Prediction of Evolution Behaviors of Transportation Hubs Based on Spatiotemporal Neural Network

Mengmeng Chang, Zhiming Ding, Zhi Cai, Zilin Zhao, and Lei Yuan

***Abstract*— With the deterioration of the transportation ecosystem and traffic congestion, traffic graph representations become more complex and lack intelligibility. The deep learning method provides a new way for traffic prediction by mining the spatiotemporal relations of historical states in traffic network. Recurrent neural networks based on gate control can avoid the gradient vanishing and graph convolutional networks provide theoretical support for spatial feature extraction of traffic graph. However, in the prediction of traffic evolution behaviors, the statuses of units and groups distributed in the traffic network have strong and continuous spatiotemporal correlations. Modeling only based on the temporal or spatial perspective usually lacks comprehensiveness and semantic relevance which results in poor prediction performance. In this paper, we propose a spatiotemporal hierarchical propagation graph convolutional network(SHPGCN) to get the whole spatiotemporal evolution features of traffic graph, which can be used to predict the propagation effects and state changes of traffic flow. Considered the inadequacy of graph convolution in the learning of hierarchical features, we construct a low-dimensional propagation graph representation that projects complex node relationships into first-order neighborhoods to capture dynamic changes at different spatial and temporal scales. The SHPGCN uses graph convolution to encode the traffic propagation features and embeds them into a bidirectional recurrent sequence for traffic prediction. The experiment results show that SHPGCN can provide good prediction accuracy and robustness for the traffic evolution behaviors.**

***Index Terms*— Traffic flow prediction, traffic propagation, spatial feature, spatiotemporal neural network, traffic evolution behaviors.**

I. Introduction

THE deterioration of traffic ecology is a common social problem in big city nowadays. According to the INRIX Global Traffic Scorecard, Bogota is the most crowded city in the world in both 2019 and 2020, with commuters wasting an average of 191 and 133 hours per year. More and more time and financial resources are wasted on traffic congestion, causing a series of environmental and social problems. Researchers have further developed traffic flow theory, traffic governance and intelligent transportation system(ITS) to address it. The evolution behaviors of the urban traffic have strong spatiotemporal correlation and predictability. In previous traffic prediction methods, statistical analysis models are mostly limited by the assumption of time series stationarity, and cannot describe the nonlinear nature of traffic flow [1], [2]. Traffic simulation models usually require complex spatial structures and microscopic parameters to accommodate traffic state changes, which limits the modeling ability to complex networks [3], [4]. Nonlinear theoretical models have been widely applied to traffic prediction field and obtained good performance [5]–[8]. The Recurrent Neural Network(RNN) [9] can accurately identify time-series features of traffic data, while Long-Short-Term Memory network(LSTM) [10] overcomes the problem of gradient disappearance during the training of time series. The Convolutional Neural Network(CNN) [11] has outstanding representative learning ability and it can encode traffic network as an image to identify multidimensional features of traffic data [12]. The essence of the traffic network is a graph structure. In order to make full use of the topological features of the traffic graph, the Graph Convolutional Network(GCN) [13], [14] uses the graph spectrum method extending the convolution operation to graph structure and capturing meaningful patterns.

Although the above-mentioned methods have strong modeling ability in traffic flow prediction, there are still some valuable enhancements of the specific traffic optimization. The dynamic traffic network is a complex network composed of a large number of interacting traffic units, which usually exhibit rich behaviors at different spatial and temporal scales. Due to the complexity and variability of traffic context, the analysis of traffic flow from the perspective of spatiotemporal unity needs to be further explored. And the traffic flow has a strong propagation correlation, which is not valued in traditional prediction methods. Furthermore, the hierarchical representation is essential to capture the important features of the traffic graph. The graph convolution method is limited by propagating messages on the edges and cannot be aggregated in a hierarchical manner [15]. It is impractical to characterize the high-latitude evolution of transportation network with fine-grained analysis.

Accordingly, in this paper we describe the evolution behaviors of the traffic graph in a coarse-grained manner by aggregating the low-dimensional features of

Manuscript received 1 September 2020; revised 19 April 2021; accepted 28 May 2021. Date of publication 11 August 2021; date of current version 8 July 2022. This work was supported in part by the Beijing Natural Science Foundation under Grant 4212016 and Grant 4192004, in part by the National Natural Science of Foundation of China under Grant 62072016, and in part by the Project of Beijing Municipal Education Commission under Grant KM201810005023 and Grant KM201810005024. The Associate Editor for this article was S. Siri. *(Corresponding author: Zhiming Ding.)*

Mengmeng Chang is with the School of Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China (e-mail: changmengmeng@emails.bjut.edu.cn).

Zhiming Ding is with the Institute of Software, Chinese Academy of Sciences, Beijing 100190, China, and also with the College of Computer Science, Beijing University of Technology, Beijing 100124, China (e-mail: zmding@bjut.edu.cn).

Zhi Cai is with the College of Computer Science, Beijing University of Technology, Beijing 100124, China.

Zilin Zhao and Lei Yuan are with the Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China.

Digital Object Identifier 10.1109/TITS.2021.3091708

1558-0016 © 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: CTRO.INV. EN MATEMATICAS (CIMAT). Downloaded on October 20,2022 at 19:50:51 UTC from IEEE Xplore. Restrictions apply.

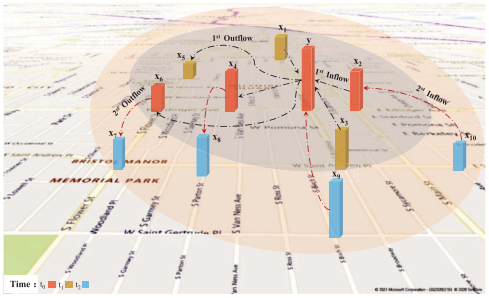

Fig. 1. Coarse-grained description of traffic flow evolution.

traffic flow. As illustrated in Fig.1, the ellipse indicates the propagation scope of the high-traffic node, which is also a coarse-grained clustering. If the traffic volume of transportation hubs (x_2, x_4, x_6) is sufficient at moment t_1, it will continue to propagate and derive the next level of hierarchical clusters. The clustering at the aggregation level of the traffic flow can well describe the evolutionary process of traffic network. Considering the directionality as an implicit feature of traffic flow propagation, we propose to aggregate traffic flow according to its trajectory in the network, which is defined as a coarse-grained description of inflow and outflow. By analyzing the coarse-grained group behaviors in the traffic network, the evolutionary pattern of traffic flow can be effectively mined. The contributions of this paper are mainly as follows.

1) In terms of the spatial feature extraction of traffic graph, the individual correlation of traffic units not only exists between adjacent nodes, but also in the non-adjacent nodes with complex causal relationships, such as the propagation effect of traffic congestion on upstream and downstream. This spatiotemporal correlation has not been fully explored in previous related studies.

2) Compared to the recurrent neural network in which memory states are transmitted along a time series, less research has considered spatial state transmission, and it is not enough to use only static topological features to reflect the spatial patterns of traffic flow. In this paper, the dynamic k-order propagation neighborhood features are extracted and used to capture the continuous variation of spatial states.

3) Interpretable spatiotemporal structure. Considering the semantic hierarchical features of the traffic network, we construct a hierarchical traffic graph based on the propagation domain of transportation hubs with the affiliated nodes projected on the set of hubs, thus transforming the hierarchical features into the hubs' first-order neighborhoods.

The structure of the paper is organized as follows. Section II lists related work. Section III introduces the preparatory knowledge. Section IV describes the model in detail. Section V presents the experiments and analyzes the results. The final section concludes the paper.

II. Related Work

The prediction of the evolution behaviors of traffic network mainly relies on the temporal dynamics and spatial correlation. The spatiotemporal models have been widely used in short-term traffic prediction to effectively model traffic evolution behaviors by capturing the significant spatiotemporal features. However, the potential of spatiotemporal models in terms of network depth, multidimensional feature extraction, data multimodality and predictive capability has not been fully explored in the traffic field. We will describe the progress made by spatiotemporal models for traffic prediction from the following aspects.

Early traffic flow prediction methods were mainly based on recurrent neural networks with temporal dynamics and their improved models. Given that statistical analysis models cannot describe the stochastic and nonlinear nature of traffic flows, Fu *et al.* [16] applied LSTM and Gated Recurrent Unit (GRU [17]) neural networks to predict short-term traffic flow and demonstrated that they outperformed Auto Regressive Integrated Moving Average (ARIMA) model [18]. Cui *et al.* [19] introduced a deep stacked bidirectional and unidirectional LSTM(SBULSTM) considering forward and backward correlation of time series, and applied it to traffic speed prediction. This structure can learn more complex nonlinear temporal correlations. In some LSTM-based enhanced models, the shallow spatial correlation was considered to improve the predictive performance. Zhao *et al.* [20] proposed an LSTM-based traffic prediction model using Origin Destination Correlation (ODC) matrix to characterize the spatiotemporal correlation between different observation nodes. And the cascaded LSTM networks decompose the long-term traffic prediction into individual short-term processes. As a result, the combination of multiple traffic flow features improves the prediction performance.

Benefiting from the outstanding representative learning ability of CNN, Ma *et al.* [12] propose a CNN-based method for traffic speed prediction. By using the traffic network integrity, the spatiotemporal features are transformed into images describing the correlation of traffic units through a two-dimensional coefficient matrix. After that CNN is adopted to extract image features and predict traffic speed on the large-scale traffic network. Yu *et al.* [21] discretized dynamic traffic graph into a series of static images and predicted future traffic using a Spatiotemporal Recursive Convolutional Network(SRCN). The spatiotemporal features are extracted in two parallel stages, with the convolutional layer captured the topological features, and LSTM acquired the temporal features of traffic evolution. Similarly, Liu *et al.* [22] proposed a recurrent network that performs convolution on a fixed multipoint adjacency region. A Conv-LSTM module was formed to extract the spatiotemporal features of traffic flow.

