Semantics-aware Dynamic Graph Convolutional Network for Traffic Flow Forecasting

Guojun Liang, Kintak U, Xin Ning, Prayag Tiwari, Sławomir Nowaczyk, Neeraj Kumar

***Abstract*—Traffic flow forecasting is a challenging task due to its spatio-temporal nature and the stochastic features underlying complex traffic situations. Currently, Graph Convolutional Network (GCN) methods are among the most successful and promising approaches. However, most GCNs methods rely on a static graph structure, which is generally unable to extract the dynamic spatio-temporal relationships of traffic data and to interpret trip patterns or motivation behind traffic flows. In this paper, we propose a novel Semantics-aware Dynamic Graph Convolutional Network (SDGCN) for traffic flow forecasting. A sparse, state-sharing, hidden Markov model is applied to capture the patterns of traffic flows from sparse trajectory data; this way, latent states, as well as transition matrices that govern the observed trajectory, can be learned. Consequently, we can build dynamic Laplacian matrices adaptively by jointly considering the trip pattern and motivation of traffic flows. Moreover, high-order Laplacian matrices can be obtained by a newly designed forward algorithm of low time complexity. GCN is then employed to exploit spatial features, and Gated Recurrent Unit (GRU) is applied to exploit temporal features. We conduct extensive experiments on three real-world traffic datasets. Experimental results demonstrate that the prediction accuracy of SDGCN outperforms existing traffic flow forecasting methods. In addition, it provides better explanations of the generative Laplace matrices, making it suitable for traffic flow forecasting in large cities and providing insight into the causes of various phenomena such as traffic congestion. The code is publicly available at *https://github.com/gorgen2020/SDGCN*.**

***Index Terms*—Traffic flow forecasting, dynamic graph construction, graph convolutional network, sparse hidden markov model.**

I. Introduction

CCURATE traffic flow forecasting is crucial for intelligent traffic management and smart city applications such as guiding road construction, sensing traffic congestion, and recommending travel routes. Exploring appropriate spatial and temporal characteristics is the key to achieving high performance. Based on this idea, many methods have been proposed in the literature in the last decades. Early traffic flow forecasting methods were based on traditional machine learning and classical statistics, including time sequence models [1–3], Kalman filter model [4], Markov chain model [5], non-parametric methods, simulation models [6, 7], and local regression models [8, 9]. These methods mainly focus on exploring temporal relationships in time sequence data but ignore the spatial correlation originating from the road network topology.

Recently, with the advances in deep learning [10, 11], Convolutional Neural Networks (CNNs) are used to capture the spatial features of grid-based data [12], and Recurrent Neural Networks (RNNs) [13, 14] are employed to exploit the temporal correlation. However, it is hard for these methods to simulate both the spatio-temporal characteristics and the dynamic relationship of traffic data. Although the hybrid model of CNN and RNN can capture some spatio-temporal features [15], the complexity of the traffic road topology surpasses the capability of CNN methods to capture the underlying spatial features in regard to these non-Euclidean traffic data. By formulating traffic prediction as a graph modeling problem, GCNs have obtained state-of-the-art performance in traffic flow forecasting [16]. However, the performance of these works is limited by the use of fixed empirical graphs and insufficient consideration of the dynamic features of the traffic data. Some scholars consider using a dynamic graph to construct GCNs[17–25], but most of them employ static graph (e.g., roads network topology) to construct a dynamic graph with deep learning methods. Moreover, since deep learning methods are usually viewed as a black box, dynamic graph generated by them often lacks strong interpretation. In addition, in order to aggregate the multi-hop neighbour node information, a high-order Laplacian matrix is applied. However, the calculation of such a high-order Laplacian matrix is time-consuming.

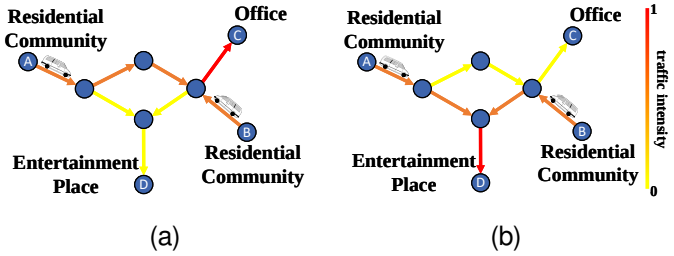


Fig. 1. The dynamic change diagram of traffic flow. (a) Traffic flow during rush hours on weekdays, and (b) Traffic flow during leisure hours on weekends.

Since global positioning systems and traffic sensors have been widely used to record large-scale trajectories of vehicles on the road [26], it becomes possible to understand the dynamic correlation of traffic data through mobility modeling, allowing for a more accurate dynamic graph embedding. Recently, several studies have been devoted to this area [27–29]. However, the main drawback of these mobility models is that they focus mainly on the local relevance of trajectories but ignore the rich semantic relational structure, i.e., trip patterns or motivation behind the mobility. For

This work was partially carried out with support from Vinnova (Sweden's innovation agency) through Vehicle Strategic Research and Innovation programme, FFI. (*Corresponding authors: Prayag Tiwari, Sławomir Nowaczyk, and Neeraj Kumar*)

G. Liang and K. U are with the Faculty of Innovation Engineering, Macau University of Science and Technology, Macau, China (E-mail: gorgen@163.com, ktu@must.edu.mo)

X. Ning is with the Laboratory of Artificial Neural Networks and High Speed Circuits, Institute of Semiconductors, Chinese Academy of Sciences, China (E-mail: ningxin@semi.ac.cn)

P. Tiwari and S. Nowaczyk are with Center for Applied Intelligent Systems Research (CAISR), Halmstad University, Sweden (Email: prayag.tiwari@hh.se, slawomir.nowaczyk@hh.se)

N. Kumar is also with the School of Computer Science, University of Petroleum and Energy Studies, Dehradun, Uttarakhand, and Department of Electrical and Computer Engineering, Lebanese American University, Beirut, Lebanon, and Faculty of Computing and IT, King Abdulaziz University, Jeddah, Saudi Arabia (Email: nehra04@gmail.com, neeraj.kumar@thapar.edu, neeraj.kumar@lau.edu.lb)

example, as shown in Fig. 1, origins appearing at far places with the same destinations and intents at rush hours on weekdays (e.g. two residential communities of going to the office for work in the far away distance) should be considered highly semantically connected. Meanwhile, during the weekend, the origins may be highly semantically connected due to their similar purposes for visiting recreational places. How to discover the dynamic hidden semantic relational graph rather than the static graph (e.g. road topology) is the key challenge in dynamic graph construction.

To address the above problem, inspired by the knowledge from the relation extraction domain, we propose to apply a sparse hidden Markov-based model to learn the intrinsic semantics embedded in the historical trajectories. Since the contextual semantics of the trajectories can provide useful insights about the relationships that exist between these origins and destinations (OD), they can reveal the hidden topology in the road network. For example, if people usually arrive at regions A and B after work, such trajectories can indicate that regions A and B are semantically connected (i.e., with similar motivation, as they may both be there for shopping), even if they are far apart.

We introduce a sparse state-sharing hidden Markov model to capture the patterns of traffic flows from the sparse trajectory data, and to reveal the latent states and transition matrices among them. Consequently, we can build dynamic Laplacian matrices adaptively by jointly considering time, space, and the trip pattern (or motivation) of users. The primary contributions of this paper are as follows:

- A novel Semantics-aware Dynamic Graph Convolutional Network (SDGCN) is proposed for traffic flow forecasting. SDGCN transforms the traditional static graphs into adaptive dynamic semantic graphs, and represents spatio-temporal road connections through dynamic semantics-aware relationships.
- To address the low accuracy problem caused by the scarcity and inconsistency of trajectory data, a State-share Sparse Hidden Markov Model is introduced. It mines personalized features, through which k-hop dynamic semantic Laplacian matrix can be obtained, and consequently simplify the computational complexity of calculating high-order Laplacian matrix.
- We conduct extensive experiments on three real-world traffic datasets. Experimental results demonstrate the superiority of SDGCN in traffic flow forecasting, compared to existing baselines. Furthermore, we showcase examples of improved interpretability of the generated graphs.

II. Related work

A. Traffic flow Forecasting

Traffic flow forecasting focuses on using past geographic information to predict future traffic flow. However, due to the complexity and nonlinear characteristics of the problem, it is not easy to obtain accurate predictions. Many researchers have investigated this field over years and developed various models to achieve higher performance. Traditional statistical models, including Historical Average (HA) [30], Auto-Regressive Integrated Moving Average (ARIMA) [31] and Vector Auto Regression (VAR) [32], are widely accepted due to their interpretability and fast computation. However, their hypothesis of a stationary environment is difficult to satisfy in real traffic scenarios; hence, their performance in the real world is low. Traditional statistical learning-based approaches, such as K-Nearest Neighbor (KNN) [33] and Support Vector Regression (SVR) [34], can model non-stationarity and correlation of multivariate data, but their accuracy often depends on manual feature engineering.

