [39] propose a novel STGCN model which combines graph convolution with standard 1D convolution together, and the computation efficiency of this proposed model is significantly better than the computation efficiencies of traditional RNNs; [44] devises a novel T-GCN model, which combines GCN with GRU to exploit the spatiotemporal correlations of urban traffics. Nevertheless, these above-mentioned graph modeling approaches are all based on predefined fixed graph structures, and the dynamic spatiotemporal correlations among different vertices are ignored by those predefined fixed graphs.

To address the dynamic issue, some SOTA solutions start to take initial steps on extracting the dynamic spatiotemporal correlations within graph-structured data. For instance, ASTGCN [11] uses spatiotemporal attention mechanism to capture the dynamic spatiotemporal correlations during different time intervals, and [48] proposes a novel Attention Temporal Graph Convolutional Network (A3T-GCN) for the application of traffic flow forecasting, and this proposed network can simultaneously capture both spatial and dynamic temporal correlations. STFGNN [18], construct temporal graphs based on the similarities between time series of vertices, and the constructed temporal graphs can facilitate the caption of global correlations between vertices. [2, 37] construct adjacency correlations by learning embeddings for each vertices and calculating similarities among embeddings. While those methods focus on the dynamic correlations, in our proposed method, both dynamic correlations and origin spatial correlations are taken into consideration.

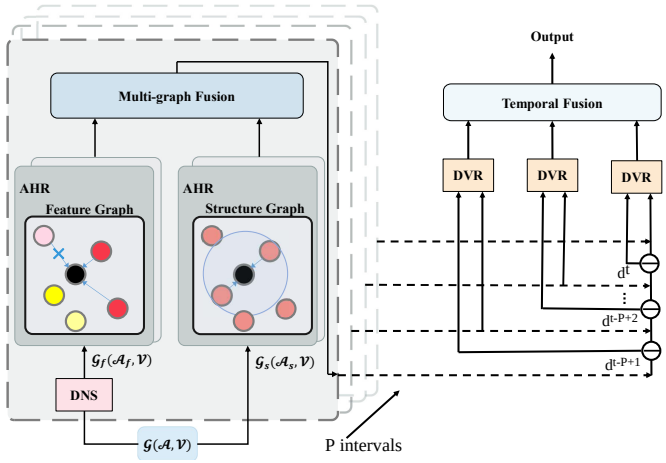


Figure 1: Solution overview.

3 PROBLEM DEFINITION

In this section, we formally define the problem of spatiotemporal data forecasting.

Def 1. Basic graph structure A spatial graph $\mathcal{G}$ can be modeled as $\mathcal{G}(\mathcal{A}, \mathcal{V}, \mathcal{E})$, where $\mathcal{A}$, $\mathcal{V}$ and $\mathcal{E}$ indicate the adjacency matrix, the sets of vertices and edges respectively, $v_i \in \mathcal{V}(1 \le i \le |\mathcal{V}|)$ corresponds to the i-th vertex in $\mathcal{V}$. Since $\mathcal{A}$ contains the same information as $\mathcal{E}$, we may also denote a graph as $\mathcal{G}(\mathcal{A}, \mathcal{V})$ or $\mathcal{G}(\mathcal{V}, \mathcal{E})$.

Def 2. Historical spatiotemporal feature The features matrix of all vertices in graph $\mathcal{G}$ in time interval t can be denoted as $\mathcal{V}^t \in \mathbb{R}^{|\mathcal{V}| \times D}$ where D is the dimensionality of the feature of each vertex. The historical spatiotemporal features of graph $\mathcal{G}$ in historical P time intervals can be formulated as $\left[\mathcal{V}^{(t-P+1)}, \mathcal{V}^{(t-P+2)}, \cdots, \mathcal{V}^t\right] \in \mathbb{R}^{|\mathcal{V}| \times D \times P}$.

Def 3. Spatiotemporal data forecasting The main task of spatiotemporal data forecasting is to learn a function f to predict the spatiotemporal data of $\mathcal{G}$ in the future Q time intervals based on the data of $\mathcal{G}$ in historical P time intervals, i.e.,

$$\left[\mathcal{V}^{t-P+1}, \cdots, \mathcal{V}^t, \psi\right] \xrightarrow{f} \left[\widehat{\mathcal{V}^{t+1}}, \cdots, \widehat{\mathcal{V}^{t+Q}}\right]. \tag{1}$$

where ψ indicates the set of learnable parameters in f.

4 METHOD

The main architecture of the proposed learning framework is illustrated in Figure 1. For a give spatiotemporal data $\mathcal{G}(\mathcal{A}, \mathcal{V})$ that corresponds to P intervals, the DNS module calculates P dynamic adjacency matrices $\mathcal{A}_f = [\mathcal{A}_f^{t-P+1}, \cdots, \mathcal{A}_f^t]$ for constructing a feature graph $\mathcal{G}(\mathcal{A}_f, \mathcal{V})$. Then the feature graph $\mathcal{G}(\mathcal{A}_f, \mathcal{V})$ and origin structure graph $\mathcal{G}(\mathcal{A}, \mathcal{V})$ are fed into AHR modules for extracting spatial patterns. The Multi-graph Fusion module further fuses the outputs of AHR modules for both graphs, and the fusion results are taken as input by DVR module for spatiotemporal prediction. The details of each individual component of our model are described below.