Although recurrent neural networks have achieved prominent development in traffic prediction, graphical network with complex spatial structure cannot yet be directly involved. With the graph-spectral convolution embedded in the recurrent neural network, Seo *et al.* [23] proposed a hybrid Graph Convolutional Recurrent Network (GCRN) to learn spatiotemporal features from graph structure and time-varying data. The GCRN is an extension of RNN on the graph structure. In the research of applying graph convolution to traffic network, Zhao *et al.* [24] proposed a traffic prediction method based on Temporal Graph Convolutional Network(TGCN), which

Authorized licensed use limited to: CTRO.INV. EN MATEMATICAS (CIMAT). Downloaded on October 20,2022 at 19:50:51 UTC from IEEE Xplore. Restrictions apply.

combines GCN with GRU. The GCN module receives topological interactions to capture spatial features, and the GRU module uses fewer gate control units for time series prediction. Considering the variation of the implicit spatial structure of traffic flow, Chen *et al.* [25] proposed a Graph Convolutional network based on LSTM (GC-LSTM) for dynamic link prediction. In the GC-LSTM spatiotemporal feature aggregation process, the GCN learns the spatial structure of each temporal propagation network and the LSTM predicts the future states based on these temporal snapshots. Based on the spatiotemporal multimodal feature fusion, Guo *et al.* [26] proposed an Attribute-augmented Spatiotemporal Graph Convolutional Network(AST-GCN) for traffic flow prediction, modeling three temporal dependence of traffic flow in recent, daily and weekly periodicity. The attention mechanism captures the spatiotemporal correlation of the traffic network, and the graph convolution extracts the spatial features. In the final output, the weighted fusion of three components obtain a better prediction accuracy. Wang *et al.* [27] proposed a Spatial Temporal Graph Neural Network (STGNN) with a learnable positional attention mechanism to better aggregate features from adjacent nodes. Meanwhile, it provided a sequential component modeling traffic evolution with complex topological and temporal dependency.

In the process of spatial features aggregation, there may also be strong spatial correlation between non-adjacent nodes. Luo *et al.* [28] proposed a spatiotemporal traffic flow prediction method combined with K-Nearest Neighbor (KNN) with LSTM. The KNN-LSTM selects the nodes that are most relevant to the observed node by sampling period similarity and uses a two-layer LSTM network to predict the traffic flow with selected nodes separately. Li *et al.* [29] modeled the traffic flow as a diffusion process on directed graph and integrated it into a Diffusion Convolutional Recurrent Neural Network(DCRNN). The DCRNN uses a bi-directional random walk to capture spatial features and an encoder-decoder architecture to capture time dependencies. The representation of spatial dependencies by extended neighborhoods does not consider the features of global similar nodes. To address this problem, Lu *et al.* [30] proposed a Spatiotemporal Adaptive Gated Graph Convolution Network(STAG-GCN). Local and long-term temporal dependencies were captured by acquiring spatial and semantic neighbors of road nodes and combining them with multi-headed attention mechanism. Considering that the limited representation of a fixed graph structure and incomplete adjacency connectivity may limit the effectiveness of the model, Li *et al.* [31] proposed a new Spatial-Temporal Fusion Graph Neural Network(STFGNN) to extract hidden correlations that given spatial graph may not present.

In the deep fusion of spatiotemporal model, temporal variation can be processed in a unified memory pool with spatial appearances. Yu *et al.* [32] constructed a Spatiotemporal Graph Convolutional Network(STGCN) for traffic prediction. Different convolution and recursion units were designed to convert the prediction problem into a spatiotemporal representation. The STGCN completely uses convolutional layers to design neural network for spatiotemporal prediction, making training faster and with fewer parameters. Wang *et al.* [33] proposed a Predictive Recurrent Neural Network(PredRNN) that allows memory states of different units to interact across layers vertically. The PredRNN consists of a stack of Spatiotemporal LSTM (ST-LSTM) cells, which process spatiotemporal representations in a unified memory pool and transfer it both vertically across layers and horizontally over states. The variation in spatial dependency cannot be captured by a fixed Laplace matrix. Diao *et al.* [34] proposed a Dynamic spatial-temporal Graph Convolutional Neural Network(DGCNN) for dynamic Laplacian matrix analysis. The tensor decomposition was used to decompose the traffic flow into a stable global component of long-term spatiotemporal relationships and a local component of short-term fluctuations. Song *et al.* [35] proposed a Spatial-Temporal Synchronous Graph Convolutional Network(STSGCN), which is different from the above-mentioned models that define temporal and spatial components separately, by designing local spatiotemporal graph to connect multiple neighboring spatial graphs together. The graph correlations are extracted using GCN, thus capturing spatial and temporal relationships simultaneously for effective prediction.

Considering that GCN may not be able to adaptively learn depth correlation between topology and node features, hierarchical representation is crucial to capture the structural information of the graph. Current graph convolutional networks applied in traffic prediction do not deeply explore the original hierarchical features through spatial multi-scale. Zhu *et al.* [36] proposed a Knowledge-Driven Spatialtemporal Graph Convolutional Network(KST-GCN) to construct urban knowledge graphs for traffic prediction. The knowledge graph is combined with the traffic network and embedded in GRU to capture the temporal variation of the traffic state. Lv *et al.* [37] added semantic association to the road topology and proposed a Temporal Multi-Graph Convolutional network(T-MGCN). Based on traffic pattern similarity and functional similarity, the association of roads is encoded into multiple graphs, and the dynamic patterns of traffic graphs are captured by combining multi-graph convolution and recurrent neural network.

As mentioned above, considering the integrity of time dependence and spatial correlation, how to deal with the inherent relationship between time and space is the focus of traffic prediction. In this paper, we analyze the traffic flow at different spatial and temporal hierarchies, so as to explore the evolution behaviors of traffic propagation.

III. Preliminaries

A. Traffic Propagation Predictive Learning

Since the traffic flow of the road network intuitively changes regularly with time, many studies have modeled the long-term and short-term traffic flow prediction of network units or groups. However, the traffic flow of the network units not only affect the status of the adjacent nodes, but also propagate to other domains that are quite far away from the topological neighbor. We define this correlation as the propagating domain. It is formed by the high-traffic breaking the equilibrium state of the previous traffic network. These traffic flows propagate through multiple regions and converge at some low-throughput nodes, causing the cross regional impact of traffic flows.

Authorized licensed use limited to: CTRO.INV. EN MATEMATICAS (CIMAT). Downloaded on October 20,2022 at 19:50:51 UTC from IEEE Xplore. Restrictions apply.

TABLE I
THE MAIN NOTATIONS USED IN THE PAPER

Notation	
R, r_i	Road sections set R. $r_i \in R$ with section number i.
$X, x; Y, y$	Node traffic features X, x. Hubs traffic features Y, y .
$N,n; N\downarrow,n\downarrow$	Total sections N,n. Total sections after coarsening $N\downarrow,n\downarrow$.
Hub,h	Transportation Hub. Total hubs h.
T, t	Coarse-grained time slice T. Fine-grained time slices t, $t \in T$.
$A, \tilde{A}, \bar{A}$	Symmetric adjacency matrix A. Normalized adjacency matrix $\tilde{A}$. Propagation adjacency matrix $\bar{A}$.
D,$\tilde{D}$	Diagonal degree matrix D with $D_{ii} = \sum_j A_{ij}$. Normalized degree matrix $\tilde{D}$.
L	Normalized graph Laplacian matrix.
$\Lambda, \tilde{\Lambda}$	Diagonal matrix Λ composed of the eigenvalues of L. $\tilde{\Lambda} = (2/\lambda_{\max})\Lambda - \mathrm{I}_N$, rescaled by the largest spectral radius of L.
S	Traffic assignment matrix.
K	Total number of propagation-related nodes.

We would describe the aggregation nodes in these propagating processes as Hubs. The traffic changes of Hubs can basically represent the evolution behaviors of the entire transportation network. The traffic propagation prediction is based on the historical data of the traffic network to predict the status of Hubs and their spatially affiliated nodes in the sequence. This prediction can be regarded as a spatiotemporal sequence prediction problem.

Consider a directed traffic graph with $G = (R, A, X^T)$. The X^T denotes the discrete time snapshots of the traffic statuses for all road sections within the period T. Let $x_i^t \in X^T$ be the traffic flow of the spatial correlation node r_i and $y_h^t \in X^T$ be the traffic flow of the observer node r_h(Hub) at time t. The y from time t-T to t can be represented as $\{y^{t-T}, \ldots, y^{t-1}, y^t\}$, t and T denote the fine and coarse granularity of time respectively. The propagation vector $\overrightarrow{y^t}$ with spatiotemporal matrix is expressed as follows:

$$\overrightarrow{y^T} = \begin{bmatrix} \overrightarrow{y^{t-T}} \\ \cdots \\ \overrightarrow{y^{t-1}} \\ \overrightarrow{y^t} \end{bmatrix} = \begin{bmatrix} x_i^{t-T} & x_j^{t-T} & \cdots & x_l^{t-T} \\ \vdots & \vdots & \ddots & \vdots \\ x_m^{t-1} & x_n^{t-1} & \cdots & x_o^{t-1} \\ x_p^t & x_q^t & \cdots & x_r^t \end{bmatrix}. \quad (1)$$

where $\overrightarrow{y^t} = < x_p^t, x_q^t, \cdots, x_r^t >$ denotes the propagation sequence of y at time t. To capture the evolution of the traffic propagation, this paper dedicates to learn a function $h(.)$ that maps the historical traffic state of y during T period to its spatiotemporal evolutive state in the future n steps.

$$\left[(y^{t-T}, \cdots, y^{t-1}, y^t) \middle| \overrightarrow{y^T} \right] \xrightarrow{h(.)} \left[y^{t+1}, \cdots, y^{t+n-1}, y^{t+n} \right]. \quad (2)$$

The traffic propagation relationship between nodes can fully characterize the dynamic features of the traffic network. Thus, the future state of the traffic network can be predicted based on the historical traffic propagation.