Due to the ability to take advantage of large amounts of data, neural networks (NNs) are widely applied to predict spatio-temporal traffic data [35–37]. These methods can be grouped into three categories: temporal-dependent models, spatial-dependent models, and spatio-temporal dependent models. In order to capture temporal features, RNN models, especially Long Short-Term Memory (LSTM), are often adopted [38]. Attention mechanism is often applied in these models, Fang et al. tried to combine attention mechanism with LSTM to exploit the long temporal features [39]. To capture spatial features, considering the success of CNNs method in the image task, CNNs are often applied in the early stage of traffic flow forecasting. Although CNNs are very suitable for these standard applications of 2D or 3D data in the image processing domain, the fact that traffic topology networks naturally exhibit non-Euclidean structures limits the application of CNNs in traffic flow forecasting [40]. Recently, researchers have tried to explore graph structure information through NNs and developed a new family of neural networks, Graph Neural Networks (GNNs) [41]. As a breakthrough during the previous 5 years, researchers have proposed GCNs to handle these non-Euclidean structured graphs [42].

B. Graph Convolution

Bruna et al. [43] attempted to apply CNNs to graph-structured data and developed a general graph convolution framework based on the spectrum of the graph Laplacian. Defferrard et al. then optimized it by implementing eigenvalue decomposition using the Chebyshev polynomial approximation [44]. Many GCNs and their variant models have been developed for traffic flow forecasting [45–49]. Zhao et al. proposed the T-GCN model, which adopted GCN and Gated Recurrent Unit (GRU) methods to represent spatio-temporal features simultaneously [49]. By using a GCN to exploit spatial features and GRU to exploit temporal features, T-GCN achieved considerable improvements in prediction performance. Different from the discriminative models, Zhou et al. introduced a generative flow-based method named Variational Graph Recurrent Attention Neural Networks (VGRAN) to construct posterior of the traffic distributions for more accurate prediction [47]. However, these models used a predefined static road topology as the adjacency matrix of GCN, ignoring the fact that the graphical topological relationships between roads change dynamically over time. Therefore, these methods exhibit limited potential for the further improvement of prediction accuracy.

C. Dynamic Graph Construction

More and more scholars have noticed that traffic network graphs are dynamic graphs. To construct dynamic graphs, many methods have been proposed [17–19]. Graph Attention Network (GAT) [20] used the attention mechanism to adaptively predict the dynamic adjacency matrices of the graphs. Following this work, Guo et al. [21] developed the Attention-based Spatio-temporal GCN (ASTGCN), which used attention mechanism and GCN to deal with the traffic flow forecasting problem, and the dynamic Laplacian matrices of the graphs were constructed. Attention-based Spatiotemporal Graph Attention Network (ASTGAT) [48] used the

attention mechanism, dilated gated convolution, and graph attention network to capture the spatiotemporal features. The Dynamic Graph Convolutional Recurrent Network (DGCRM) adopted two super networks capturing dynamic information, and then used them to generate dynamic graphs [22]. Most of them are still based on the empirical Laplace matrix of the road network to generate dynamic Laplace matrices, ignoring the intrinsic temporal connection between Laplace matrices of adjacent time periods [23]. DGCN combined spatial attention and LSTM as graph generation layers to obtain higher accuracy [50].Dynamic Spatial-Temporal Aware Graph Neural Network (DSTAGNN) represented dynamic spatial relevance among nodes with an improved multi-head attention mechanism [25]. Spatial-Temporal Graph Ordinary Differential Equation Networks (STGODE) noticed the spatial-temporal dynamics of node signals and captured the dynamic graph of nodes through ordinal differential equation [24]. However, due to the complexity of its model, it was time-consuming to train and predict.

Although some of the dynamic graph construction models mentioned above can obtain high accuracy, most of them lack strong explanations or interpretability for the generation of the adjacency matrix or Laplacian matrix. In addition, a high computational cost is required for computing high-order Laplacian matrices, which is not always practical for realistic applications. In contrast to the above work, we employ a state-sharing sparse hidden Markov model to generate dynamic semantics-aware graphs from traffic trajectory data. The traffic trajectory data generates dynamic semantics-aware graphs, resulting in more interpretable and accurate dynamic graphs for end-to-end learning.

III. Methodology

The purpose of traffic information forecasting is to predict the state of traffic information at several moments in the future based on the current or past observed traffic state. Traffic information includes a variety of sensor data, such as traffic speed, traffic density, and trip time. In this study, we focus on traffic speed prediction in traffic flow forecasting. In a smart city, the traffic speed can be measured through a number of relevant sensors.

A. Problem Definition

Definition 1: ***Traffic network G***. The structure of the transportation network can be naturally represented by a graph $G=(V,E,A)$, where V represents a set of roads in the city, $V=\{v_1,v_2,...,v_N\}$ and N is the total number of roads. In this study, we take roads as nodes of the graph, and v or v_n represents a common road node. E is a set of edges between the nodes. We adopt an adjacency matrix A to represent the connections between the road nodes. In particular, $A\in R^{N\times N}$ for a weighted graph, and the elements of A are either 0 or 1 for a static graph (e.g. based on road network topology).

Definition 2: ***Trajectory flow observation O***. The probability distribution of vehicle trajectories over time T is denoted as matrix $O = [O_1, O_2, ..., O_T] \in R^{N\times T}$, where $O_t = [{o_t}^{(v_1)};{o_t}^{(v_2)};...;{o_t}^{(v_N)}] \in R^{N\times 1}$ represents the traffic flow trajectory at time step t, and ${o_t}^{(v_n)} \in V$ indicates the traffic flow changes between origins and destinations of road nodes at time t. For example, ${o_t}^{(v_n)} = v_m$, means traffic flow from v_n to road v_m at time t.

Definition 3: ***Dynamic semantic adjacency matrix A_t*** extends the previous Definition 1 to a dynamic graph. Different from traditional methods, where edges are fixed by static graph topology, dynamic semantic edges are applied to represent relationships of roads. Each dynamic edge at time t is expressed as the conditional probability of traffic flow from v_n to v_m at time t, which can be formulated as:

$$A_t=\{a_{mn}=p({o_t}^{(v_n)}=v_m)=p(o_t=v_m|o_{t-1}=v_n\}. \quad (1)$$

where $A_t\in R^{N\times N}$ is the semantics dynamic adjacent matrix, a_{mn} is the element of m-th row and n-th column. Correspondingly, $G=(V,E,A)$ will become $G_t=(V,E_t,A_t)$.

Definition 4: ***Feature matrix X***. Traffic data contains various data, which can be expressed as a matrix, $X\in R^{N\times P}$, where P represents the number of node attributes. In this study, we consider traffic flow speed as a feature of interest, and capture the historical P time steps in feature matrix $X=[X_{t-p},...X_{t-2},X_{t-1}]$, where $X_t\in R^{N\times 1}$ is used to represent the speed across all nodes V at time t.

With the above definition, the dynamic semantic adjacent matrix can be calculated as follows:

$$[O_1,O_2,...O_{t-1}] \xrightarrow{g(.)} \hat{O}_t \xrightarrow{Equation1} \hat{A}_t. \quad (2)$$

In the first step, we learn a function $g(.)$ to map past $t-1$ time series observations $[O_1,O_2,...O_{t-1}]$ into the future observation $\hat{O}_t$ at time t. According to Equation 1, we can thus obtain the predictive dynamic semantic adjacent matrix $\hat{A}_t$.

In the second step, the traffic speed prediction can be formulated as follows:

$$\left[X_{t-p},...X_{t-1};\hat{A}_t\right] \xrightarrow{f(.)} [X_t,...,X_{t+T-1}]. \quad (3)$$

Our goal is to learn a function $f(.)$ that maps past p time steps of graph signals $X_{t-p},...X_{t-1}$ (utilizing the predictive dynamic semantic adjacent matrix $\hat{A}_t$) to predict the graph signals in the future T time steps.

B. Method Overview

As shown in Fig. 2, SDGCN consists of three components: dynamic semantic graph generator, graph convolutional network, and temporal gate recursion predictor. Firstly, previous $t-1$ observed traffic trajectories $[O_1,O_2,...,O_{t-1}]$ are fed into the dynamic state-share latent layer to obtain a common latent state-share space C. k-hop (k-hop represents general multi-hop, $k=\{0,1,2,...,K\}$ and K is the Max-hop number) personal feature of each road node v is governed by a personal latent variable Z_t^v and changing with time t, which is sparsely connected to state-share latent space C. For each road node v, we can aggregate k-hop probability into emission probability to capture the inherent semantic features relationship of v. Then, the dynamic semantic graph is generated after all road nodes V are processed this way. Secondly, the output dynamic semantic graph will be conveyed to GCN in the form of a combination of various k-order Laplacian matrices. Thirdly, the node signals $[X_{t-p},...X_{t-2},X_{t-1}]$, together with the generated graph, are fed to the GCN to capture spatial features. Finally, the latest P output features of GCN are input to a temporal gate recursion predictor, capturing the temporal features and obtaining the predicted values $\hat{X}_t$.