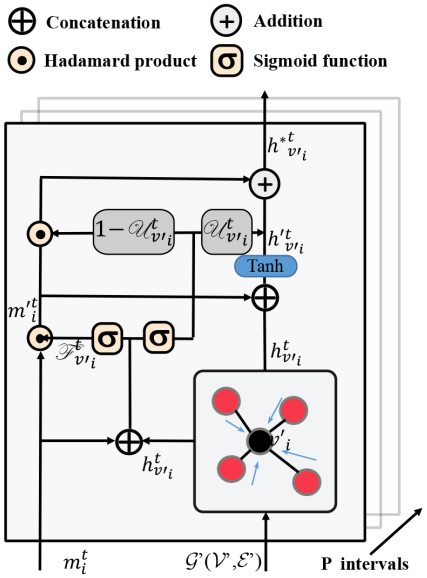


Figure 2: Calculation of learnable gating mechanism.

4.1 Dynamic neighbor search for feature graph

To dynamically and adaptively capture the global spatial correlations among features of vertices, regarding a specific time interval t, we denote the feature graph as $\mathcal{G}_f^t = \left(\mathcal{A}_f^t, \mathcal{V}_f^t\right)$, where $\mathcal{V}_f^t$ indicates the vertex set of $\mathcal{G}_f^t$, and the adjacency matrix $\mathcal{A}_f^t$ is generated by DNS component which adaptively and dynamically selects related neighboring vertices to construct dynamic correlations among vertices in spatial perspective based on the features $\mathcal{V}^t$ of vertices $\mathcal{V}_f^t$.

Inspired by the feature selection approach [13], we embed the feature selection layer in our model. Based on the importance between a specified vertex and other vertices, DNS dynamically determine the neighbors of this vertex from the entire graph to avoid the issue of localized receptive field in fixed graphs. We here design an importance function, which calculates a value for each vertex to represent the importance of this vertex with regard to all vertices, to realize dynamic and adaptive neighbor selection of each vertex.

Given two vertice $v_i, v_c \in \mathcal{V}$, and the features of them in time interval t denoted as v_i^t, v_c^t, the importance value $s_{i\to c}^t$, which represents the importance of v_i to v_c, can be calculated by,

$$s_{i\to c}^t = \text{Relu}\left(\omega_c^{t\,T} v_i^t + \beta_c^t\right) \tag{2}$$

where ω_c^t and β_c^t are both learnable parameters which are corresponding to vertex v_c. Thus, for each vertex, we have an importance vector representing the importance of other vertices to it. Given the importance vectors for all vertices, for each vertex, we then select vertices with bigger importance values during interval t as neighbors. To achieve this target, we set a threshold τ. In case that $s_{i\to c}^t > \tau$, vertex v_i is connected with v_c as a neighbor in feature graph $\mathcal{G}_f^t$.

4.2 Adaptive heterogeneous representation for spatial smoothness

To address the issue of spatial smoothness, we here propose a novel AHR module, which contains two sub-components, a state-of-the-art Attention based aggregating function to respectively extract the correlations from feature and structure graphs, and a novel learnable gating mechanism to reduce information redundancy by adaptively flushing irrelevant information. Notice that our proposed AHR method can be employed on both feature and structure graphs, to simplify subsequent expressions, we unify the formal expressions of these two graphs to $\mathcal{G}'(\mathcal{A}', \mathcal{V}')$.

Attention based aggregating function. By using a state-of-the-art attention based aggregating function $\phi(\mathcal{A}'^t, \mathcal{V}'^t, \zeta^t)$, which is proposed in [32], we extract the spatial correlations among vertices in time interval t. Notice here ζ^t is the learnable parameter set. Given a vertex $v_j' \in \mathrm{N}(v_i') \bigcup \{v_i'\}$ where function $\mathrm{N}(\cdot)$ calculates the neighbors of a vertex, we calculate the significance of vertex v_j' to vertex v_i', i.e.,

$$\mathrm{I}\left(v'^t_i, v'^t_j\right) = \frac{\exp\left(\mathrm{LR}\left((\gamma^t)^T \left[\theta_i^t v'^t_i \parallel \theta^t v'^t_j\right]\right)\right)}{\sum\limits_{v_k' \in \mathrm{N}(v_i') \bigcup \{v_i'\}} \exp\left(\mathrm{LR}\left((\gamma^t)^T \left[\theta^t v'^t_i \parallel \theta^t v'^t_k\right]\right)\right)} \tag{3}$$

where $\parallel$ means the operation of concatenation, and LR denotes the LeakyRelu function with negative input slope of 0.2. After calculating the significance, we finally calculate the embedded features of each vertex by

$$h_{v_i'}^t = \text{Tanh}\left(\sum_{v_j' \in \mathrm{N}(v_i') \bigcup \{v_i'\}} \mathrm{I}\left(v'^t_i, v'^t_j\right) \theta'^t_j v'^t_j\right) \tag{4}$$

Notice that in the above two equations, γ^t and θ^t are learnable parameters involved in ζ^t. Here $h_{v_i'}^t \in \mathbb{R}^F$ denotes the outputted embedded features with dimensionality F.

Learnable gating mechanism. To flush irrelevant information and reduce redundancy, we propose a learnable gating mechanism with two sub-components, memory and gating function to adaptively select and filter features of neighboring vertices. The detailed calculation procedure of the learnable gating mechanism is illustrated in Figure 2. The memory sub-component is used to store the initial state vectors of vertices which represent the states of vertices before aggregation. Given a vertex v_i' and time interval t, its state vector m_i^t can be initialized by v'^t_i, and then concatenates with the output of the aggregating function. The concatenation is then fed into the gating function of v_i', which contains a forgetting gate $\mathcal{F}_{v_i'}^t$ and an updating gate $\mathcal{U}_{v_i'}^t$, i.e.,

$$\begin{cases} \mathcal{F}_{v_i'}^t = \text{Sigmoid}\left(\tau_{\mathcal{F}}^t \left[m_i^t \parallel h_{v_i'}^t\right]\right) \\ \mathcal{U}_{v_i'}^t = \text{Sigmoid}\left(\tau_{\mathcal{U}}^t \left[m_i^t \parallel h_{v_i'}^t\right]\right) \end{cases} \tag{5}$$