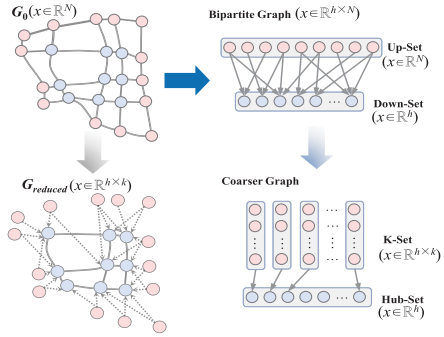

Fig. 2. Fine-grained traffic graph reduced to coarse graph.

B. Construct Transportation Hub Graph With Hierarchical Coarsening

High-traffic in the transportation network usually have a great impact on the capacity of the key roads over a period of time. This influence not only exists between adjacent nodes, but also extends to some non-adjacent nodes that may be only weakly connected, dynamically adjusting by spatiotemporal changes. It was previously considered as a special type of association rule based on sequential frequent pattern mining [38]. Due to the multiplicity of node relationships in the traffic graph, the extraction of frequent itemsets requires plenty of statistical calculations, and it is difficult to adapt the dynamic changes of the rules. Inspired by the time-dependent dynamical similarity of complex networks [39] which uses the trajectory of node state, here we construct a coarsening traffic graph considering the propagation domain to extract the spatiotemporal correlations.

The transportation hubs as an important layer of semantic network, can well express the evolution behaviors of the whole network. As shown in Fig.2, we first transform the fine scale traffic network into a bipartite graph $R = (Up\text{-}Set, Down\text{-}Set)$ with affiliated nodes and Hubs. Afterwards, the *Up-Set* is coarsened on the bipartite graph, which means that the nodes in the *Down-Set* are matched in the *Up-Set*. The *Down-Set* nodes adaptively select k superior nodes and connected edges to form a small-scale coarse graph, while preserving the important features of the original graph. Thus a trainable projection vector p can be used to select k affiliated nodes for projection to the first-order neighborhood of Hub by performing K-max pooling. gPool [40] uses the projection score to select the highest ranked node, obtained by the dot product of p and the features of all nodes. Given the Hub and its feature vector X, Equation 3 describes the coarsening process of X.

$$\begin{aligned} \mathrm{X}^{(T+1)} &= \frac{\mathrm{X}^{(T)} p^T}{\| p^T \|}, \quad \mathrm{idx} = \mathrm{top-rank}(\mathrm{X}^{(T+1)}, [k]) \\ \mathrm{A}^{(T+1)} &= \mathrm{A}^{(T)}_{\mathrm{idx,idx}}. \end{aligned} \quad (3)$$

where $X^{(T)}$ denotes the node feature matrix, $p^{(T)}$ is the scalar projection, and $A^{(T)}$ is the adjacency matrix in period T. The top-rank$(\cdot)$ is the node ranking function, which returns indices of the k-related nodes selected for the coarser graph.

Authorized licensed use limited to: CTRO.INV. EN MATEMATICAS (CIMAT). Downloaded on October 20,2022 at 19:50:51 UTC from IEEE Xplore. Restrictions apply.

The $X^{(T+1)}$ measures the retention information of the T snapshot after projection in the p-direction.

For traffic network, we would like to preserve as many relevant features between nodes as possible. So the top-k most relevant nodes in the projection of $X^{(T+1)}$ are chosen, that is, the projection of each Hub based on the trajectory of its k most relevant *Up-Set* nodes. The associated nodes of Hubs generate a propagation state vector in the network after applying a traffic pulse at period T. After that, the impulse responses are collected into the *K-Set* $Y_k^T = [\overrightarrow{y_1}, \overrightarrow{y_2}, \ldots, \overrightarrow{y_k}]$, and the projection vector P is trained by back-propagation. For each hub, the *K-Set* within its propagation scope provides us with a vector representation of the dynamic traffic network. Nodes with similar propagation vectors in this representation induce a same state in the network at a specific period T. Thus the derived vector space representation of the nodes can be used for the spatial feature learning tasks.

C. Time-Dependent Dynamic K-Order Features Extraction

The receptive field of Hub varies dynamically with time due to the propagation distance and peak traffic, which can be explained by the change of inflow aggregation and outflow diffusion. Thus the receptive field of each Hub is not fixed, that is, they are mainly affected by the k associated nodes under snapshot t (containing inflows and outflows), and may be adjusted at the next time snapshot. The t can be considered as the sampling frequency of the network state within a period T. According to the time dependence of the node state and its k-order correlation, the spatiotemporal receptive field of Hub is defined as follows:

Definition 1 (Time-Dependent k-Order Receptive Field): The spatial associated attributes of Hub are defined by $\bar{\mathrm{A}}^T = \mathrm{idx}[X_1, X_2, \ldots, X_k]^T$, where the component of $\bar{\mathrm{A}}^T$ adjusted with period T.

Frequent traffic state updates are more refined relative to spatial changes, so it is analyzed using fine-grained time ($[t_1, t_n] \in T$). The traffic flow equation of Hub is:

$$\mathrm{Y}^{(t+1)} = S^t \bar{\mathrm{A}}^T = \xi_1^{(t)} X_1^T + \xi_2^{(t)} X_2^T + \ldots + \xi_k^{(t)} X_k^T. \quad (4)$$

where $S^t = [\xi_1, \xi_2, \ldots, \xi_n]^t$ are the assignment matrix of t, i.e., the Hub state at $t+1$ is determined by the inflow and outflow coefficients from t. Each row of S^t corresponds to the set of correlation coefficients in the receptive field on which Hub depends, and each column of S^t indicates the traffic contribution of the affiliated node to one Hub(if the node is not in the receptive domain of the Hub, $\xi = 0$).

In this way, we can more accurately predict the evolution behaviors of Hubs through its inflow and outflow. To capture the spatiotemporal correlation between Hubs and affiliated nodes, we use a graph pooling method [15] to learn the assignment matrix within each time t. A learned assignment matrix S^t contains the coefficient being assigned to the next traffic state X_{hub}^{t+1}, which generates a new traffic graph, as follows.

$$\mathrm{X}_{hub}^{(t+1)} = S^t \bar{\mathrm{A}}^T X_{\mathrm{idx}}^T \in \mathbb{R}^{n^{(t+1)} \times h}. \quad (5)$$

Equation 5 uses the embedding X^T in coarse-grained T and adjusts them to X_{hub} with the allocation matrix S in fine-grained t. The $\bar{\mathrm{A}}^T$ is calculated by the projection of the period T to obtain the next $\bar{\mathrm{A}}^{(T+1)}$. In contrast, S^t can be generated using the receptive field $\bar{\mathrm{A}}^T$ and X^t.

$$S^t = Softmax(Relu(\overline{D}^{-\frac{1}{2}} \overline{A}^T \overline{D}^{\frac{1}{2}} X^t W^t)). \quad (6)$$

where $\overline{D} = \sum_j \overline{A}_{ij}$ and W^t is the trainable weight matrix. The Softmax is a probability distribution mapping function. And ReLU is applied to the convolution filter of the propagation graph.

We construct the above k-order features aggregation process as a propagation layer. After training in the propagation layer, the spatiotemporal correlations between Hubs and affiliated nodes in the time-slice are obtained. It embeds the impact of the inflow and outflow from the receptive field of Hubs into their first-order neighborhood, so the output of the propagation layer can be used to graph convolutional network.

IV. The Framework of SHPGCN

In this study, we propose a spatiotemporal hierarchical propagation graph convolutional network named SHPGCN for traffic flow prediction. Fig.3 shows the framework of SHPGCN. In the input part of the model, the hierarchical spatial features in the traffic network build a spatiotemporal transform through the propagation layer, and aggregate their propagation neighborhoods through the graph convolution layer to encode into the potential feature space. After that, the spatiotemporal coding uses bidirectional LSTM to achieve sequence features, and make the overall features of the current network to transform to the next moment. And the spatiotemporal feature transformations are fused into the unified memory pool, containing both temporal and spatial state conveys. Finally, the output of the sequence unit is decoded to reconstruct the original data space of the hubs in the network, or as input for the next step prediction.

A. Spatial Graph Feature Aggregation

Considering the inapplicability of time series models in heterogeneous graph, graph convolution can extract meaningful features from spatial patterns. It process the graph features as the product $\Theta * x = U\Theta(\Lambda)U^T x$ of the signal $x \in \mathbb{R}^N$ and the filter $\Theta = diag(\theta)$ parameterized by $\theta \in \mathbb{R}^N$ [13], [41]. The eigenvalue decomposition of the Laplacian matrix for large graph is computationally expensive with $O(N^2)$ so the Kth-order truncation of Chebyshev polynomial $T_k(x)$ was used to approximate $\Theta(\Lambda)$ [42]:

$$\Theta * x \approx \sum_{k=0}^{K} \theta_k U T_k(\tilde{\Lambda}) U^T x. \quad (7)$$

where $\theta \in \mathbb{R}^K$ denotes the Chebyshev coefficient. By using this K-localized convolution, the relevant nodes in the graph can be converted into k-order neighborhood with the complexity of $O(K|\varepsilon|)$ [41]. And Kipf *et al.* [13] further simplified the convolution with the first-order Chebyshev approximation(GCN) to alleviate the overfitting on local neighborhood structures as follows:

$$\Theta * x \approx \theta(\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{\frac{1}{2}}) x. \quad (8)$$

Authorized licensed use limited to: CTRO.INV. EN MATEMATICAS (CIMAT). Downloaded on October 20,2022 at 19:50:51 UTC from IEEE Xplore. Restrictions apply.