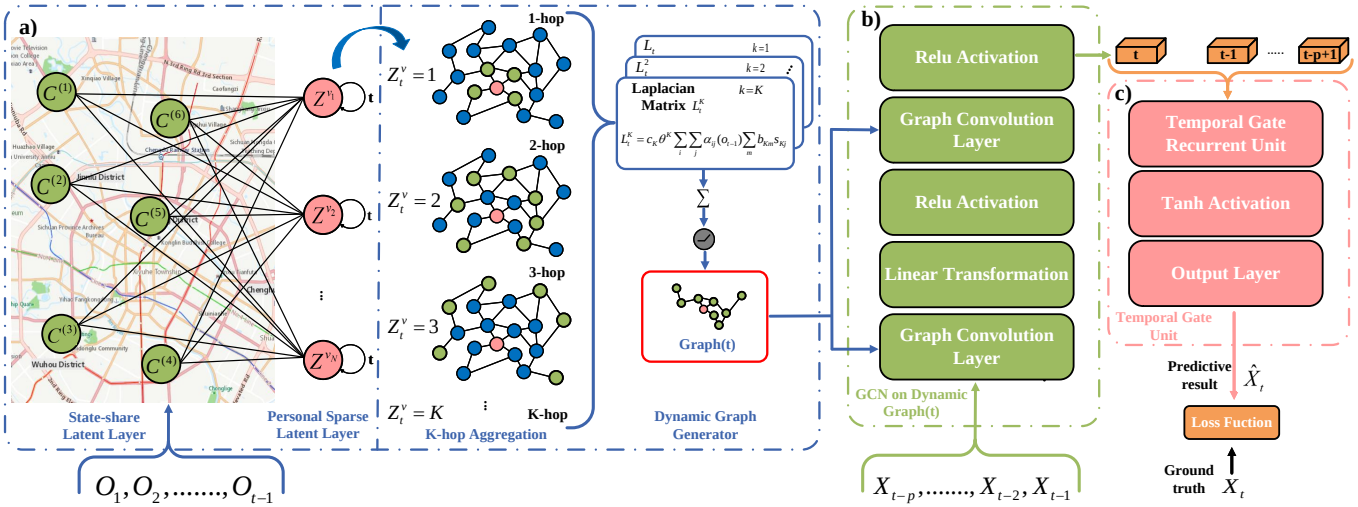


Fig. 2. Overview of the SDGCN architecture: a) dynamic semantic graph generator, b) graph convolution network, and c) temporal gate recursion predictor. In the above, $\hat{X}_t$ is the predictive traffic speed at time t, and L^k is the k-order of the semantic Laplacian matrix.

C. Graph Convolution Network

GCN model builds a filter in the Fourier domain. Laplacian matrix can be formulated as $L = D^{-\frac{1}{2}}(A+I)D^{\frac{1}{2}}$, where I is the identity matrix, which means adding self-connections, D is the degree matrix. For GCN, considering K-hop aggregation, GCN model can be expressed by the following formula [51]:

$$X^{(l+1)} = f\left(A, X^{(l)}\right) = \sigma\left(\sum_{k=1}^{K} L^k X^{(l)} w^{(l)}\right), \tag{4}$$

where $X^{(l)} \in R^{N\times P}$ is the input of l layer, while $X^{(l+1)} \in R^{N\times T}$ is output of l layer in future T time steps, k denotes the number of hops aggregation and $k \in \{1,...,K\}$. In addition, $w \in R^{P\times T}$ is a learnable parameter, and σ represents the nonlinear activation function. To capture the dynamic semantic relationship, Definition 3 is employed as a new adjacency matrix. Two-layer GCN model [52] is used to obtain dynamic semantic traffic relationships. In T-GCN [49], K is set to 1 for simplification, which means only the first-order neighborhood nodes are considered while ignoring high-order neighborhood nodes, since it is time-consuming to compute the K-order Laplacian matrix L^K. If we calculate it directly, the time complexity would be $\mathcal{O}\left(N^{3(K-1)}\right)$. Moreover, with a higher-order of Laplacian matrix, it is prone to over-smoothing [53]. To exploit the high-hop neighbour nodes information, the k-order Laplacian matrix should be considered. Therefore, how to calculate semantic dynamic k-order Laplacian matrix L^k and $\sum_{k=1}^{K} L^k$ becomes a key problem.

D. Dynamic Semantic Graph Generator

The Hidden Markov Model (HMM) is a way of modeling sequential processes with unobservable states. As shown in Fig. 3a, HMM assumes that some latent states govern the sequence observations, and these latent states obey the Markov assumption: that the probability of state transition (next state) only depends on its current state.

1) Dynamic Semantic Graph by HMM: HMM is a powerful method for learning dynamic semantic relations. Formally, let K be the number of latent states and k-th latent state govern the dynamic

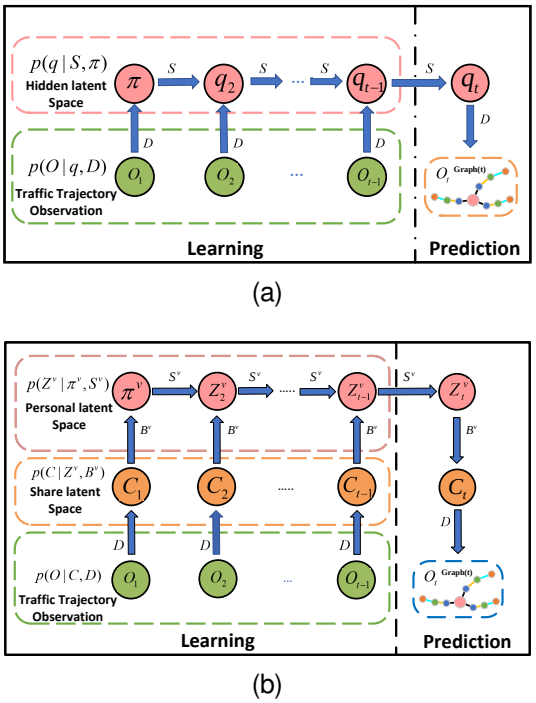


Fig. 3. The framework of HMM and State-share Sparse HMM: (a) illustration of the HMM-based prediction model details in the latent space and its relationship with the traffic trajectory, and (b) illustration of the details of State-share Sparse HMM-based prediction model in the latent space and its relationship with the traffic trajectory.

semantic relationship of k-hop, then in the classical HMM, there is a triplet of parameters $\lambda = \{\pi, S, D\}$. To facilitate the subsequent discussion, the parameters are defined as follows [54]:

- π denotes the initial probability of the k latent state, $\pi \in R^{K\times 1}$, where $\pi_i = p(q_1 = i)$.
- State transition matrix is defined as $S = \{s_{ij}\} \in R^{K\times K}$, which shows the transition probabilities between K hidden states. The probability of transiting from the i-th state to the j-th state is defined by s_{ij}, $s_{ij} = p(q_t = j \,|\, q_{t-1} = i)$.
- Emission matrix $D = \{d_i(O_t)\} \in R^{K\times N}$, which denotes the

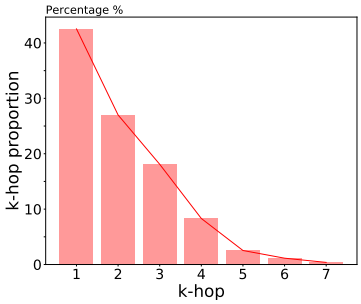


Fig. 4. Trajectory data proportion of k-hop in Xi'an

probabilities of i-th hidden state emitting to O_t observation with $d_i(O_t)=p(O_t\,|\,q_t=i)$.

According to the HMM theory, a_{mn} from Equation 1 can be calculated by the forward algorithm in the prediction period as follows:

$$a_{mn}=\sum_{j=0}^{K}d_j(O_t=v_m)\sum_{i=0}^{K}s_{ij}\alpha_i(O_{t-1}=v_n), \tag{5}$$

where $\alpha_i\ (O_{t-1}=v_n) = p\ (O_1,O_2,...,O_{t-1}=v_n,q_{t-1}=i)$, $d_j\ (O_t=v_m) = p\ (O_t=v_m\,|\,q_t=j)$. In this study, we apply latent states $q_t=j$ to govern the j-hop relationship of traffic trajectory. For the transition probability of traffic trajectories, the probability of one hop is much larger than the probability of multiple hops, and multiple hops can be regarded as an accumulation of one hop. Therefore, the observation can be considered a priori as conforming to the multinomial distribution. Some restrictions must be attached to the form of the model to ensure that the parameters of the Probability Density Function (PDF) can be re-estimated in this way. In particular, $D=\{d_j(O_t)=c_j\theta_m^j\}$, where c_j is the normalization factor, and θ_m is considered as the single-hop probability to node v_m. Thus, Equation 5 can be expressed as follows:

$$a_{mn}=\sum_{j=0}^{K}c_j\theta_m^j\sum_{i=0}^{K}s_{ij}\alpha_i(O_{t-1}=v_n). \tag{6}$$

Let $H_j=c_j\sum_{i=0}^{K}s_{ij}\alpha_i(O_{t-1}=v_n)$. Then, $a_{mn}=\sum_{j=0}^{K}H_j\theta_m^j$, which is equivalent to the semantic dynamic k-order Laplacian matrix L^k and $\sum_k^K L^k$. Moreover, if the hidden layer has K states, the time complexity of computing the K-hop Laplacian matrix can be reduced to $\mathcal{O}(NK)$ in the prediction stage.