We first employ the forgetting gate to filter the initial state vector m_i^t and flush its contained irrelevant information, i.e.,

$$m'^t_i = \varpi_m^T m_i^t \odot \mathcal{F}^t_{v'_i} \tag{6}$$

where $\varpi_m^T \in \mathbb{R}^{F \times F_m}$ is a learnable parameter matrix and F_m denotes the dimensionality of m_i^t, operation $\odot$ corresponds to Hadamard product, then we update the embedded feature of vertex v'_i with this filtered state vector, i.e.,

$$h'^t_{v'_i} = \text{Tanh}\left(\varsigma^t \left[m'^t_i \parallel h^t_{v'_i}\right]\right) \tag{7}$$

Notice that $\tau^t_{\mathcal{U}}$ and ς^t are both learnable parameters. Finally, we use the updating gate $\mathcal{U}^t_{v'_i}$ to select the most correlated information to update the embedded feature of vertex v'_i,

$$h^{*t}_{v'_i} = \left(1 - \mathcal{U}^t_{v'_i}\right) \odot h^t_{v'_i} + \mathcal{U}^t_{v'_i} \odot h'^t_{v'_i} \tag{8}$$

Our AHR method can be used as the basic component of graph convolution for addressing the over-smoothing issue.

4.3 Multi-graph fusion

After calculating the filtered embedded features of vertex v'_i, the filtered embedded features $h^{*t}_{v'_i}$ of vertex v'_i can be separably denoted as $hf^t_{v'_i}$ and $hs^t_{v'_i}$ for feature and structure graphs. For multi-graph fusion, we first concatenate these two filtered embedded features, and then use a fully connected layer to aggregate the information of both feature and structure graphs, i.e.,

$$s^t_{v'_i} = \text{Tanh}\left(\omega^t_{v'_i}\left[hf^t_{v'_i} \parallel hs^t_{v'_i}\right] + b^t_{v'_i}\right) \tag{9}$$

where $\omega^t_{v'_i}$ and $b^t_{v'_i}$ are both learnable parameters. So far, we then obtain the multi-graph fused spatial correlations of all vertices.

4.4 Differential vertex representation based temporal learning

Traditional spatiotemporal learning methods usually employ similar spatial learning methods on different time intervals, and this kind of duplication of methods in different intervals may ignore some key variations of spatiotemporal data in temporal perspective, hence the poor ability to handle drastically and frequently changing temporal series. To this end, we devise a novel DVR module to dynamically enhance the distinctiveness and diversities of features of vertices in temporal perspective. The details of DVR are demonstrated in Figure 3. By utilizing attention mechanism to fuse the information of P historical time intervals, our model are able to capture the global temporal trends of vertices.

4.4.1 *Differential vertex representation.*

Regarding a vertex v'_i, we first calculate the differential vector between $s^t_{v'_i}$ and $s^{t+1}_{v'_i}$ by

$$d^{t+1}_{v'_i} = \text{FC}\left(s^{t+1}_{v'_i}\right) - \text{FC}\left(s^t_{v'_i}\right) \tag{10}$$

where FC denotes fully connected layers. Note that if interval t is the earliest interval in a sequence, we let $d^t_{v'_i} = \text{FC}\left(s^t_{v'_i}\right)$. With these differential vectors, we then update the representations of vertex v'_i during each time interval by

$$s^{*t}_{v'_i} = \eta^t_{v'_i} \odot s^t_{v'_i} + \left(1 - \eta^t_{v'_i}\right) \odot d^t_{v'_i} \tag{11}$$

where

$$\eta^t_{v'_i} = \text{Sigmoid}\left(\pi^t_{v'_i}\left[s^{*t}_{v'_i} \parallel d^t_{v'_i}\right]\right) \tag{12}$$

Here $\pi^t_{v'_i}$ is a learnable parameter, and the output of activation function Sigmoid is within the range of $[0, 1]$.

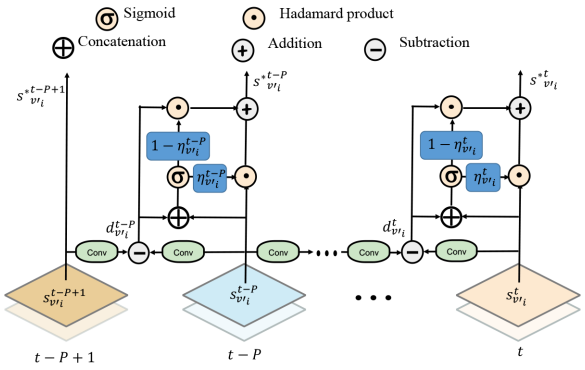


Figure 3: Illustration of diversified vertex representation.