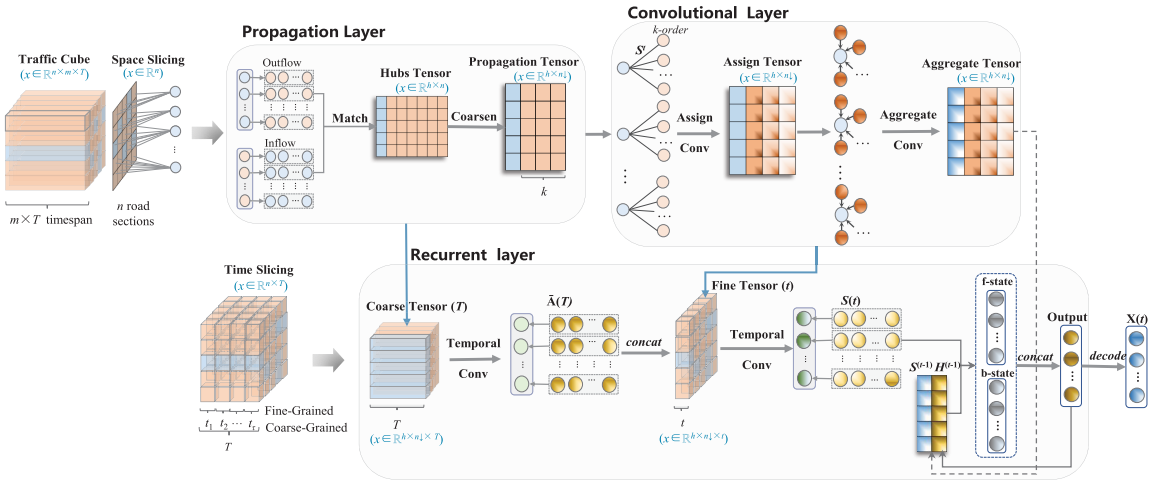

Fig. 3. The SHPGCN framework.

The node state in the traffic graph can be considered as particular instances of filtering in the vertex domain, where it can be computed from its corresponding k-order neighborhood [43]. Each node's filter assigns weights in its k-neighborhood, thus the GCN offers an efficient architecture to aggregate the propagation features of Hubs and encode the graph snapshots. Specifically, the input of traffic prediction is composed of m frame of traffic graphs as in Fig.3.

Each spatial slice $x \in \mathbb{R}^n$ is composed of the traffic features of n sections, and $n \downarrow$ are extracted Hubs. The k-order neighborhoods of each Hub is computed by the propagation layer so that the GCN can further aggregate it. We join the normalization propagation neighborhood $\widetilde{A}$ and its dependent degree matrix $\tilde{D}$ to define the convolutional layer as:

$$H_a = \sigma_a(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{\frac{1}{2}}\mathrm{X}W). \tag{9}$$

The $\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{\frac{1}{2}}$ calculates the convolution filter of the coarsened network. The σ_a is the ReLU activation function for performing the first-order spectral convolution and $H^S \in \mathbb{R}^{n\downarrow}$ is the output of the convolutional layer for the spatial slice. The convolutional layer finally converts the input traffic tensor $X^t = (x_1^t, \ldots, x_{i-1}^t, x_i^t)$ into a code $H_S^t = (h_1^t, \ldots, h_{i-1}^t, h_i^t)_S$. The SHPGCN uses the convolutional layer to extract the spatial features of propagation-related nodes in the graph snapshot. In the following section, we built a coding and decoding framework by integrating convolutional layer and recurrent layer to predict the evolution of traffic state in time and space.

B. Spatiotemporal Recurrent Memory Pool

Considering the coarse granularity of propagation changes and the fine granularity of traffic state changes, we use two different granularities of slicing in recurrent layer, as shown in the lower layer of Fig.3. Coarse-grained time slice $x \in \mathbb{R}^{n\downarrow\times T}$ is used to capture the time-varying features of the propagation, while fine-grained time slice $x \in \mathbb{R}^{n\downarrow\times t}$ is used to capture the traffic state change($t \in T$). A time slice in X^t can be regarded as a matrix whose column is the Hub

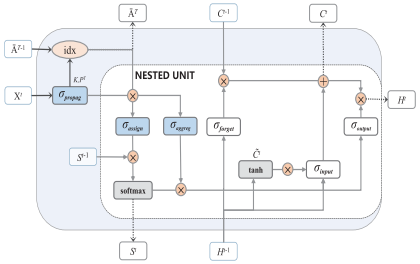

Fig. 4. The spatiotemporal memory cell of SHPGCN.

dimension, and row is the number of associated nodes k. For each time step T, the convolution of Hub neighborhoods is performed in a coarse-grained level for temporal feature extraction, and similarly, in time step t, the assigned weights of the neighborhoods are fitted in a fine-grained level. The neighborhood states of Hubs are finally aggregated into a spatiotemporal snapshot. In this process, the propagation and convolutional layers are fused in a nested form into recurrent layer, and the spatiotemporal representation is encoded and passed layer by layer. Through layer-by-layer convolution at coarse and fine granularity, the hidden feature representations are filtered and refined from outside to inside, resulting in more reasonable features.

The prediction of spatiotemporal sequences is different from the prediction of time series. The target of SHPGCN prediction is to extract the time-series features of the spatiotemporal coding. Both the spatial situation and the tidal effect of time make the changes of traffic flow. And the spatiotemporal traffic state is not only related to the previous state but also related to the future state. Besides, in the prediction of the evolution of traffic flow, the propagation of traffic flow may be driven by multiple hubs at the same time. And if the propagation of multiple hubs has overlapping neighbors, the original spreading path will be enhanced or blocked.

The bidirectional LSTM memory [19] contains a forward and a backward state connected to the same output, which can take into account both the previous and future context.

Authorized licensed use limited to: CTRO.INV. EN MATEMATICAS (CIMAT). Downloaded on October 20,2022 at 19:50:51 UTC from IEEE Xplore. Restrictions apply.

Combining the bidirectional LSTM, we propose a spatiotemporal recurrent architecture with hierarchical memory pool. It consists of 2 basic units nested to increase the depth of contextual memory of SHPGCN. As shown in Fig.4, the layer-by-layer variation of spatiotemporal status is calculated by introducing different state gates to control and enhance the memory ability of the model. In the outer layer of the memory pool, k-neighbor selection is performed by introducing propagation gate. Then it enters into the nested unit to adjust the contribution of the neighborhood by assign gate and calculate the spatial state S. The aggregation gate encodes the states of the neighborhoods and feeds them into the forward and backward temporal gate control units. The temporal control unit processes the input through the forget gate and input gate. Finally, the output gate is introduced as the output of the SHPGCN and used for the next spatiotemporal memory cell.

The SHPGCN cell takes X^t as input, and after processing by the internal memory pool will generate a hidden state H^t as the input of the next cell. Inside the memory cell, the spatial context and the forward time series use the forward sequence, which means the output of t-1 step as the input for t step. The backward time series use the output of $t+1$ step as the input for t step. At time step t of T, SHPGCN cell need to input the propagation vector $\overline{A}^{T-1}$, spatial assign vector S^{t-1}, cell state C^{t-1} and temporal hidden state H_{t-1} (including $\overrightarrow{h}_t$, $\overleftarrow{h}_t$ by forward and backward) of the previous cell. The X^t is the input of the propagation gate to obtain k-correlative neighborhood index idx, and then combines with the previous spatial encoding S^{t-1} to get S^t. The activation function of the gate update in the propagation layer is shown in Equation 10:

$$\begin{aligned} P^t &= \sigma_{propag}(W_p * \mathrm{idx}(\mathrm{X}^t) + B_p), \\ S^t &= \sigma_{assign}(W_s * \overline{D}^{-\frac{1}{2}} \bar{\mathrm{A}}^t \overline{D}^{\frac{1}{2}} [P^t, S^{t-1}] + B_s). \end{aligned} \tag{10}$$

The $idx(.)$ function locates the spatial position in X^t and extracts the k-th correlation vector at time t. After that, GCN is used to aggregate the propagation features of S^t to obtain X'_t which is generated through the aggregate gate.

$$\mathrm{X}'_t = \sigma_{aggreg}(W_a * \tilde{D}^{-\frac{1}{2}} \tilde{\mathrm{A}} \tilde{D}^{\frac{1}{2}} S^t + B_a). \tag{11}$$

Afterward, the spatial encoding is combined with previous spatial hidden state H^{t-1} into the temporal gate units. The forget gate, input gate and output gate in the forward and backward sequences respectively delete, add and filter the input spatial code X'_t. The activation function of the gate update in the recurrent layer is shown in Equation 12:

$$\begin{aligned} F_t &= \sigma_{forget}(W_f[X'_t, H_{t-1}] + B_f), \\ I_t &= \sigma_{input}(W_i[X'_t, H_{t-1}] + B_i), \\ O_t &= \sigma_{output}(W_o[X'_t, H_{t-1}] + B_o). \end{aligned} \tag{12}$$

where W_f, W_i, W_o are the weight matrix of the input vector to the gate, B_f, B_i, B_o are the corresponding bias. And σ_{forget}, σ_{input}, σ_{output} are the Sigmod activation function.