2) Dynamic Semantic Graph by State-sharing Sparse HMM: HMM model can be a powerful tool for learning semantic multi-hop features of traffic trajectories when the amount of trajectory data transmitted by road is sufficient and consistent. However, as shown in Fig. 4, multi-hop data is often scarce, especially for far distances, where the data of close range is much larger than that of the remote. In this situation, there is often too little data to train a reliable model, and the accuracy of the HMM is relatively low.

To solve this problem, the State-share Sparse HMM model [55] is employed, where the hidden layer is divided into two layers. The design of the State-share Sparse HMM obeys the following principles: 1) the shared latent layer shares the same latent state set to avoid data scarcity; and 2) the individual latent layer maintains its personal transition matrices, which is sparse and small, to build personalized transition models.

As shown in Fig. 3b, the input $O=\{O_1,O_2,...,O_{t-1}\}$ is a set of observations of the traffic trajectory information. The observations are controlled by a set of hidden variables in a shared latent layer, and for each sequence of observations, two levels of latent states can be defined: 1) $C=\{c_1,c_2,...,c_t\}$ is a set of latent states shared by all roads, which governs the distribution of observations, and 2) $Z^v=\{\pi^v,z_2^v,...,z_t^v\}$ is the individual latent states, which dominates a few shared latent states associated with each road. The shared latent state space, which consists of a series of shared latent states, is shared by each road to avoid suffering from data scarcity. The personal latent space is an individual latent states layer to capture the personalized characteristics of each road. $D=\{d_m\}$ defines the probabilities of m-th latent states emitting specific observations. For each road, the triple personalization parameter $\Phi^v=\{\pi^v,S^v,B^v\}$ can be defined [55]:

- A vector $\pi^v\in R^{K\times 1}$, where $\pi_i^v=p(z_1^v=i)$, which denotes the initial probability distribution of personal latent states.
- A matrix $S^v=\{s_{ij}^v\}\in R^{K\times K}$, which denotes the probabilities transiting among K personal latent states by $s_{ij}^v=p\big(z_t^v=j\,|\,z_{t-1}^v=i\big)$. Similarly, K is the number of personal latent states and k-th latent states govern road k-hop relationship.
- A matrix $B^v=\{b_{im}^v\}\in R^{K\times M}$, which denotes the probabilities of i-th personal hidden state emitting to the m-th shared latent state with $b_{im}^v=p(c_t=m\,|\,z_t^v=i)$.

Although the State-share Sparse HMM can solve the scarcity of multi-hop data, how to compute a_{mn} becomes a new problem, as the forward algorithm applied to solve the classical HMM is no longer suitable. Therefore, we develop a new forward algorithm as follows. We define

$$\begin{aligned}&\alpha_{km}(O_t)=p(o_1,...,o_t,c_t=m,z_t=k)\\&=\sum_i\sum_j p(o_1,...,o_t,c_t=m,z_t=k,c_{t-1}=i,z_{t-1}=j).\end{aligned} \tag{7}$$

By Bayes theorem and Markov assumption:

$$\begin{aligned}&\alpha_{km}(O_t)\\&=\sum_i\sum_j\alpha_{ij}(O_{t-1})p(c_t\,|\,z_t)p(z_t\,|\,z_{t-1})p(o_t\,|\,c_t).\end{aligned} \tag{8}$$

By triple-tuple personalized definition:

$$\alpha_{km}(O_t=v_m)=\sum_i\sum_j\alpha_{ij}(O_{t-1}=v_n)b_{km}S_{kj}d_k(O_t=v_m). \tag{9}$$

Analogous to HMM, $D=\{d_k(O_t=v_m)=c_k\theta_m^k\}$, where $k=1,2,...,K$, c_k is the normalization factor, θ_m is regarded as one-hop probability to v_m. Thus, $d_k(O_t)$ defines a multinomial distribution of the k-hop latent state over the shared latent state. Thus, a_{mn} from Equation 1 can be calculated as follows:

$$a_{mn}=\sum_k c_k\theta_m^k\sum_i\sum_j\alpha_{ij}(O_{t-1}=v_n)b_{km}S_{kj}. \tag{10}$$

Similarly, let $H_k=c_k\sum_i\sum_j\alpha_{ij}(O_{t-1}=v_n)b_{km}S_{kj}$, we have $a_{mn}=\sum_k^K H_k\theta_m^k$, which is equivalent to the semantic dynamic k-order Laplacian matrix L^k and $\sum_k^K L^k$. Moreover, if the personal hidden layer has K states, the time complexity of computing the K-hop Laplacian matrix can be reduced to $\mathcal{O}(MK+K^2)$ in the

prediction period. Note that M is much larger than K, and the time complexity can be approximately equal to $\mathcal{O}(MK)$. In sum, the time computational complexities of the three methods to calculate the high-order Laplace matrix are shown in Tab I. For the general case, the number of road nodes N is larger than the shared state latent parameter M. Therefore, the lowest time complexity can be achieved by the State-share Sparse HMM.

TABLE I
THE TIME COMPLEXITIES OF CALCULATING HIGH-ORDER LAPLACIAN MATRIX IN THEORY.

Method	Directly	HMM	State-Share Sparse HMM
Time complexity of calculating L^K	$\mathcal{O}(N^{3(K-1)})$	$\mathcal{O}(NK)$	$\mathcal{O}(MK)$

E. Temporal Gate Recursion Predictor

To capture the temporal features, GRU is adopted as a temporal gate recursion predictor in this study. GRU is an RNN network similar to LSTM, but more simplified, since it has only two gates. GRU merges the input gate and the forgetting gate from LSTM into a single gate, called the update gate, which controls the amount of the previously remembered information retained to the current moment. The other gate is the reset gate, which controls the amount of past information that would be forgotten in the current moment. The mechanism of its operation can be explained by the following equations [38]:

$$g_t = \sigma(W_z \cdot [h_{t-1}, x_t] + b_u) \tag{11}$$

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t] + b_r) \tag{12}$$

$$\tilde{h}_t = \tanh(W \cdot [r_t * h_{t-1}, x_t] + b_c) \tag{13}$$

$$h_t = (1 - g_t) * h_{t-1} + g_t * \tilde{h}_t, \tag{14}$$

where h_t represents hidden state at time t, while $\widetilde{h}_t$ is the candidate state at time t, g_t is the update gate, r_t is the reset gate.

IV. EXPERIMENT

A. Dataset Description

In this study, three real traffic datasets, Xi'an city and Chengdu city in China are collected, which are obtained from the data source: Didi Chuxing GAIA Initiative (*https://gaia.didichuxing.com*). The traffic data (15 days of Xi'an and 10 days of Chengdu) are selected to train the model as the traffic speed has obvious single-week periodic properties. The other dataset is Beijing urban city in china from T-Drive dataset, which is obtained from the data source: Microsoft company (*https://www.microsoft.com/en-us/research/publication/t-drive-driving-directions-based-on-taxi-trajectories/*)[56, 57]. The traffic data (4.2 days of Beijing) are selected to train the model with the non-week periodic property.