4.4.2 *Temporal fusion of historical intervals.*

In temporal perspective, we employ a simplified attention mechanism to fuse the information of P historical intervals into one single graph. Given the historical embedded features $\left\{s^{*t-P+1}_{v'_i}, \cdots, s^{*t}_{v'_i}\right\}$ of vertex v'_i, we first construct the dynamic correlations between the predicted future values and all P historical embedded features by

$$\sigma^t_{v'_i} = \text{Softmax}\left(\rho^t_{v'_i}\right) = \frac{\exp\left(\rho^t_{v'_i}\right)}{\sum_{k=t-P+1}^{t} \exp\left(\rho^k_{v'_i}\right)} \tag{13}$$

where

$$\rho^t_{v'_i} = \text{Tanh}\left(\delta^t_{v'_i} s^{*t}_{v'_i} + a^t_{v'_i}\right) \tag{14}$$

Note that $\delta^t_{v'_i}$ and $a^t_{v'_i}$ are both learnable parameters. The value of $\sigma^t_{v'_i}$ denotes that the attention score of the embedded feature of vertex v'_i in time interval t on the future value of v'_i. After calculating the scores for all P historical time intervals, we then predict the status of vertex v'_i by calculating the aggregation of the embedded features of vertex v'_i during P historical intervals, i.e.,

$$\widehat{x^t_{v'_i}} = \sum_{k=t-P+1}^{t} \sigma^k_{v'_i} s^{*k}_{v'_i} \tag{15}$$

Note that the embedded features $s^{*k}_{v'_i}$ has involved the spatial correlations between v'_i and its neighbors.

Table 1: Evaluations on PeMSD4, PeMSD8, AirBJ and TrafficSIP. The best results are in bold and _ denotes the second-best results.

Model	PEMSD4			PEMSD8			AirBJ			TrafficSIP		
	MAE	RMSE	MAPE(%)	MAE	RMSE	MAPE(%)	MAE	RMSE	MAPE(%)	MAE	RMSE	MAPE(%)
ARIMA	33.73	48.8	24.18	31.09	44.32	22.73	44.44	66.84	104.06	1.5	1.93	74.17
LSTM	26.77	40.65	18.23	23.09	35.17	14.99	61.94	66.61	98.18	2.09	2.54	54.65
STGCN	21.16	34.89	13.83	17.5	27.09	11.29	48.7	70.56	172.7	1.98	2.9	50.5
STG2Seq	25.2	38.48	18.77	20.17	30.71	17.32	44.27	68.11	120.93	1.74	2.54	46.6
STSGCN	21.19	33.65	13.9	17.13	26.8	10.96	50.11	72.24	149.64	1.44	2.42	43.46
AGCRN	19.83	32.26	12.97	15.95	25.22	10.09	33.43	54.81	100.95	1.99	2.97	57.02
STFGNN	19.83	31.88	13.02	16.64	26.22	10.6	33.9	56.77	68.47	1.34	2.08	34.16
STG-NCDE	19.21	**31.09**	12.76	15.45	24.81	**9.92**	**33.15**	58.3	70.45	1.36	2.14	37.18
MGTF	**19.12**	31.67	**12.35**	**15.32**	**24.62**	10.07	33.28	**56.56**	**69.61**	**0.91**	**1.74**	**19.72**

4.5 Loss and training

To train our model, we select Mean Absolute Error (MAE) as the training object of our proposed approach, and the loss function can be defined as,

$$\text{Loss}\left(\widehat{\mathcal{V}^{t+1}}, \cdots, \widehat{\mathcal{V}^{t+Q}}\right) = \frac{1}{Q * |\mathcal{V}|} \sum_{j=1}^{Q} \sum_{i=1}^{|\mathcal{V}|} \left| \widehat{v_i^j} - v_i^j \right| \tag{16}$$

Where $\widehat{v_i^j}$ is the predicted status of vertex v_i during interval j and v_i^j indicates the corresponding ground truth. To avoid the issue of overfitting, we use L2-Regularization [8] during training phase.

5 EXPERIMENTS

In this section, to evaluate the performances of our proposed multi-graph fusion based model, we conduct a series of experiments on six real-world spatiotemporal datasets, including four public datasets (PeMSD4, PeMSD8, BikeNYC, and TaxiBJ) [11, 40, 41] and two private datasets (AirBJ and TrafficSIP). Worth noting that, with these six datasets, our experiments consider four different categories of spatiotemporal data, i.e., traffic flow, traffic speed, air quality, and crowd flow. And with these multiple datasets from different domains of urban traffic, human mobility, and air quality, we roundly validate the scalability of our proposed approach.

5.1 Dataset Description

In this subsection, we then introduce the detailed information about all used datasets. The four used public datasets contain PeMSD4, PeMSD8, BikeNYC, and TaxiBJ. **PeMSD4** refers to the traffic data in Bay Area, San Francisco during January to February in 2018. In PeMSD4, there are 307 detectors in total, and the collected data are aggregated every 5 minutes, so each detector contains 288 data points per day. **PeMSD8** indicates the traffic data collected by 170 detectors in San Bernardino from July to August in 2016. The traffic data in PeMSD8 are aggregated with the same sampling frequency in PeMSD4. For both PeMSD4 and PeMSD8, they contain the total traffic flows, average speeds, and average occupancies of monitored road segments, and our proposed method is used to predict the traffic flows of road segments. **BikeNYC** dataset is taken from the NYC Bike system during 1st Apr. to 30th Sept., 2014. For constructing this dataset, the urban area of New York city is partitioned into a regular 16×8 grid map based on longitude and latitude, and the hourly crowd inflow and outflow of each grid are recorded as the crowd flow in the dataset. **TaxiBJ** corresponds to the taxicab GPS data in Beijing during 1st Nov., 2015 to 10th Apr., 2016 and the sampling interval is 30 minutes. Similar to the construction of BikeNYC dataset, the urban area of Beijing is divided into a 32×32 Manhattan grid, and each record in TaxiBJ corresponds to the taxicab inflow and outflow of a specific grid within 30 minutes. To further investigate our proposed model, we then investigate the performances of our proposed model on two private datasets. **AirBJ** includes the air quality data of 36 air monitoring stations in Beijing during 1st May, 2014 to 30th Apr., 2015, and the sampling interval of air quality is one hour, and each air quality record in this dataset consists of the concentration of six different air pollutants, i.e., NO_2, SO_2, O_3, CO, $PM_{2.5}$ and PM_{10}. **TrafficSIP** dataset [47] contains the traffic volumes of 108 surveilled intersections in Suzhou Industrial Park during 1st Jan. to 31th, Mar., 2017, the surveilled data includes the traffic flows and traffic speeds of surveilled intersections, and our model is used to predict the traffic speed of each intersection.