The status update of $\tilde{C}_t$ depends on the activation function of X'_t and hidden state H_{t-1} in the Equation 13, where W_c is the weight matrix between input vector and hidden vector, and B_c is the state equation bias.

$$\tilde{C}_t = tanh(W_c[X'_t, H_{t-1}] + B_c). \tag{13}$$

The input gate and forget gate are used to update the cell state C_t, and the output H_t is controlled by the output gate and the activation unit, as in Equation 14:

$$\begin{aligned} C_t &= F_t \odot C_{t-1} + I_t \odot \tilde{C}_t, \\ H_t &= O_t \odot tanh(C_t). \end{aligned} \tag{14}$$

where $\odot$ is Hadamard procuct. The output H_t is calculated from the spatial encoding X'_t, so it needs to be decoded. The final output Y_t is composed of the forward $\overrightarrow{h}_t$ and backward $\overleftarrow{h}_t$, which ensures that the prediction results are integrated with the spatial and temporal contexts of the traffic environment. The Equation of the output of the SHPGCN cell as follows:

$$\mathrm{Y}_t = \sigma_d(\cup[\overrightarrow{h_t}, \overleftarrow{h_t}] + B_d). \tag{15}$$

where σ_d denotes the $tanh$ activation function, U denotes the concat operation, and B_d is the output bias.

From the above, we can take the traffic cube at time t of the road network as input. The SHPGCN memory unit uses $\overline{A}^{T-1}$, S^{t-1}, C^{t-1} and H^{t-1} as the prior knowledge for prediction. The final prediction Y_t is obtained through the SHPGCN spatiotemporal memory unit processes. The SHPGCN sequences extract highly abstract features layer-by-layer and then make predictions by mapping them back to a unified spatiotemporal transform domain. This prediction result effectively fuses the history and future spatiotemporal knowledge, and shows high correspondence to their real evolution.

V. Performance Evaluation

All experiments are compiled and tested on a Windows10 system with 64GB memory (CPU: Intel(R) Xeon Silver 4210CPU@2.20GHz, GPU: NVIDIA GeForce RTX 2080 Ti). We validate the model on two actual traffic datasets from PeMSD and Didi.[1]

Data Processing: The PeMSD dataset collects real-time traffic flow, occupancy, and average speed data for major highways in California, USA from the Caltrans Performance Measurement System (PeMS). The processed PeMSD04 dataset contains 16992 traffic records collected during the 59-days (from Jan 1st to Feb 28th in 2018). Each record contains the traffic flow of 307 sensors, with a sampling interval of 5 minutes. The Xi'an dataset, from the Didi Chuxing GAIA Initiative, contains average travel speeds, road attributes, and network topology at the road level within Xi'an's Second Ring Road. The dataset intercepted 8496 speed records collected from 792 nodes from Jan 1st to Feb 28th in 2018. The data are discretized as one record every 10 minutes, and we consider the road state to be uniform within 10 minutes.

The adjacency matrix of 2 datasets is calculated based on the downstream connection between the nodes. Missing values are filled by the linear interpolation and the topological adjacency

[1] gaia.didichuxing.com

Authorized licensed use limited to: CTRO.INV. EN MATEMATICAS (CIMAT). Downloaded on October 20,2022 at 19:50:51 UTC from IEEE Xplore. Restrictions apply.

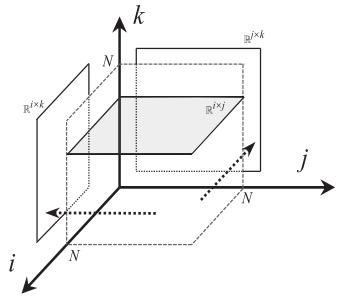

Fig. 5. Propagation vector space.

average(missing range>60 minutes) in both of those datasets. For the low fidelity values, the outliers are identified and deleted, and then filled according to the missing values.

Extraction of Hubs: We construct a subgraph composed of shortest path trajectories called the propagation graph $G_p \in G$, G_p takes the Hubs in the traffic network as the source and gradually expands its neighbors according to the traffic propagation direction (inflow and outflow). The propagation betweeness [44] of G_p reflects the contribution degree of one node as a bridge from Hub to the others or in reverse.

Given $p(r_i, r_i)$ denotes the shortest path from r_i to r_j, $p(r_i, r_i|r_k)$ is the shortest path of $p(r_i, r_i)$ through r_k, and r_i, r_j, $r_k \in \boldsymbol{R}$ and each other not equal, φ is the statistics of paths. The propagation betweeness in G_p is:

$$C\left(r_i, r_k, r_j\right) = \frac{\varphi\left(p\left(r_i, r_j \mid r_k\right)\right)}{\varphi\left(p\left(r_i, r_j\right)\right)}. \tag{16}$$

The C is a value between 0 and 1 that bigger the value means stronger its propagation receptive capacity. The spatial affiliated nodes of Hubs are measured by the propagation betweeness. Fig.5 shows the spatial representation of the propagation graph $\mathbb{R}^{i\times k\times j}$, the elements of this vector space $C(r_i, r_k, r_j) \in \mathbb{R}^{i\times k\times j}$ represent the propagation effects of r_k on $<r_i, r_j>$. The matrix corresponding to the observation plane $\mathbb{R}^{i\times j}$ along the k-axis is a symmetric matrix. Projection of the space vector to the j-axis leads to the conversion of the propagation betweenness of r_i into $C(r_i, r_k) = \sum_{r_j \in G_p, j\neq i\neq k} \frac{\varphi(p(r_i, r_j|r_k))}{\varphi(p(r_i, r_j))}$.

idx(): For the zero state of each Hubs, a propagation graph with Hubs as the destination(inflow) or as the origin(outflow) is adopted. We perform BFS traversal for each Hub and compute the propagation betweeness of its affiliated nodes separately, and finally construct a row vector of $X \in \mathbb{R}^{h\times k}$ based on the position of the largest k affiliated nodes.

Experiments Settings: We take 1 hour as the training window, and the target is to predict the traffic state of Hubs in the next 20,40 and 60 minutes. We use the first 80% of the 2 datasets as the training data, and the remaining 20% as the validation data. In the training process of SHPGCN, we use Huber loss to measure the error between the actual value x_t with their predictive value y_t. The model uses Adam optimizer by minimizing the loss function for 2000 iterations with huber loss parameter $\delta = 1.35$. We choose 4 kinds of prediction errors to evaluate the performance of the model, Mae, Rmse, R-square, and Accuracy, as in Equation 17:

$$\begin{aligned} Mae &= {}^{1}\!/_{n} \sum_{t=1}^{n} |y_t - x_t|, \\ Rmse &= \sqrt{{}^{1}\!/_{n} \sum_{t=1}^{n} (y_t - x_t)^2}, \\ R{-}square &= 1 - \frac{\sum_{t=1}^{n} (y_t - x_t)^2}{\sum_{t=1}^{n} (x_t - \bar{x})^2}, \\ Accuracy &= 1 - \frac{|y_t - x_t|}{x_t}. \end{aligned} \tag{17}$$

where $\overline{x}$ denotes the average of the real traffic state. And n is the prediction step length. Smaller Mae, Rmse and larger R-square, Accuracy reflect better performance of the model.

A. Comparison With Baseline Methods for Traffic Flow Prediction

The experiment compares the performance of our proposed SHPGCN model with the following baseline methods training on PeMSD04 and Xi'an datasets respectively:

- BILSTM [19]: A bidirectional LSTM network that takes into account the forward and backward dependencies of time series data.
- GRU [16]: GRU is a simplified variant of LSTM with a new type of hidden layer unit.
- ChebyGCN [41]: Fast K-localized convolutional neural network on graph use Chebyshev polynomial approximation. The initial kernel size of graph convolution is set to 3.
- TGCN [24]: A temporal graph convolutional network combining GCN and GRU.
- GC-LSTM [25]: A graph convolution network using GCN embedded LSTM for dynamic graph prediction.
- KNN-LSTM [28]: A spatiotemporal traffic flow prediction method combined with k-nearest neighbor (KNN) and LSTM.
- STGCN [32]: A fully convolutional network integrating graph convolution and gated time convolution. The convolution kernel is set to 3.
- PredRNN [33]: A predictive recurrent neural network memorizing spatial appearances and temporal variations in a unified pool.

TABLE II and TABLE III show the Mae and Rmse of all prediction methods in the future 20, 40 and 60 minutes.

It can be seen that the Mae and Rmse of the SHPGCN on the 2 datasets are better than the time series prediction models GRU, BILSTM, graph convolution model ChebGCN, and spatiotemporal prediction models TGCN, STGCN, PredRNN, GC-LSTM, KNN-LSTM. The time series model focuses on extracting sequence features to predict the future traffic state. The GRU performs slightly better than BILSTM in dataset 1, while they perform essentially the same in dataset2. The KNN-LSTM adds the screening of correlation nodes and further fuses spatial features based on time series prediction, so its performance (Mae, Rmse) is better than

Authorized licensed use limited to: CTRO.INV. EN MATEMATICAS (CIMAT). Downloaded on October 20,2022 at 19:50:51 UTC from IEEE Xplore. Restrictions apply.