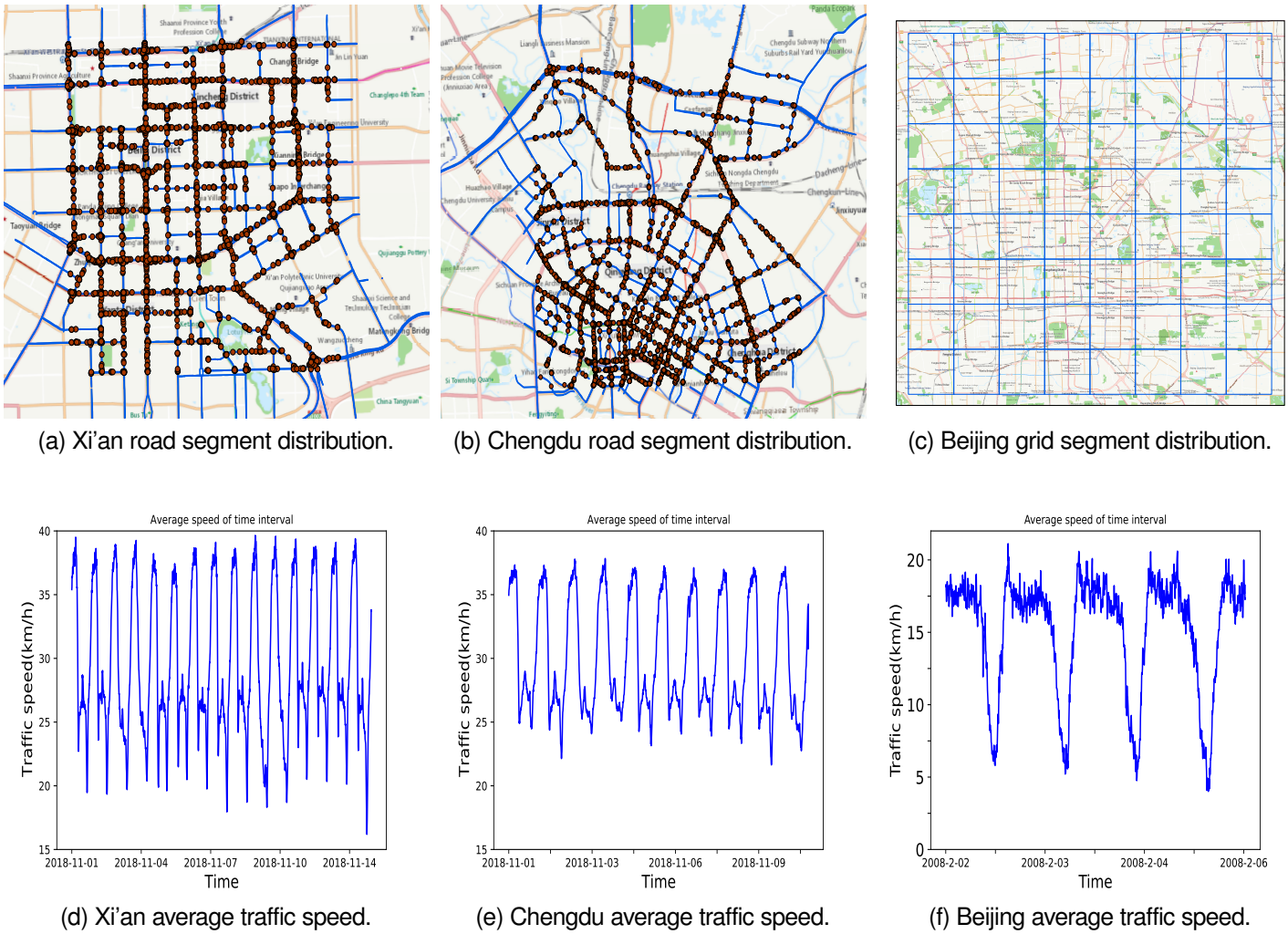
- Xi'an: Totally 251 road sections in the urban area of Xi'an were chosen. The period was from 1st Nov. 2018 to 15th Nov. 2018, a total of 15 days. The traffic speed of each road was sampled every 10 minutes. A total of 541,909 speed sampling points were collected. There were 859,088 matched vehicle trajectories at the same time, which were applied to generate the semantic dynamic adjacency matrices.
- Chengdu: Totally 505 road sections in the city proper of Chengdu were chosen. The period was from 1st Nov. 2018 to 10th Nov. 2018, a total of 10 days. The traffic speed of each road was sampled every 10 minutes. A total of 727,200 speed sampling points were collected. There were 909,697 matched vehicle trajectories at the same time, which were applied to generate the semantic dynamic adjacency matrices.
- T-drive of Beijing city: We partition the Beijing city into 8 $\times$ 8 grids, and then the traffic speed of each grid was sampled every 5 minutes. The period was from 2nd Feb. 2008 to 6th Feb. 2008. A total of 77,248 speed sampling points were collected. There were 5,049,436 matched vehicle trajectories at the same time, which were applied to generate the semantic dynamic adjacency matrices.

As for some missing traffic speed data, the average speeds are calculated to fill in the missing values. The data were preprocessed by Min-Max scaling and the 70% of the first-time series is used as the training set (10.5 days of Xi'an, 7 days of Chengdu and 3 days of Beijing), and the remaining 30% is used as the test set. Horizon 1, horizon 2 and horizon 3 indicate the traffic speed of the next 10 minutes, 20 minutes and 30 minutes in Xi'an and Chengdu and the next 5 minutes, 10 minutes and 15 minutes in Beijing. The overall road area distribution is shown in Fig. 5. The blue lines in Fig. 5a and 5b are the road networks and road segment, and the red circles are the trajectory points of vehicles on the road segment, Fig. 5d and Fig. 5e are the average traffic speed of all roads, showing a certain periodicity. The blue lines in Fig. 5c are the grid segment of Beijing, and Fig. 5f are the average traffic speed of all grids.

B. Baselines

The proposed model is compared with the notable traffic flow forecasting methods. These methods can basically reflect the baselines of traffic flow speed forecasting. The baselines are described as follows:

- HA [30], predicts the future speed by the average traffic information in the historical periods.
- ARIMA [58], integrates auto-regression with moving average model on the time sequence to predict future traffic data.
- SVR [34], is another classical time sequence analysis regression method by using linear support vector machine for regress task, which applies the historical time sequence data to train the model and then predicts the future traffic speed by the trained model.
- GCN [52], is a neural network for processing convolution on graph data structures.
- GRU [38], is a RNN method like LSTM, but with fewer parameters than LSTM, which is utilized in learning time series.
- T-GCN [49], adopts GCN and GRU to exploit spatial and temporal dependencies.
- ASTGCN [21], applies attention module and GCN to train and predict dynamic Laplacian matrices and speed.
- DGCN [50], applies attention and LSTM module, together with GCN to train and predict dynamic Laplacian matrices and speed.
- DSTAGNN [25], applies multi-head attention, together with Multi-scale Gated Tanh Unit (M-GTU) convolution module to train and predict dynamic Laplacian matrices and speed.



(a) Xi'an road segment distribution. (b) Chengdu road segment distribution. (c) Beijing grid segment distribution.

(d) Xi'an average traffic speed. (e) Chengdu average traffic speed. (f) Beijing average traffic speed.

Fig. 5. Traffic segment distribution of three cities.

In order to facilitate the comparison with other methods of baselines, widely used metrics are applied in this study:

- Root Mean Squared Error(RMSE):

$$RMSE(y,\hat{y}) = \sqrt{\frac{1}{|\Omega|}\sum_{i\in\Omega}(y_i-\hat{y}_i)^2}$$

- Mean Absolute Error(MAE):

$$MAE(y,\hat{y}) = \frac{1}{|\Omega|}\sum_{i\in\Omega}|y_i-\hat{y}_i|$$

- Explained Variance Score (EVS):

$$EVS = 1-\frac{\sum_{i\in\Omega}\mathrm{var}(y_i-\hat{y}_i)}{\sum_{i\in\Omega}\mathrm{var}(y_i)}$$

where $y = \{y_1,y_2,...,y_N\}$ denotes the ground truth speed value, while $\hat{y} = \{\hat{y}_1,\hat{y}_2,...,\hat{y_N}\}$ denotes the predicted speed value. Ω denotes the test samples. $RMSE$ and MAE are quantitatively used to describe the difference between the predictive value and the ground truth value. The smaller the value is, the more accurate the model is. EVS weighs the ability of the predicted result to represent the actual data, the larger the value is, the better the predictive effect is.

- Accuracy of traffic trajectory prediction:

$$Accuracy = \sum_{top\_num} I\left(O_t,\hat{O}_t\right)/N$$

I is the indicative function. In this study, $top\_num = 5$, the top five predicted probabilities of traffic trajectory of the next time with ground truth values are evaluated.

C. Hyper Parameters

The proposed model is implemented in Tensorflow 1.9.0 on a workstation with 16G memory Nvidia GTX 1070 GPU and i7-7700HQ CPU. As for the deep Learning framework, the initial learning rate is set to 0.0001. Adam Optimization is utilized, and the batch size is 32. 12 samples are used to train. Training epochs is 1000, L2 loss is applied as loss function of our model, with a regularization parameter 0.0015. Two-layer GCN architecture and Relu activation function are adopted, and the number of GRU hidden units is set to 256. There are two key parameters for State-share Sparse HMM: 1) Shared states number M, and 2) Personal latent states number K. As the number of roads in Chengdu is twice that of Xi'an, the M and K values of Chengdu are 400 and 5 while Xi'an are 200 and 5, Beijing dataset are 32 and 2.

D. Results

Taking T-drive data as an example, firstly, outlier trajectory points that deviated from 8×8 grids are removed, then the preprocessed trajectory data are sliced according to the 1207 node signal time interval. Thirdly, the historical trajectory data are fed into the State-share Sparse HMM model for training, and the 64×64 Laplacian matrix of the next time is predicted. The Laplacian matrices predicted by HMM or State-share Sparse HMM are asymmetric in most cases. However, GCN requires that the input Laplacian matrices are symmetric, so the larger of the road or grid relationships between two roads or grids at adjacent moments in Definition 3 is taken as the common value:

$$a_{m,n} = a_{n,m} = \max\{p(O_t = v_n \,|\, O_{t-1} = v_m), \\ p(O_t = v_m \,|\, O_{t-1} = v_n)\}. \tag{15}$$

Finally, through Equation 15, the symmetric Laplacian matrix is applied to GCN and GRU to extract spatio-temporal features and predict the value of traffic flow speed at the next several moments.