Table 2: Evaluations on BikeNYC and TaxiBJ. The best results are in bold and _ denotes the second-best results.

Model	BikeNYC			TaxiBJ		
	MAE	RMSE	MAPE(%)	MAE	RMSE	MAPE(%)
ARIMA	8.79	16.6	108.25	37.76	60.03	70.26
LSTM	7.57	10.53	102.3	46.69	55.77	104.23
STGCN	12.93	19.61	108.54	82.51	110.84	258.89
STG2Seq	11.76	18.32	105.53	73.35	105.5	239.1
STSGCN	12.64	19.01	95.62	80.92	113.6	248.54
AGCRN	11.63	21.17	126.55	23.1	42.27	64.39
STFGNN	5.37	9.19	48.03	22.16	36.46	62.62
STG-NCDE	5.19	9.53	41.98	22.97	36.95	63.56
MGTF	**4.84**	**8.45**	**41.14**	**21.99**	**35.85**	**57.83**

5.2 Experiment Settings

To verify the effectiveness of our proposed model, we compare it with the following alternative baselines. **ARIMA** [4] is a well-known time series analysis method for predicting future values. **LSTM** [14] treats different nodes separately by utilizing LSTM.

STGCN [39] deploys graph convolution and temporal convolution for capturing spatial and temporal dependencies, respectively. **STG2Seq** [1] uses multiple gated graph convolutional module and seq2seq architecture with attention mechanism to make multi-step predictions. **STSGCN** [31] is able to effectively capture the complex localized spatiotemporal correlations through an elaborately designed spatiotemporal synchronous modeling mechanism. **AGCRN** [2] proposes node-adaptive graph convolution, which generates node-specific parameters from a parameter pool shared by all nodes according to learnable node embeddings, and combines it with GRU. **STFGNN** [18] can effectively learn hidden spatiotemporal dependencies based on a novel fusion operation of various spatial and temporal graphs. **STG-NCDE** [7] extends the concept of neural controlled differential equations and design two novel NCDEs for spatial and temporal processing, respectively.

To eliminate the influence of different value ranges of different data items, we first normalize all input data by standard normalization for more stable training. And for all datasets and experiments, we select earliest 60% of data as the training sets, use the following 20% of data as validation sets, and retain the rest 20% for validation. The model is trained on Tesla V100 GPU with the batch size of 64, and Adam [15] is adopted as optimizer. Further, we implement the model in Python 3.6 with PyTorch 1.9.0, and trained for 100 epochs on all datasets.

For multi-step forecasting, we use the data during 12 continuous past time steps as the historical time window ($P = 12$) to forecast the data during next 12 continuous time steps ($Q = 12$). We employ grid search strategy to locate the best parameters on validations, and use three evaluation metrics, i.e., Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE), to evaluate the performances.

5.3 Main experiments

We evaluate our proposed model and all alternatives based on four public and two private datasets, and the results are demonstrated in Table 1 and 2. As can be obviously observed, our proposed model outperforms all alternatives on all datasets.

As can be observed from Table 1 to 2, our proposed method can outperform all alternative solutions with all different datasets and predicting time steps, and this verify the superiority and cross-domain scalability of our proposed model. Since AirBJ contains more drastically changing data, our model outperforms other methods with a considerable margin by taking advantage of DVR module. Meanwhile, the performance enhancements of our proposed model are not such significant on BikeNYC and TaxiBJ, the reason may be that the inflow and outflow data of quite a few grids equal to 0 during some intervals, and this kind of data sparsity can significantly influence the stability of complex deep learning methods [33, 46].

5.4 Ablation Study

To further illustrate the effectiveness of each individual novel component in our model, we evolve four variants and exploit extensive ablative experiments on AirBJ and PEMSD4. Notice that, in this part, we use past 12 continuous time steps ($P = 12$) to predict future 12 continuous time steps ($Q = 12$). The hyper-parameters are determined by the performance of 6 variants, which are described as below:

- **AHR**: This variant only employs individual AHR module on the fixed adjacent matrix from the structure graph.
- **AHR+DVR**: We add DVR module with individual AHR module to additionally solve the smoothness problem in time series in origin graph.
- **AHR+DNS**: We add DNS module with individual AHR module to achieve the dynamic selection of node neighbors in origin graph.
- **AHR+DNS+DVR**: We investigate the effectiveness of single Feature Graph (FG) by making predictions without structure graph, and AHR and DVR are employed to respectively capture both spatial and temporal correlations. Notice that DNS should be employed during the construction of FG.

The results of ablation study are shown in Table 3. Comparing with AHR, the performances of AHR+DVR are invariably better on both two datasets in terms of two different metrics, and this indicates the effectiveness of DVR module on temporal predictions. Homogeneously, AHR+DNS always outperforms AHR on two datasets and two metrics, and this verifies our DNS module can significantly solve the localized receptive field issue by dynamically selecting neighboring vertices. Regarding the single feature graph mechanism, the performance of AHR+DNS+DVR is better than the performances of AHR+DVR and AHR+DNS on both PeMSD4 and AirBJ. So far, the effectiveness of fusing these two graphs has been verified.