TABLE II

PERFORMANCE OF THE PREDICTION METHODS AT DIFFERENT TIME LENGTHS ON THE PEMSD04

Time Length	20 *minutes*		40 *minutes*		60 *minutes*	
Metric	Mae	Rmse	Mae	Rmse	Mae	Rmse
BILSTM	21.476	33.820	24.584	37.788	27.998	42.625
GRU	21.303	33.418	24.458	37.646	27.545	41.949
KNN-LSTM	21.658	33.232	24.063	37.002	26.789	40.831
GC-LSTM	21.501	33.475	24.555	37.664	27.461	42.069
TGCN	21.297	33.447	24.486	37.711	27.644	41.983
ChebGCN	20.494	32.1687	23.032	35.895	25.814	39.927
STGCN	19.877	31.419	22.270	34.595	22.398	34.714
PredRNN	19.554	31.073	22.0662	33.563	22.963	36.984
SHPGCN	**19.188**	**30.171**	**21.236**	**32.569**	**22.082**	**33.835**

TABLE III

PERFORMANCE OF THE PREDICTION METHODS AT DIFFERENT TIME LENGTHS ON THE XI'AN DATASET

Time Length	20 *minutes*		40 *minutes*		60 *minutes*	
Metric	Mae	Rmse	Mae	Rmse	Mae	Rmse
BILSTM	4.48	8.102	4.920	8.681	5.173	8.977
GRU	4.493	8.195	4.889	8.624	5.224	9.002
KNN-LSTM	4.501	8.081	4.867	8.483	5.190	8.847
ChebGCN	4.439	8.179	4.885	8.627	5.223	9.052
GC-LSTM	4.420	8.107	4.874	8.607	5.182	8.949
TGCN	4.426	8.106	4.846	8.600	5.169	8.958
STGCN	4.394	8.080	5.268	8.949	5.109	8.977
PredRNN	4.435	8.088	4.871	8.448	5.265	8.679
SHPGCN	**4.315**	**7.949**	**4.838**	**8.411**	**5.036**	**8.541**

GRU(24.43, 37.67), BILSTM(24.68, 38.08) with the average (24.17, 37.02) in three prediction periods on dataset1. However, it utilizes KNN for spatial feature selection with long time similarity (daily average), which is separated from the training of LSTM. It leads to a poor performance in short time prediction compared to the graph convolution model in 20 minutes. As the prediction step increases, the long-time based spatial correlation is highlighted and its performance is gradually improved in 40 and 60minutes predictions.

Obviously, the graph convolution has good results, as can be seen from the performance of ChebGCN that uses it exclusively. Its performance is better than that of the time series model. The TGCN and GC-LSTM are combined with GCN, which is implemented with the layer of the general adjacency matrix. Their spatial fitting ability are completely dependent on the Laplace matrix. However, since the Laplace matrix is constant and time-independent, it leads to the superposition of redundant spatial features during training. Their Mae and Rmse are higher than the performance of more complex spatiotemporal model.

The spatiotemporal module in STGCN combines graph convolution and gated temporal convolution to extract the most useful features, and performs well on dataset1. As can be seen from TABLE III, the Chebyshev prediction ranking in dataset2 has dropped relative to TABLE II. Since STGCN uses ChebyGCN for spatial convolution, its performance has also been affected. For PredRNN, its memory state is allowed to across stacked RNN layers vertically. So, it has stronger spatiotemporal memory and performs well on 2 datasets. But for

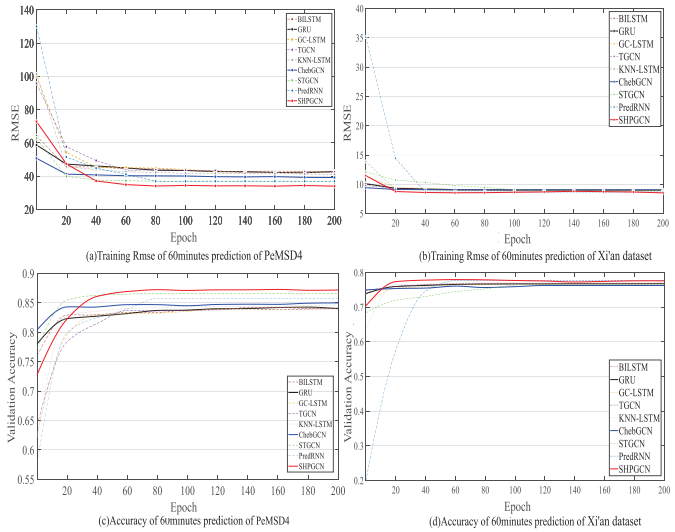

Fig. 6. Training Rmse and validation Accuracy change for first 200 epochs.

traffic prediction, PredRNN needs to process the traffic map as an image and then computes the neighborhood convolution between the image rasters. The neighborhood convolution between rasters does not involve spatial correlation between non-adjacent nodes.

In contrast, SHPGCN uses the propagation layer to further process spatial features and uses bidirectional recurrent sequences to avoid the negative impact of spatiotemporal irrelevance. Compared with the classical models, the average Mae, Rmse reduction of SHPGCN over GRU is (-14.31%, -14.18%) on dataset1 and (-2.94%, -3.69%) on dataset2. Similarly, compared with ChebGCN, the reduction is (-9.54%, -10.24%) and (-2.52%, -3.84%) respectively. Compared with the complex spatiotemporal models, SHPGCN also has (-3.17%, -4.12%), (-4.10%, -4.43%) than STGCN, and (-3.16%, -4.79%), (-2.69%, -1.26%) than PredRNN in three prediction periods. In particular, SHPGCN shows the best Mae:19.188 and Rmse:30.171 on dataset1 for the 20-minute prediction. For dataset2, the performance of SHPGCN is consistent with dataset1. The prediction ability of SHPGCN at 20 minutes is better than that at 40 and 60 minutes, indicating that the prediction performance decreases as the prediction scope expands.

B. Training Performance and Validation of Different Models

Considering that SHPGCN is a network based on the fusion of spatiotemporal features, we verified its ability to fit temporal variations and spatial appearances. Fig.6 shows the changes in training Rmse and validation Accuracy over the 60minutes prediction on dataset1 and dataset2. In the first 20 epochs of Fig.6(a), the maximum error of the training Rmse is PredRNN. This is because the memory state of PredRNN is not limited to a single LSTM unit, and the stacked units are spatially correlated with each other leading to difficulty in training. The TGCN, GC-LSTM and KNN-LSTM descent curves are basically the same, and the reason for the larger error compared to the time series model is that redundant spatial features are

Authorized licensed use limited to: CTRO.INV. EN MATEMATICAS (CIMAT). Downloaded on October 20,2022 at 19:50:51 UTC from IEEE Xplore. Restrictions apply.

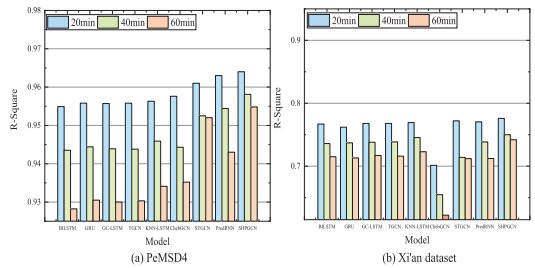

Fig. 7. The R-square change in different predicting periods.

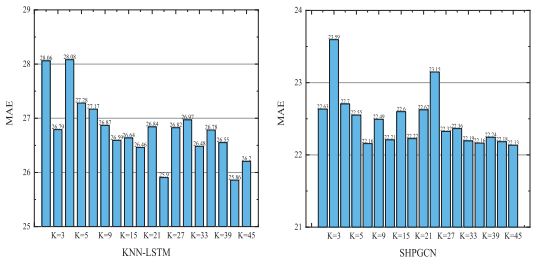

Fig. 8. Effects of k on the performance of the model.

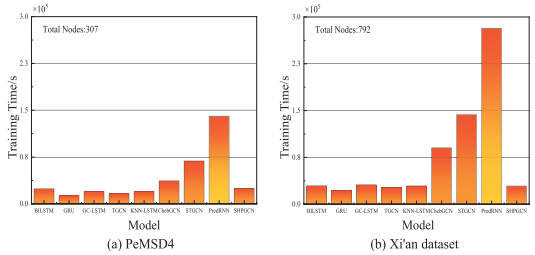

Fig. 9. Training time of different models.

superimposed in the initial phase of training. The performance of the time series models GRU and BILSTM stabilized after the first 40 epochs.

The ChebyGCN uses a deep convolutional kernel to achieve a low error with the smallest iteration step of all models and remains stable. The curves of STGCN and our SHPGCN are initially between general spatial convolution and deep convolution. On the one hand, shallow convolution has insufficient ability to capture spatial features, while deep convolution requires a larger reception field difficult to compute. It can be seen that STGCN outperforms ChebyGCN after 20 epochs and SHPGCN after 40 epochs. The error of PredRNN also levelled off after 40 epochs and reached a level comparable to STGCN at 80 epochs.

It has a similar effect in Fig.6(b), where the differentiation is not significant because the Xi'an dataset uses speed as a measure of traffic state, but the complex spatiotemporal models still maintain the lead. This is since STGCN, PredRNN and SHPGCN all use a dynamic mechanism to memorize spatial features and train better than shallow graph convolution and temporal convolution. The SHPGCN uses propagation vectors to dynamically adjust the spatial mapping and further stratify to refine more important spatial features, thus improving the accuracy of prediction. From the accuracy

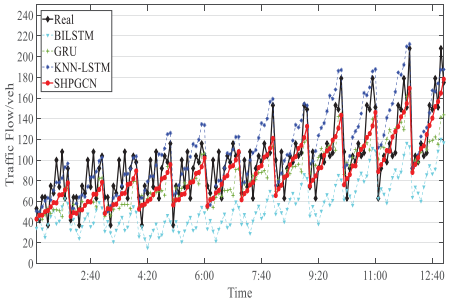

Fig. 10. The traffic flow prediction of time sequence models.

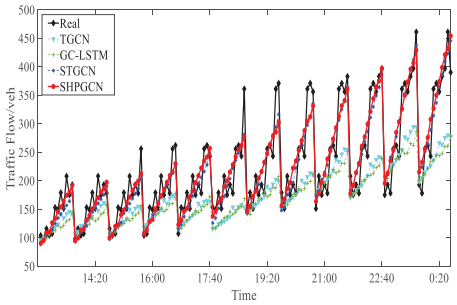

Fig. 11. The traffic flow prediction of spatiotemporal models.

variations in Fig.6(c) and Fig.6(d), we can see that SHPGCN has achieved better accuracy than the general spatiotemporal model in the initial stage, indicating that the spatially relevant features of Hubs can predict their states well. The Accuracy of SHPGCN exceeds that of ChebyGCN at 40 and 20 epochs, respectively, which can indicate that the propagation features outperform the 3-order neighborhood features. And by feature layering and multi-granularity nested memory, SHPGCN still has comparable prediction accuracy compared with complex spatiotemporal models.