Tab. II shows the performance of the baseline models based on Chengdu, Xi'an and Beijing in the next three time horizons. The performance is compared for Horizon1, Horizon2 and Horizon3. It can be seen from Tab. II that SDGCN achieves the best results in three datasets. Traditional methods cannot guarantee that each dataset will produce the desired results. As a side note, for complex nonlinear spatio-temporal traffic problems, other methods usually fail to capture the essential spatial-temporal features, while methods that consider both space and time such as T-GCN, SDGCN, ASTGCN, DGCN and DSTAGNN have better performance than methods that consider only one factor. In addition, for complex traffic problems, dynamically tuned GCNs often achieve better results than approaches using static graphs.

Interestingly, without changing any other parameters, simply replacing the fixed Laplacian matrices with semantic dynamic Laplacian matrices provides a considerable improvement to the conventional GCN model, which is denoted as GCN_improved in Tab. II.

The top-performing models are compared in more detail in Fig. 6. The $RMSE$ and MAE of all models increase as the time increases, and the prediction results of different methods vary for three datasets. Therefore, long-time prediction of traffic flow is still difficult. SDGCN model achieves better performance for long-term prediction. It is worth noting that for the roads in Chengdu, the number is twice than that in Xi'an. SDGCN has the least difference in prediction accuracy for three datasets and obtains better robustness. This is due to the fact that it dynamically constructs the topological relationship of roads based on real traffic trajectories. Since it is difficult to capture traffic flow with only a single speed dataset, dynamically adjusting the topological relationship of roads by tracking traffic trajectories can better reflect the internal relationship and exhibits better interpretability, which will be discussed in the next section.

To better explore the performance of the graph construction model, the elapsed time of the dynamic graph model is shown in Tab. III. The prediction time increases as the number of road nodes increases. Note that the number of road nodes in Chengdu is twice than that in Xi'an, and the prediction time changes slightly by applying SDGCN, since the use of State-share Sparse HMM has less impact on the increase of road nodes compared to other methods (i. e., attention mechanism). As to the time of generating high-order Laplacian matrix, experiments show that SDGCN takes less time than ASTGCN, DGCN and DSTAGNN, as expected based on its lower time complexity. Thus, it is efficient and beneficial to analyze the traffic flow even in big cities, where a large number of roads makes previous approaches impractical.

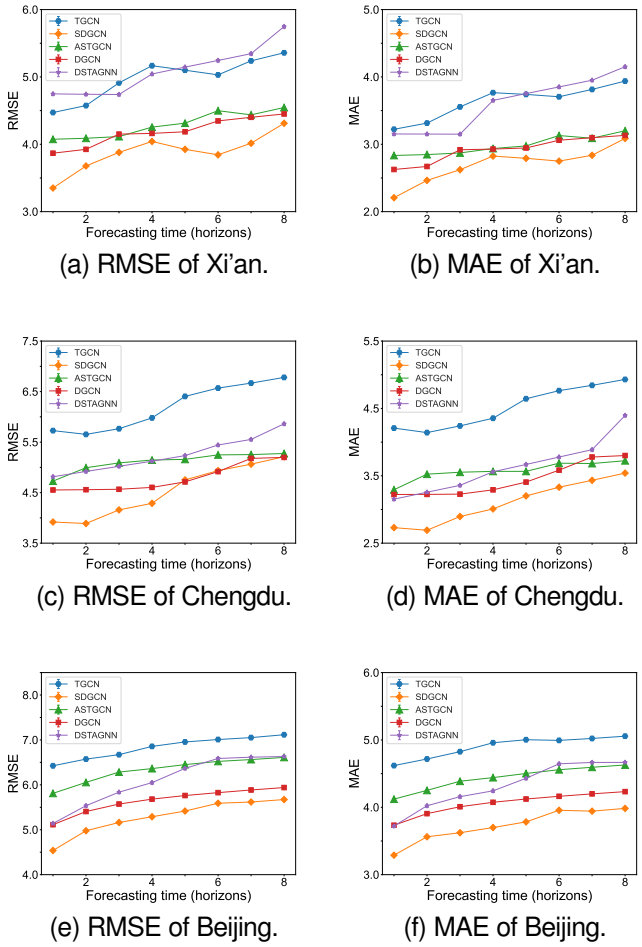


Fig. 6. The predictive results of different methods on Xi'an, Chengdu and Beijing.

E. Ablation Study

In order to verify the effectiveness of the SDGCN model, we have made variants of SDGCN. (1) TGCN: completely replace the dynamic graph with the fixed road topology but retain the GRU module, then there will be the same network structure as TGCN. (2) SDGCN*: completely replace State-share Sparse HMM with traditional HMM dynamic graph constructive module but retain GRU module. (3) GCN_improved: retain the State-share Sparse HMM dynamic graph generation module, and remove the GRU module. (4) GCN: completely replace the dynamic graph generation module with the fixed road topology and remove the GRU module, then there will be the same network structure as traditional GCN. We conduct the ablation experiments on three datasets. Fig. 7 show the results of every variant. It can be seen that the performance of our SDGCN is better than other variants, which confirms the effectiveness of each component in our model.

TABLE II
THE PREDICTION RESULTS OF SDGCN AND OTHER BASELINE METHODS ON THREE DATASETS.

	Time	Horizon 1			Horizon 2			Horizon 3		
	Metric	RMS	MAE	EVS	RMS	MAE	EVS	RMS	MAE	EVS
Xi'an	HA[30]	4.8580	3.3804	0.7794	5.1645	3.6133	0.7506	5.4483	3.8344	0.7222
	ARIMA[58]	7.1284	5.6280	0.0082	7.1300	5.6290	0.0082	7.1326	5.6309	0.0078
	SVR [34]	4.9242	3.3237	0.7759	5.3821	3.6770	0.7315	6.0527	4.1831	0.6595
	GCN[52]	5.3607	3.8000	0.7347	5.5673	3.8846	0.7139	6.5662	4.2323	0.6021
	GRU[38]	3.5479	2.2420	0.8822	4.1200	2.6310	0.8412	5.5594	3.0830	0.7116
	TGCN[49]	4.4719	3.2226	0.8120	4.5747	3.3145	0.7984	4.9122	3.5553	0.7880
	ASTGCN[21]	4.0746	2.8342	0.7977	4.0896	2.8471	0.7960	4.1144	2.8716	0.7942
	DGCN[50]	3.8679	2.6260	0.8597	3.9250	2.6707	0.8589	4.1509	2.9187	0.8491
	GCN_improved	4.2793	2.9415	0.8310	4.1200	2.6310	0.8412	5.5594	3.0833	0.7116
	DSTAGNN[25]	4.7483	3.1531	0.7869	4.7424	3.1520	0.7875	4.7394	3.1501	0.7876
	SDGCN (ours)	**3.3507**	**2.2069**	**0.8973**	**3.6780**	**2.4639**	**0.8753**	**3.8796**	**2.6224**	**0.8611**
Chengdu	HA[30]	4.8537	3.3429	0.7855	5.0947	3.5171	0.7635	5.3269	3.6859	0.7408
	ARIMA[58]	6.1944	4.8990	0.0074	6.1963	4.9009	0.0074	6.1996	4.9037	0.0070
	SVR[34]	4.7893	3.3402	0.7915	4.9997	3.4924	0.7727	5.2213	3.6708	0.7519
	GCN[52]	6.1273	4.3052	0.6594	6.5924	4.7229	0.6058	6.8606	5.1157	0.5731
	GRU[38]	4.6305	2.9511	0.8054	5.9112	3.4386	0.6866	7.2614	4.9793	0.5453
	TGCN[49]	5.7276	4.2080	0.7098	5.6533	4.1413	0.6667	5.7656	4.2409	0.6656
	ASTGCN[21]	4.7280	3.2928	0.7616	4.9932	3.5238	0.7683	5.0884	3.5529	0.7674
	DGCN[50]	4.5525	3.2220	0.8219	4.5579	3.2239	0.8213	4.5666	3.2277	0.8207
	GCN_improved	5.8484	4.1401	0.6885	5.7493	4.0000	0.6997	5.9231	4.1404	0.6819
	DSTAGNN[25]	4.8142	3.1532	0.7911	4.9193	3.2561	0.7614	5.0211	3.3582	0.7515
	SDGCN (ours)	**3.9181**	**2.7303**	**0.8640**	**3.8890**	**2.6914**	**0.8421**	**4.1583**	**2.8955**	**0.8260**
Beijing	HA[30]	5.5293	3.7527	0.6728	5.7077	3.8633	0.6517	5.8445	3.9529	0.6342
	ARIMA[58]	7.7196	6.1278	0.0012	7.7209	6.1292	0.0012	7.7216	6.1296	0.0012
	SVR[34]	5.8192	3.9078	0.6446	6.2670	4.1119	0.5829	6.2952	4.2240	0.5784
	GCN[52]	6.7045	4.7545	0.5086	6.8244	4.8042	0.4930	6.8415	4.8501	0.4904
	GRU[38]	6.7582	4.7222	0.5112	6.8904	4.9704	0.4956	7.3706	5.3292	0.4281
	TGCN[49]	6.4244	4.6195	0.5494	6.5732	4.7173	0.5313	6.6722	4.8259	0.5192
	ASTGCN[21]	5.8108	4.1214	0.6855	6.0542	4.2514	0.6585	6.2844	4.3870	0.6320
	DGCN[50]	5.1132	3.7339	0.7603	5.4057	3.9058	0.7315	5.5727	4.0083	0.7116
	GCN_improved	6.4270	4.5140	0.5586	6.4350	4.5171	0.5580	6.4630	4.5560	0.5553
	DSTAGNN[25]	5.1389	3.7216	0.7612	5.5359	4.0227	0.6841	5.8371	4.1578	0.6712
	SDGCN (ours)	**4.5375**	**3.2890**	**0.7739**	**4.9778**	**3.5621**	**0.7347**	**5.1615**	**3.6224**	**0.7142**