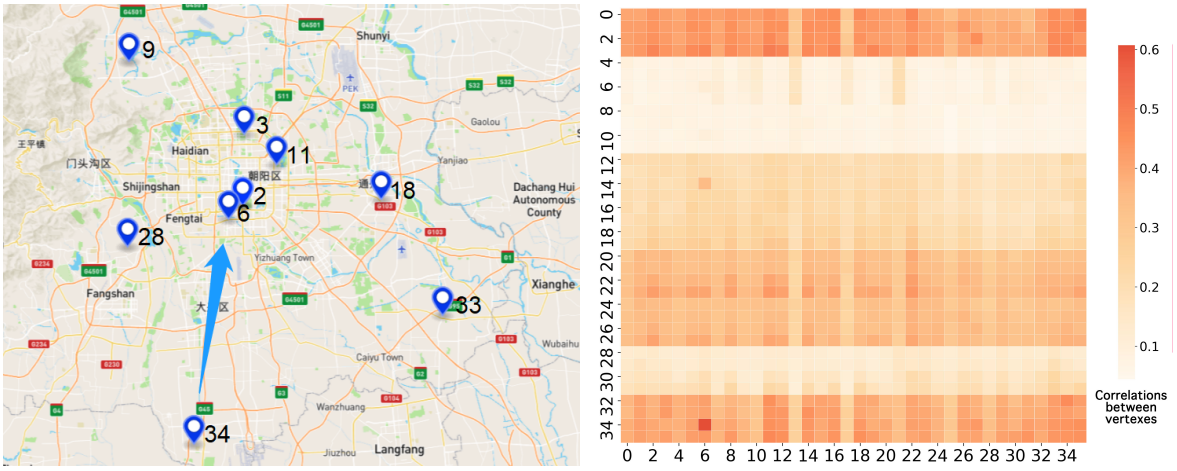
Table 3: Ablation studies on AirBJ and PeMSD4

Model	PeMSD4		AirBJ	
	MAE	RMSE	MAE	RMSE
AHR	32.65	44.36	49.99	75.76
AHR+DVR	27.15	38.51	44.36	69.66
AHR+DNS	21.99	33.44	38.62	63.92
AHR+DNS+DVR	21.49	32.82	37.92	63.52

5.5 The effectiveness of AHR on alleviating over-smoothing issue

Since our proposed AHR module aims at tackling the over-smoothing issue and can be used as a basic component of graph convolution, here we evaluate AHR module based on the metrics of Group Distance Ratio and Instance Information Gain which are introduced in [45]. Group distance ratio measures the ratio of inter-group distance over intra-group distance in Euclidean space, and a small group distance ratio leads to the over-smoothing issue where all groups are mixed together. Instance information gain represents how much input feature information is contained in the final representation, and the over-smoothing issue would lead to a small value of instance information gain.

We compare our AHR module with GCN and GAT when the number of layers is 1 to 10, the results are shown in Figure 5. Due to our well-designed attention and gated mechanism, the proposed AHR has overall better group distance ratios and instance information



(a) Geographical distribution of selected air monitoring stations.

(b) Heatmap of dynamic vertex-wise correlations among 36 nodes.

Figure 4: Visualization of dynamic vertex-wise correlations derived with DNS

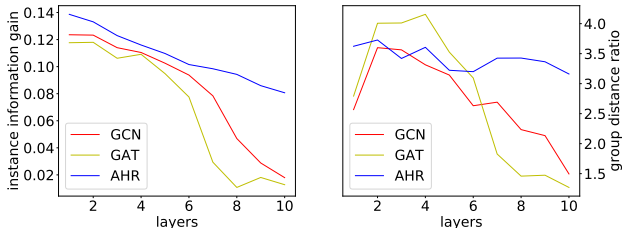


Figure 5: Evaluate AHR on over-smoothing metrics

gains and performs best on tackling over-smoothing. Also, when the number of layers grows, both metrics of AHR change slower than the other two alternatives which indicates the robustness of AHR on deeper network structure.

5.6 Visualization of dynamic vertex-wise correlations derived with DNS

In this subsection, we elaborate how DNS captures time-varying spatial correlation patterns, which are without explicit Euclidean associations, for further boosting entire dynamic graph representations. Specifically, we illustrate the selected vertices, i.e., air monitoring stations, of Beijing in Figure 4(a) and subsequently derive the vertex-wise correlation heat map with DNS in Figure 4(b). In particular, we observe that vertex 34 and 6 reveal a spatially distant correlation but tend to have a stronger connectedness beyond other pairs. This is really interesting and inscrutable intuitively, so we carry out an in-depth analysis on this. We think the inherent cause of this abnormal phenomena maybe lie in the particular industrial layout around Beijing. Specifically, pollutants and air pollution in Beijing are mostly from Hebei Province, which locates at the southwest of Beijing and is with intensive industrial parks, and as a consequence, pollutants, which move from the direction of vertex 34 and go directly to vertex 6, bring strong correlations between vertex 34 and 6. Further, directed seasonally prevailing wind and some specific terrains may also facilitate to the formation of this kind of strong correlations [6, 25]. From Figure 4, we also discover that there exist heterogenous regional pollution influences among different vertices, and this further imply various diffusion patterns of air pollution. For instance, the regional pollution around vertex 34 covers a larger spatial scope than the regional pollution around other vertices, and as illustrated, the surrounding area of vertex 11 is limitedly polluted. In conclusion, above thinking is consistent with real-world phenomena and can exactly verify the effectiveness and task-aware interpretability of our DNS module.