To compare the performance of the models in different prediction periods, we examine the R-square at different prediction ranges of 20, 40 and 60 minutes. As shown in Fig.7, SHPGCN has higher R-square in all three prediction periods than other models on 2 datasets. The performances of BILSTM and GRU are approximately the same because the time series model has good fitting ability to smooth series. The TGCN, GC-LSTM and KNN-LSTM equally matched at 20minutes, and the R-square of KNN-LSTM gradually increased with time spans, corresponding to the adoption of long-time spatial similarity. The ChebyGCN outperforms TGCN, GC-LSTM in dataset1, indicating that deep graph convolution outperforms the first-order convolution. However, the performance of ChebyGCN in dataset2 is poor, indicating that the neighboring features do not improve significantly with the expansion of the neighborhood depth. The STGCN is affected by ChebyGCN and has a reduced performance in the 40, 60minutes prediction in dataset2. The PredRNN and SHPGCN can still maintain stable performance in dataset2 with insignificant spatial features.

1) The Optimal Choice of k: The k-neighborhood represents the spatiotemporally related nodes that are upstream and downstream of Hub and are not necessarily neighboring.

Authorized licensed use limited to: CTRO.INV. EN MATEMATICAS (CIMAT). Downloaded on October 20,2022 at 19:50:51 UTC from IEEE Xplore. Restrictions apply.

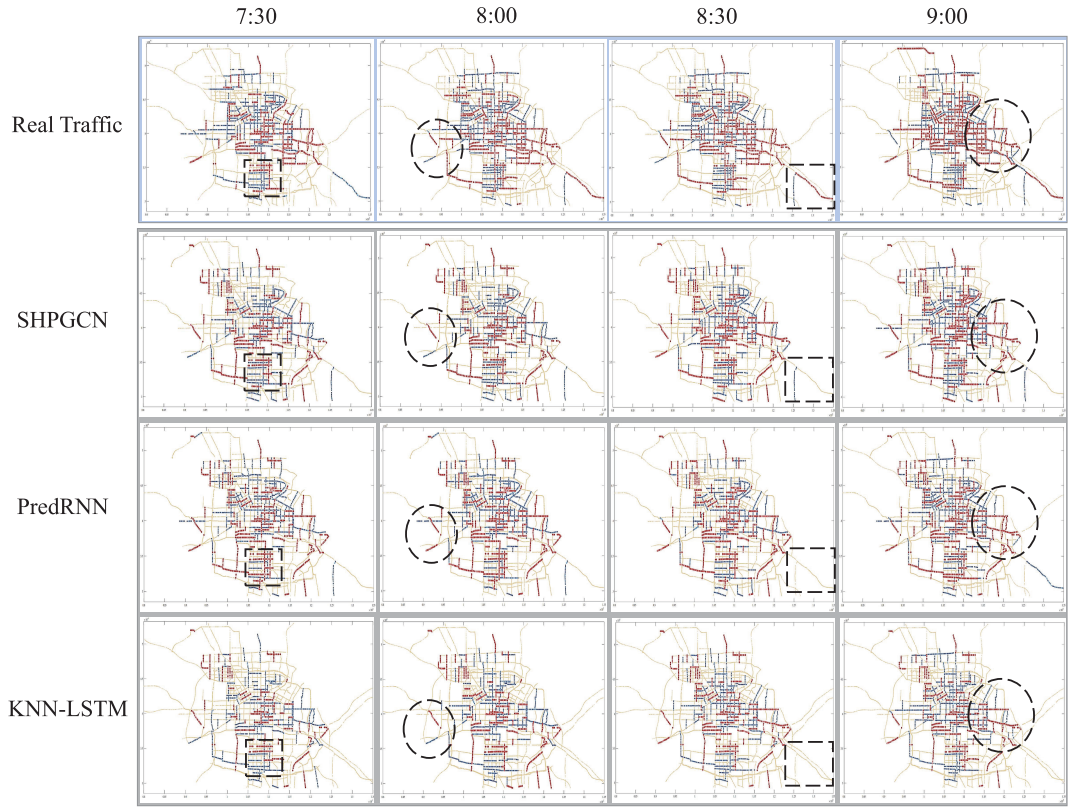

Fig. 12. Dynamic traffic graph prediction of Xi'an Feb 17, 2018.

We conduct experiments based on KNN-LSTM and SHPGCN with different K to illustrate its effect on performance, respectively. As shown in Fig.8, the performance of KNN-LSTM improves as K increases. It reaches a relatively good performance when $K = 24$. The Mae becomes larger when K continues to increase, which is because the correlation between Hub and affiliated nodes decreases when the K is too large, resulting in a decrease in performance. By contrast, our SHPGCN uses a trainable feature selection process, and theoretically the model performs best when K takes the maximum (the total number of network nodes). However, as the number of k-nodes increases, the contribution of the lower ranked nodes gradually becomes smaller. The model performance does not improve significantly after $K \geq 27$. Therefore, it is reasonable to choose K at which the model tends to be stable.

2) Model Training Time: We compared the training time of each model in the same experimental setting, and the elapsed time is shown in Fig.9. It can be seen that the GRU with simplified gate unit has the shortest time on both datasets. While BILSTM for bidirectional sequences, KNN-LSTM using similarity clustering, and shallow graph convolution GC-LSTM, TGCN, the training time of these models are controlled in the range of $4 \times 10^4 s$. For ChebyGCN and STGCN,the time cost of performing K-localized convolutional operation is costly. The PredRNN uses a more complex spatiotemporal stacked LSTM structure and adds spatial vertical propagation between units, so its training time is the largest. The SHPGCN uses the hierarchical division of semantic road networks to extract spatial features, which alleviates the computational cost of complex graph convolution.

C. Comparison of Traffic Evolution Prediction of Different Models During Traffic Peak

To clearly show the evolution process of traffic flow of Hubs, we select the variation curves of a specified observation node in the 20minutes prediction of PEMS04 on February 18, 2018. As shown in Fig.10, we compare the performance of BILSTM, GRU, KNN-LSTM and our SHPGCN under the morning peak, the black curve is the real value and the red curve is the SHPGCN predicted value. It can be seen that the predicted value of BILSTM is below the real value, which is consistent with the performance in TABLE II. The curve of GRU captures the fluctuations of the real value well, but the fit is poor. The KNN-LSTM fitting curve fluctuates widely because the superposition of redundant spatial features increases the amplitude of the predicted flow.

In terms of spatiotemporal model prediction, we choose TGCN, GC-LSTM, STGCN and SHPGCN to compare flow variation trend under the evening peak, as shown in Fig.11. These models both take into account the spatiotemporal multi-dimensional features of the traffic flow, and it can be seen that they all perform relatively well, with no dramatic deviations in the prediction curves. The STGCN and SHPGCN extracted deep neighborhood features by Chebyshev and propagation graph convolution, respectively, and the performance matched best. The TGCN and GC-LSTM use shallow GCN network, which is less capable of nonlinear expression than SHPGCN.

Authorized licensed use limited to: CTRO.INV. EN MATEMATICAS (CIMAT). Downloaded on October 20,2022 at 19:50:51 UTC from IEEE Xplore. Restrictions apply.

According to the predictive outcome of Fig.11, the SHPGCN still has outstanding fitting ability in complex spatiotemporal models.

Evolution Behaviors of Traffic Graph: To facilitate the presentation of evolution behaviors for dynamic traffic graph, we project the traffic states of all road sections as a planar diagram, as shown in Fig.12. The road section state is displayed in different colors, where the red dot indicates the lowest speed of 10-30km/h, the blue dot indicates 30-35km/h, and greater than 35km/h is indicated by the yellow dot. The experiment is based on the 20minutes prediction of Xi'an dataset. We compare the ability of KNN-LSTM, PredRNN and SHPGCN to extract the features of the traffic graph. The top shows the real speed variation of the road network in 1.5 hours.

From the markers in the figure, we can see that SHPGCN is closest to the real value for traffic graph prediction. In the third column, the flow diagram of SHPGCN captures the change in the marker and the continuity of that change can be seen in the row of SHPGCN. Although PredRNN has considerable performance, however, there is no continuous focus on this change in its prediction sequence. The KNN-LSTM using long-term neighbor similarity for spatial association performs poorly in short-term traffic prediction. Due to its constant neighbor features, it only captures a few spatial related nodes in the first column of the figure. As can be seen from the circles in the fourth column of the figure, SHPGCN has clear discrimination, while PredRNN and KNN-LSTM have over- or under-fitting in this region. Therefore, SHPGCN can effectively predict the continuous evolution behaviors of traffic graph during traffic peaks.

VI. Conclusion and Future Work

In this paper, we study the evolution of transportation hub to uncover structure where there is an implicit flow transfer in traffic graph. In response to the fact that previous models do not have in-depth mining of spatial features, we propose a recurrent convolutional network SHPGCN that incorporates spatiotemporal hierarchical features. The propagation effects and time-varying correlations between nodes in traffic network are extracted and embedded into the sequential prediction network. In the experiments, we conducted performance evaluation on two datasets to verify the usability of the model. The results verify that SHPGCN can effectively predict the evolution behaviors of traffic graph by modeling the propagation process of traffic flow. We expect to provide an expression for the inherent relation between traffic flow propagation and traffic congestion through complex spatiotemporal feature extraction. Future work is to study multi-modal spatial features in traffic network and improve the predictive ability by fusing more features.

References

[1] 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.

[2] M. Lippi, M. Bertini, and P. Frasconi, "Short-term traffic flow forecasting: An experimental comparison of time-series analysis and supervised learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 2, pp. 871–882, Jun. 2013.