TABLE III
THE PREDICTION RESULTS OF SDGCN AND OTHER BASELINES METHODS ON THREE DATASETS.

Method	Xi'an			Chengdu			Beijing		
	training time (s/epoch)	predicting time(s)	Gernerating Laplacian Matrix L^5time (ms)	training time (s/epoch)	predicting time(s)	Gernerating Laplacian Matrix L^5time (ms)	training time (s/epoch)	predicting time(s)	Gernerating Laplacian Matrix L^5time (ms)
ASTGCN[21]	**1.73**	3.11	998.35	**4.16**	3.44	2015.63	**0.70**	0.10	10.14
DGCN[50]	18.95	12.59	1382.49	28.02	27.57	5495.40	5.09	0.49	17.45
DSTAGNN[25]	17.81	2.39	310.41	123.47	22.21	509.62	3.91	0.47	4.32
SDGCN(ours)	7.25	**1.34**	**234.61**	9.04	**1.52**	**497.25**	2.90	**0.09**	**2.11**

F. Effects of parameters

For the dynamic graph construction, the main parameters are the number of shared hidden states M and the number of personal states K. We evaluate the performance when these parameters change on the condition that all other parameters are kept at their default values.

- Effect of M: For the three datasets, the changes are roughly similar. As shown in Fig. 8a, 8d and 8g, when the value of M is small, accuracy of TOP 1-5 increases greatly with the increment of M, but when M increases to a certain level, it becomes saturated, and a further increase is slow.
- Effect of K: When K is relatively small, as shown in Fig. 8b, 8e and 8h, TOP 1-5 accuracy changes a lot with the increase of K, but as K reaches a certain level, the accuracy decreases dramatically. It shows that the probability of very high-hop events in a short period is relatively low.

Applying the State-share Sparse HMM to represent the traffic trajectories, the dynamic semantic relationships of the road nodes will be obtained. The output Laplacian matrices are Markov matrices based on the previous position to infer the probability of the next transition. The accuracy of the HMM and the State-share Sparse HMM learned in the time series of three cities are analyzed. As shown in Fig. 8c, 8f and 8i, due to the sparsity and inconsistency of the trajectory dataset, the accuracy of the conventional HMM model is relatively poor, reaching no more than 40%, and exhibits large fluctuations over time. At the same time, the accuracy of the State-share Sparse HMM model is significantly improved, with accuracy basically about $80\%-90\%$ and slighter fluctuations.

G. Interpretation of the Generative Laplace Matrices

To visualize the Laplacian matrices generated by the State-share Sparse HMM model, Fig. 9 plot part of the output Laplacian

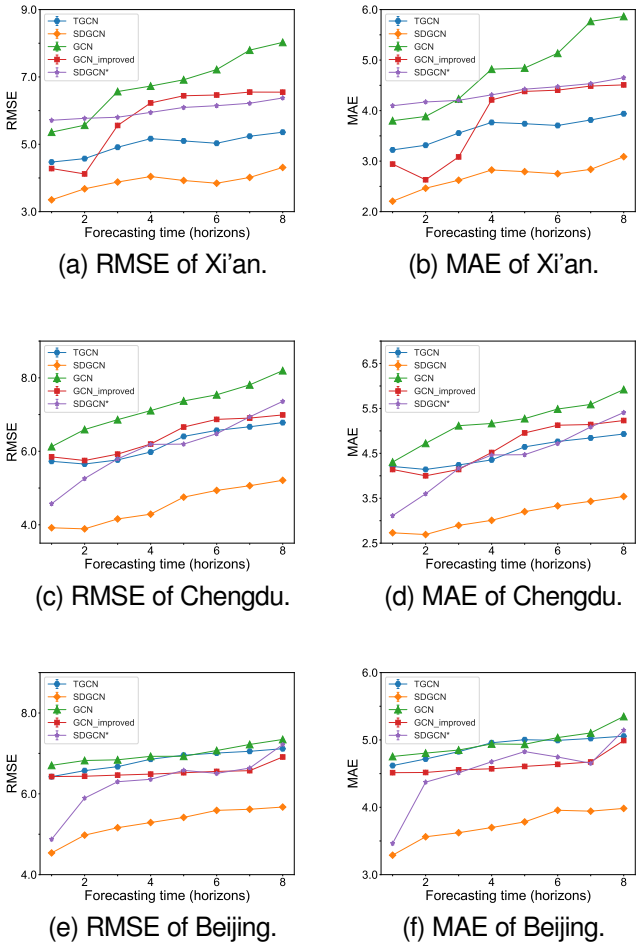


Fig. 7. Ablation experiment of module effectiveness.

matrices (50×50 road segment or grid segment) and Tab. IV to compare them with the Laplacian matrices generated by the classical HMM model. The prediction results of classical HMM are limited to a smaller number of local road nodes and they have difficulties in capturing the global features, which exhibits a lot of zero values, while the State-share Sparse HMM model can take the global network into account through two-layer structure, which displays more non-zero value. The State-share layer tries to capture the common features of all roads and the personal states layer can preserve the features of each road through sparse connections. In this way, the output Laplacian matrix of the State-share Sparse HMM can represent the dynamic semantic relationships, even if the distances between different roads are far away. Therefore, the output Laplacian matrices of the State-share Sparse HMM can better extract the dynamic features of the global road network. The dynamic Laplacian matrices not only make the speed prediction more accurate, but also result in a better interpretation of the generative Laplacian matrices. It provides a strong basis for the generated graphs in the next moment to infer the motivation behind the trip, which is important for subsequent traffic management. For example, as shown in Fig. 10, during weekday working hours, the road connections between residential community areas and work offices will become closer with more dense traffic trajectory points and the semantic multi-hop reveals the relationship from residence community to working offices, however, during weekend leisure hours, the road connections between community areas and entertainment halls will become closer with more dense traffic trajectory points and the semantic multi-hop reveals the relationship from residence community to entertainment halls. Therefore, the semantics-aware dynamic map can reflect these internal relationships more clearly.

V. Conclusion

In this paper, a novel traffic flow forecasting method, Semantics-aware Dynamic Graph Convolutional Network (SDGCN) is proposed. Different from the existing dynamic GCN methods, SDGCN constructs dynamic graphs not from the node's signal information, but rather from trajectory information of traffic flow. This way it fully considers both semantic dynamic graph and node signals to make higher speed predictions. The State-share Sparse HMM method is presented to obtain the dynamic semantics-aware graphs based on traffic trajectories. The motivation or trip patterns behind the traffic flow can be captured through trajectories data. Also, the trend of traffic flow can be predicted to build a dynamic graph. Considering the complexity of calculation, we develop a new forward algorithm for the State-share Sparse HMM method in the predictive stage, and finally the high-order semantics-aware Laplacian matrices with low time computational complexity can be obtained. Experimental results demonstrate that SDGCN outperforms state-of-the-art traffic flow forecasting methods and also better interprets the generated Laplacian matrices. This way it can dynamically represent the trip patterns or motivation behind the traffic flows. However, considering that both trajectory and traffic speed information are required, it is difficult to obtain both datasets in some application scenarios.

In future work, we will further explore the dynamic relationships of traffic flow and extend the semantics-aware dynamic graph to deep GCN as well as the external factors to the dynamic graph construction work.