6 CONCLUSION

In this paper, we propose a novel multi-graph fusion based spatiotemporal dynamic learning framework to simultaneously address the issue of local receptive field and spatial smoothness for spatiotemporal data prediction. With a carefully designed feature graph, our model can learn global dynamic information in feature space by selecting most important neighbors from an entire graph with a newly designed Dynamic Neighbor Search (DNS) mechanism. Next, a newly designed Adaptive Heterogeneous Representation (AHR) is employed on these two graphs to reduce redundancy in spatial perspective by filtering irrelevant embedded feature information. Finally, to handle complex inputs with drastic and frequent changes, we propose a novel temporal Differential Vertex Representation (DVR) module to take temporal differential information among intervals into account. Extensive experiments on six real-world cross-domain and cross-city datasets demonstrate the superiority our proposed method.

ACKNOWLEDGMENTS

This paper is partially supported by the National Natural Science Foundation of China (No.62072427, No.12227901), the Project of Stable Support for Youth Team in Basic Research Field, CAS (No.YSBR-005), Academic Leaders Cultivation Program, USTC.

REFERENCES

[1] Lei Bai, Lina Yao, Salil Kanhere, Xianzhi Wang, Quan Sheng, et al. 2019. Stg2seq: Spatial-temporal graph to sequence model for multi-step passenger demand forecasting. *arXiv preprint arXiv:1905.10069* (2019).

[2] Lei Bai, Lina Yao, Can Li, Xianzhi Wang, and Can Wang. 2020. Adaptive graph convolutional recurrent network for traffic forecasting. *Advances in Neural Information Processing Systems* 33 (2020), 17804–17815.

[3] Jie Bao, Pan Liu, and Satish V Ukkusuri. 2019. A spatiotemporal deep learning approach for citywide short-term crash risk prediction with multi-source data. *Accident Analysis & Prevention* 122 (2019), 239–254.

[4] G. E. P. Box and David A. Pierce. 1970. Distribution of Residual Autocorrelations in Autoregressive-Integrated Moving Average Time Series Models. *J. Amer. Statist. Assoc.* 65, 332 (1970), 1509–1526.

[5] Longbiao Chen, Xiaoliang Fan, Leye Wang, Daqing Zhang, Zhiyong Yu, Jonathan Li, Nguyen Thi Mai Trang, Gang Pan, and Cheng Wang. 2018. RADAR: Road Obstacle Identification for Disaster Response Leveraging Cross-Domain Urban Data. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1 (01 2018), 1–23. https://doi.org/10.1145/3161159

[6] Yuan Chen, Nina Schleicher, Mathieu Fricker, Kuang Cen, Xiu-li Liu, Uwe Kaminski, Yang Yu, Xue-fang Wu, and Stefan Norra. 2016. Long-term variation of black carbon and PM2. 5 in Beijing, China with respect to meteorological conditions and governmental measures. *Environmental pollution* 212 (2016), 269–278.

[7] Jeongwhan Choi, Hwangyong Choi, Jeehyun Hwang, and Noseong Park. 2022. Graph Neural Controlled Differential Equations for Traffic Forecasting. (2022).

[8] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. 2001. *The elements of statistical learning*. Vol. 1. Springer series in statistics New York.

[9] Xu Geng, Yaguang Li, Leye Wang, Lingyu Zhang, Qiang Yang, Jieping Ye, and Yan Liu. 2019. Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 3656–3663.

[10] Kan Guo, Yongli Hu, Zhen Qian, Hao Liu, Ke Zhang, Yanfeng Sun, Junbin Gao, and Baocai Yin. 2020. Optimized graph convolution recurrent neural network for traffic prediction. *IEEE Transactions on Intelligent Transportation Systems* (2020).

[11] Shengnan Guo, Youfang Lin, Ning Feng, Chao Song, and Huaiyu Wan. 2019. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 922–929.

[12] Shengnan Guo, Youfang Lin, Shijie Li, Zhaoming Chen, and Huaiyu Wan. 2019. Deep spatial–temporal 3D convolutional neural networks for traffic data forecasting. *IEEE Transactions on Intelligent Transportation Systems* 20, 10 (2019), 3913–3926.

[13] Isabelle Guyon and André Elisseeff. 2003. An introduction to variable and feature selection. *Journal of machine learning research* 3, Mar (2003), 1157–1182.

[14] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[15] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[16] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).

[17] Guohao Li, Matthias Muller, Ali Thabet, and Bernard Ghanem. 2019. Deepgcns: Can gcns go as deep as cnns?. In *Proceedings of the IEEE International Conference on Computer Vision*. 9267–9276.

[18] Mengzhang Li and Zhanxing Zhu. 2021. Spatial-temporal fusion graph neural networks for traffic flow forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 35. 4189–4196.

[19] Qimai Li, Zhichao Han, and Xiao-Ming Wu. 2018. Deeper insights into graph convolutional networks for semi-supervised learning. *arXiv preprint arXiv:1801.07606* (2018).

[20] Ruoyu Li, Sheng Wang, Feiyun Zhu, and Junzhou Huang. 2018. Adaptive graph convolutional neural networks. *arXiv preprint arXiv:1801.03226* (2018).

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

[22] Yuxuan Liang, Songyu Ke, Junbo Zhang, Xiuwen Yi, and Yu Zheng. 2018. Geoman: Multi-level attention networks for geo-sensory time series prediction.. In *IJCAI*, Vol. 2018. 3428–3434.

[23] Yuxuan Liang, Kun Ouyang, Lin Jing, Sijie Ruan, Ye Liu, Junbo Zhang, David S Rosenblum, and Yu Zheng. 2019. Urbanfm: Inferring fine-grained urban flows. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 3132–3142.

[24] Ye Liu, Yu Zheng, Yuxuan Liang, Shuming Liu, and David S Rosenblum. 2016. Urban water quality prediction based on multi-task multi-view learning. In *Proceedings of the 25th international joint conference on artificial intelligence*.