[3] M. Ben-Akiva, M. Bierlaire, H. N. Koutsopoulos, and R. Mishalani, "Real time simulation of traffic demand-supply interactions within dynamit," in *Transportation and Network Analysis: Current Trends*. Boston, MA, USA: Springer, 2002, pp. 19–36.

[4] D. Helbing and A. Greiner, "Modeling and simulation of multilane traffic flow," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 55, no. 5, p. 5498, 1997.

[5] Y. Zhengxiang, X. Guimin, and W. Jinwen, "Transport volume forecast based on GRNN network," in *Proc. 2nd Int. Conf. Future Comput. Commun.*, vol. 3, 2010, pp. V3-629–V3-632.

[6] H.-J. Yang and X. Hu, "Wavelet neural network with improved genetic algorithm for traffic flow time series prediction," *Optik*, vol. 127, no. 19, pp. 8103–8110, Oct. 2016.

[7] W. Huang, G. Song, H. Hong, and K. Xie, "Deep architecture for traffic flow prediction: Deep belief networks with multitask learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 5, pp. 2191–2201, Oct. 2014.

[8] P. Cai, Y. Wang, and G. Lu, "Tunable and transferable RBF model for short-term traffic forecasting," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 11, pp. 4134–4144, Nov. 2019.

[9] Z. C. Lipton, J. Berkowitz, and C. Elkan, "A critical review of recurrent neural networks for sequence learning," 2015, *arXiv:1506.00019*. [Online]. Available: http://arxiv.org/abs/1506.00019

[10] W. Zaremba, I. Sutskever, and O. Vinyals, "Recurrent neural network regularization," 2014, *arXiv:1409.2329*. [Online]. Available: http://arxiv.org/abs/1409.2329

[11] A. Khan, A. Sohail, U. Zahoora, and A. S. Qureshi, "A survey of the recent architectures of deep convolutional neural networks," *Artif. Intell. Rev.*, vol. 53, no. 8, pp. 5455–5516, Dec. 2020.

[12] X. Ma, Z. Dai, Z. He, J. Ma, Y. Wang, and Y. Wang, "Learning traffic as images: A deep convolutional neural network for large-scale transportation network speed prediction," *Sensors*, vol. 17, no. 4, p. 818, Apr. 2017.

[13] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, *arXiv:1609.02907*. [Online]. Available: http://arxiv.org/abs/1609.02907

[14] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 1, pp. 4–24, Jan. 2021.

[15] R. Ying, J. You, C. Morris, X. Ren, W. L. Hamilton, and J. Leskovec, "Hierarchical graph representation learning with differentiable pooling," 2018, *arXiv:1806.08804*. [Online]. Available: http://arxiv.org/abs/1806.08804

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

[17] K. Cho *et al.*, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," 2014, *arXiv:1406.1078*. [Online]. Available: http://arxiv.org/abs/1406.1078

[18] S. Lee and D. B. Fambro, "Application of subset autoregressive integrated moving average model for short-term freeway traffic volume forecasting," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 1678, no. 1, pp. 179–188, Jan. 1999.

[19] Z. Cui, R. Ke, Z. Pu, and Y. Wang, "Deep bidirectional and unidirectional LSTM recurrent neural network for network-wide traffic speed prediction," 2018, *arXiv:1801.02143*. [Online]. Available: http://arxiv.org/abs/1801.02143

[20] Z. Zhao, W. Chen, X. Wu, P. C. Y. Chen, and J. Liu, "LSTM network: A deep learning approach for short-term traffic forecast," *IET Intell. Transp. Syst.*, vol. 11, no. 2, pp. 68–75, Mar. 2017.

[21] 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.

[22] Y. Liu, H. Zheng, X. Feng, and Z. Chen, "Short-term traffic flow prediction with conv-LSTM," in *Proc. 9th Int. Conf. Wireless Commun. Signal Process. (WCSP)*, Oct. 2017, pp. 1–6.

[23] Y. Seo, M. Defferrard, P. Vandergheynst, and X. Bresson, "Structured sequence modeling with graph convolutional recurrent networks," in *Proc. Int. Conf. Neural Inf. Process.* Cham, Switzerland: Springer, 2018, pp. 362–373.

[24] L. Zhao *et al.*, "T-GCN: A temporal graph convolutional network for traffic prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 9, pp. 3848–3858, Sep. 2020.

[25] J. Chen, X. Xu, Y. Wu, and H. Zheng, "GC-LSTM: Graph convolution embedded LSTM for dynamic link prediction," 2018, *arXiv:1812.04206*. [Online]. Available: http://arxiv.org/abs/1812.04206

Authorized licensed use limited to: CTRO.INV. EN MATEMATICAS (CIMAT). Downloaded on October 20,2022 at 19:50:51 UTC from IEEE Xplore. Restrictions apply.

[26] 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, vol. 33, no. 1, pp. 922–929.
[27] X. Wang *et al.*, "Traffic flow prediction via spatial temporal graph neural network," in *Proc. Web Conf.* New York, NY, USA: Association for Computing Machinery, 2020, pp. 1082–1092.
[28] X. Luo, D. Li, Y. Yang, and S. Zhang, "Spatiotemporal traffic flow prediction with KNN and LSTM," *J. Adv. Transp.*, vol. 2019, pp. 1–10, Feb. 2019.
[29] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Graph convolutional recurrent neural network: Data-driven traffic forecasting," Jul. 2017, *arXiv:1707.01926*. [Online]. Available: https://arxiv.org/abs/1707.01926
[30] B. Lu, X. Gan, H. Jin, L. Fu, and H. Zhang, "Spatiotemporal adaptive gated graph convolution network for urban traffic flow forecasting," in *Proc. 29th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2020, pp. 1025–1034.
[31] M. Li and Z. Zhu, "Spatial-temporal fusion graph neural networks for traffic flow forecasting," 2020, *arXiv:2012.09641*. [Online]. Available: http://arxiv.org/abs/2012.09641
[32] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," 2017, *arXiv:1709.04875*. [Online]. Available: http://arxiv.org/abs/1709.04875
[33] Y. Wang, M. Long, J. Wang, Z. Gao, and P. S. Yu, "PredRNN: Recurrent neural networks for predictive learning using spatiotemporal LSTMS," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 879–888.
[34] 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.
[35] 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.
[36] J. Zhu *et al.*, "KST-GCN: A knowledge-driven spatial-temporal graph convolutional network for traffic forecasting," 2020, *arXiv:2011.14992*. [Online]. Available: http://arxiv.org/abs/2011.14992
[37] M. Lv, Z. Hong, L. Chen, T. Chen, T. Zhu, and S. Ji, "Temporal multi-graph convolutional network for traffic flow prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 6, pp. 3337–3348, Jun. 2021.
[38] H. Zhou and K. Hirasawa, "Traffic conduction analysis model with time series rule mining," *Expert Syst. Appl.*, vol. 41, no. 14, pp. 6524–6535, Oct. 2014.
[39] M. T. Schaub, J.-C. Delvenne, R. Lambiotte, and M. Barahona, "Multi-scale dynamical embeddings of complex networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 99, no. 6, Jun. 2019, Art. no. 062308.
[40] J. Lee, I. Lee, and J. Kang, "Self-attention graph pooling," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 3734–3743.
[41] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," 2016, *arXiv:1606.09375*. [Online]. Available: http://arxiv.org/abs/1606.09375
[42] D. K. Hammond, P. Vandergheynst, and R. Gribonval, "Wavelets on graphs via spectral graph theory," *Appl. Comput. Harmon. Anal.*, vol. 30, no. 2, pp. 129–150, Mar. 2011.
[43] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Process. Mag.*, vol. 30, no. 3, pp. 83–98, May 2013.
[44] L. C. Freeman, "A set of measures of centrality based on betweenness," *Sociometry*, vol. 40, no. 1, pp. 35–41, Mar. 1977.

Mengmeng Chang received the B.E. degree from Zhengzhou University, Zhengzhou, China, in 2011, the M.E. degree from the Beijing University of Technology, Beijing, China, in 2018, where he is currently pursuing the Ph.D. degree with the College of Computer Science. His research interests include intelligent transportation systems, data mining, and deep learning.

Zhiming Ding received the B.E. degree from Wuhan University in 1989, the M.E. degree from the Beijing University of Technology, China, in 1996, and the Ph.D. degree from the Institute of Computing Technology, Chinese Academy of Sciences (ISCAS), in 2002.

He is currently a Professor and the Director of the Research Center of Spatial-Temporal Data Management and Data Sciecne (STDMS), Institute of Software, ISCAS. He owns nine invention patents, and has published four books and about 150 articles in academic journals and papers in conferences. His main research interests include database systems, moving objects databases, spatial-temporal and the IoT big data management, and data intelligence.

Zhi Cai received the M.Sc. degree from the School of Computer Science, The University of Manchester, in 2007, and the Ph.D. degree from the Department of Computing and Mathematics, Manchester Metropolitan University, U.K., in 2011. He is currently an Associate Professor with the College of Computer Science, Beijing University of Technology, China. His research interests include information retrieval, ranking in relational databases, keyword search, and intelligent transportation systems.

Zilin Zhao received the B.E. degree from the University of Jinan, Jinan, China, in 2017, and the M.E. degree from the Beijing University of Technology, Beijing, China, in 2020, where she is currently pursuing the Ph.D. degree with the College of Computer Science. Her research interests include intelligent transportation systems, traffic data mining, and gee systems.

Lei Yuan received the bachelor's degree in engineering from the Beijing University of Technology, in 2018, where he is currently pursuing the master's degree with the School of Computer Science. His research interests include vehicle emergency navigation, data mining, and deep learning.

Authorized licensed use limited to: CTRO.INV. EN MATEMATICAS (CIMAT). Downloaded on October 20,2022 at 19:50:51 UTC from IEEE Xplore. Restrictions apply.