[25] Zhaofeng Lv, Xiaotong Wang, Fanyuan Deng, Qi Ying, Alexander T Archibald, Roderic L Jones, Yan Ding, Ying Cheng, Mingliang Fu, Ying Liu, et al. 2020. Source–Receptor Relationship Revealed by the Halted Traffic and Aggravated Haze in Beijing during the COVID-19 Lockdown. *Environmental science & technology* (2020).

[26] Hao Miao, Jiaxing Shen, Jiannong Cao, Jiangnan Xia, and Senzhang Wang. 2022. MBA-STNet: Bayes-enhanced Discriminative Multi-task Learning for Flow Prediction. *IEEE Transactions on Knowledge and Data Engineering* (2022).

[27] Zheyi Pan, Yuxuan Liang, Weifeng Wang, Yong Yu, Yu Zheng, and Junbo Zhang. 2019. Urban traffic prediction from spatio-temporal data using deep meta learning. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 1720–1730.

[28] Hongbin Pei, Bingzhe Wei, Kevin Chen-Chuan Chang, Yu Lei, and Bo Yang. 2020. Geom-gcn: Geometric graph convolutional networks. *arXiv preprint arXiv:2002.05287* (2020).

[29] Yu Rong, Wenbing Huang, Tingyang Xu, and Junzhou Huang. 2019. Dropedge: Towards deep graph convolutional networks on node classification. In *International Conference on Learning Representations*.

[30] Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. 2015. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. *Advances in neural information processing systems* 28 (2015), 802–810.

[31] Chao Song, Youfang Lin, Shengnan Guo, and Huaiyu Wan. 2020. Spatial-Temporal Synchronous Graph Convolutional Networks: A New Framework for Spatial-Temporal Network Data Forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence* 34 (04 2020), 914–921. https://doi.org/10.1609/aaai.v34i01.5438

[32] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903* (2017).

[33] Bao Wang, Xiyang Luo, Fangbo Zhang, Baichuan Yuan, Andrea L Bertozzi, and P Jeffrey Brantingham. 2018. Graph-based deep modeling and real time forecasting of sparse spatio-temporal data. *arXiv preprint arXiv:1804.00684* (2018).

[34] Pengkun Wang, Chuancai Ge, Zhengyang Zhou, Xu Wang, Yuantao Li, and Yang Wang. 2021. Joint Gated Co-attention Based Multi-modal Networks for Subregion House Price Prediction. *IEEE Transactions on Knowledge and Data Engineering* (2021).

[35] Pengkun Wang, Chaochao Zhu, Xu Wang, Zhengyang Zhou, Guang Wang, and Yang Wang. 2022. Inferring Intersection Traffic Patterns with Sparse Video Surveillance Information: An ST-GAN method. *IEEE Transactions on Vehicular Technology* (2022).

[36] Senzhang Wang, Hao Miao, Hao Chen, and Zhiqiu Huang. 2020. Multi-task adversarial spatial-temporal networks for crowd flow prediction. In *Proceedings of the 29th ACM international conference on information & knowledge management*. 1555–1564.

[37] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, and C. Zhang. 2019. Graph WaveNet for Deep Spatial-Temporal Graph Modeling. In *IJCAI*.

[38] Sijie Yan, Yuanjun Xiong, and Dahua Lin. 2018. Spatial temporal graph convolutional networks for skeleton-based action recognition. *arXiv preprint arXiv:1801.07455* (2018).

[39] Bing Yu, Haoteng Yin, and Zhanxing Zhu. 2018. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. In *Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*.

[40] Junbo Zhang, Yu Zheng, and Dekang Qi. 2016. Deep spatio-temporal residual networks for citywide crowd flows prediction. *arXiv preprint arXiv:1610.00081* (2016).

[41] Junbo Zhang, Yu Zheng, Dekang Qi, Ruiyuan Li, and Xiuwen Yi. 2016. DNN-based prediction model for spatio-temporal data. In *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. 1–4.

[42] Qi Zhang, Jianlong Chang, Gaofeng Meng, Shiming Xiang, and Chunhong Pan. 2020. Spatio-Temporal Graph Structure Learning for Traffic Forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 1177–1185.

[43] Qi Zhang, Jianlong Chang, Gaofeng Meng, Shiming Xiang, and Chunhong Pan. 2020. Spatio-Temporal Graph Structure Learning for Traffic Forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence* 34, 1 (2020), 1177–1185.

[44] Ling Zhao, Yujiao Song, Chao Zhang, Yu Liu, Pu Wang, Tao Lin, Min Deng, and Haifeng Li. 2019. T-gcn: A temporal graph convolutional network for traffic prediction. *IEEE Transactions on Intelligent Transportation Systems* (2019).

[45] K Zhou, X Huang, Y Li, D Zha, R Chen, X Hu, et al. 2020. Towards deeper graph neural networks with differentiable group normalization. (2020).

[46] Zhengyang Zhou, Yang Wang, Xike Xie, Lianliang Chen, and Hengchang Liu. 2020. RiskOracle: A Minute-Level Citywide Traffic Accident Forecasting Framework. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 1258–1265.

[47] Zhengyang Zhou, Yang Wang, Xike Xie, Lianliang Chen, and Chaochao Zhu. 2020. Foresee Urban Sparse Traffic Accidents: A Spatiotemporal Multi-Granularity Perspective. *IEEE Transactions on Knowledge and Data Engineering* (2020).

[48] Jiawei Zhu, Yujiao Song, Ling Zhao, and Haifeng Li. 2020. A3T-GCN: Attention Temporal Graph Convolutional Network for Traffic Forecasting. *arXiv preprint arXiv:2006.11583* (2020).