References

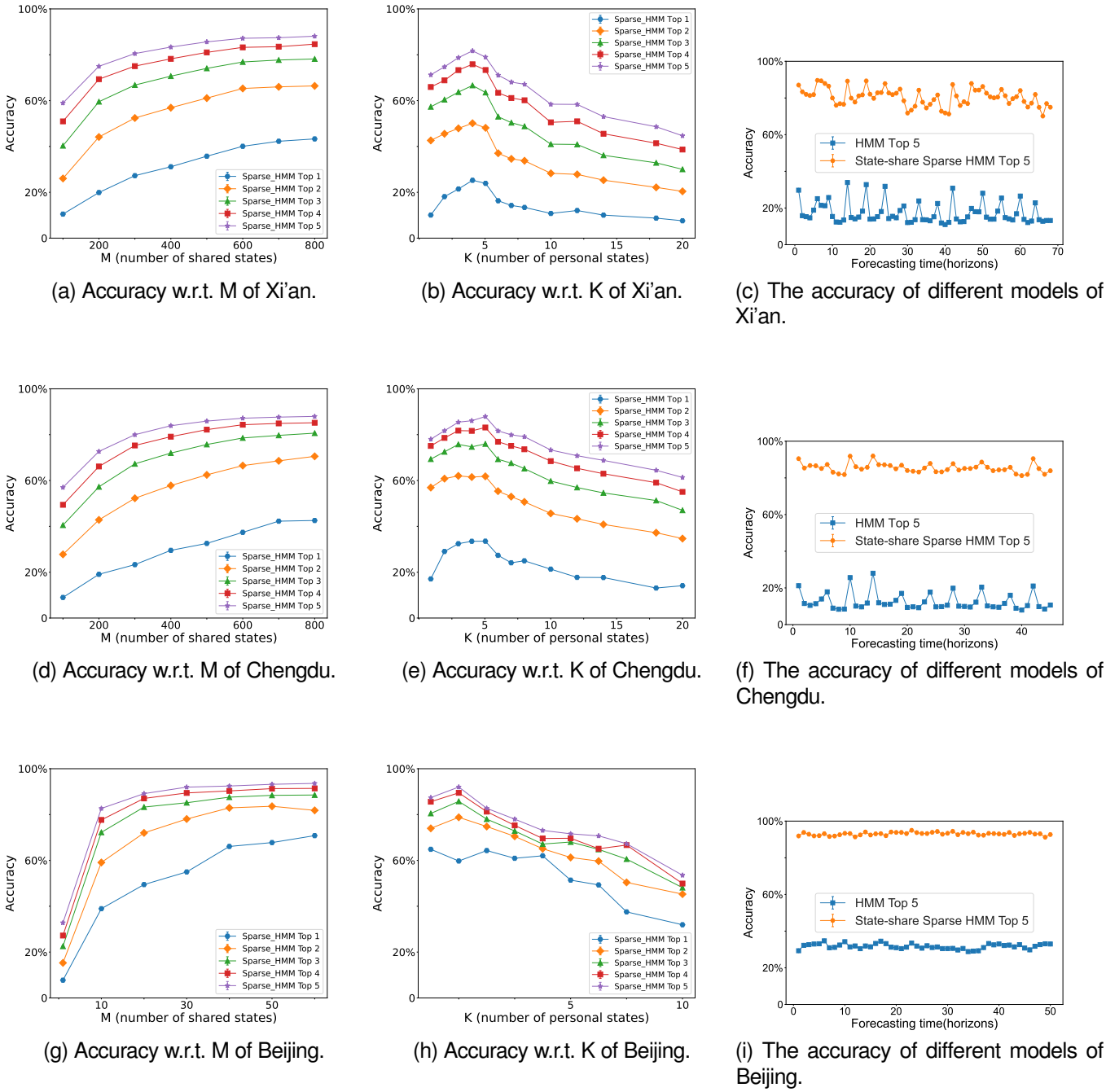
[1] G. Comert and A. Bezuglov, "An online change-point-based model for traffic parameter prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 3, pp. 1360–1369, 2013.

[2] B. Ghosh, B. Basu, and M. O'Mahony, "Multivariate short-term traffic flow forecasting using time-series analysis," *IEEE transactions on intelligent transportation systems*, vol. 10, no. 2, pp. 246–254, 2009.

[3] D. Chen, "Research on traffic flow prediction in the big data environment based on the improved rbf neural network," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 4, pp. 2000–2008, 2017.

[4] C. P. I. J. van Hinsbergen, T. Schreiter, F. S. Zuurbier, J. W. C. van Lint, and H. J. van Zuylen, "Localized extended kalman filter for scalable real-time traffic state estimation," *IEEE transactions on intelligent transportation systems*, vol. 13, no. 1, pp. 385–394, 2012.

[5] S. Sun and C. Zhang, "The selective random subspace predictor for traffic flow forecasting," *IEEE Transactions on intelligent transportation systems*, vol. 8, no. 2, pp. 367–373, 2007.



(a) Accuracy w.r.t. M of Xi'an.

(b) Accuracy w.r.t. K of Xi'an.

(c) The accuracy of different models of Xi'an.

(d) Accuracy w.r.t. M of Chengdu.

(e) Accuracy w.r.t. K of Chengdu.

(f) The accuracy of different models of Chengdu.

(g) Accuracy w.r.t. M of Beijing.

(h) Accuracy w.r.t. K of Beijing.

(i) The accuracy of different models of Beijing.

Fig. 8. The accuracies of different parameters and models of Xi'an, Chengdu and Beijing.

[6] A. Abadi, T. Rajabioun, and P. A. Ioannou, "Traffic flow prediction for road transportation networks with limited traffic data," *IEEE transactions on intelligent transportation systems*, vol. 16, no. 2, pp. 653–662, 2014.

[7] S. Sun, C. Zhang, and G. Yu, "A bayesian network approach to traffic flow forecasting," *IEEE Transactions on intelligent transportation systems*, vol. 7, no. 1, pp. 124–132, 2006.

[8] J. Haworth, J. Shawe-Taylor, T. Cheng, and J. Wang, "Local online kernel ridge regression for forecasting of urban travel times," *Transportation research part C: emerging technologies*, vol. 46, pp. 151–178, 2014.

[9] P. Dell'Acqua, F. Bellotti, R. Berta, and A. De Gloria, "Time-aware multivariate nearest neighbor regression methods for traffic flow prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 6, pp. 3393–3402, 2015.

[10] Y. Zhang, J. Li, Y. Guo, C. Xu, J. Bao, and Y. Song, "Vehicle driving behavior recognition based on multi-view convolutional neural network with joint data augmentation," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 5, pp. 4223–4234, 2019.

[11] D. A. Tedjopurnomo, Z. Bao, B. Zheng, F. Choudhury, and A. Qin, "A survey on modern deep neural network for

TABLE IV
STATISTICAL METRICS OF THE GENERATED LAPLACIAN MATRICES ON THREE DATASETS.

Dataset	Method	Total Number	Time	Non Zero Number	Non Zero Number Ratio
Xi'an	HMM	63001	0:00AM	5499	8.73%
			8:00AM	6066	9.63%
			6:00PM	8062	12.80%
	Sparse HMM	63001	0:00AM	14628	23.22%
			8:00AM	25372	40.72%
			6:00PM	33502	53.18%
Chengdu	HMM	255025	0:00AM	15883	6.23%
			8:00AM	19612	7.69%
			6:00PM	21024	8.24%
	Sparse HMM	255025	0:00AM	33965	13.32%
			8:00AM	60065	23.55%
			6:00PM	74881	29.36%
Beijing	HMM	4096	0:00AM	159	3.88%
			8:00AM	189	4.61%
			6:00PM	190	4.64%
	Sparse HMM	4096	0:00AM	315	7.69%
			8:00AM	379	9.25%
			6:00PM	312	7.62%

traffic prediction: Trends, methods and challenges," *IEEE Transactions on Knowledge and Data Engineering*, 2020.

[12] Y. Wu, H. Tan, L. Qin, B. Ran, and Z. Jiang, "A hybrid deep learning based traffic flow prediction method and its understanding," *Transportation Research Part C: Emerging Technologies*, vol. 90, pp. 166–180, 2018.

[13] F. Zhao, G.-Q. Zeng, and K.-D. Lu, "Enlstm-wpeo: Short-term traffic flow prediction by ensemble lstm, nnct weight integration, and population extremal optimization," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 1, pp. 101–113, 2019.

[14] G. Gui, Z. Zhou, J. Wang, F. Liu, and J. Sun, "Machine learning aided air traffic flow analysis based on aviation big data," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 5, pp. 4817–4826, 2020.

[15] M. Cao, V. O. Li, and V. W. Chan, "A cnn-lstm model for traffic speed prediction," in *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*. IEEE, 2020, pp. 1–5.

[16] W. Jiang and J. Luo, "Graph neural network for traffic forecasting: A survey," *arXiv preprint arXiv:2101.11174*, 2021.

[17] Y. Liang, Z. Zhao, and L. Sun, "Dynamic spatiotemporal graph convolutional neural networks for traffic data imputation with complex missing patterns," *arXiv preprint arXiv:2109.08357*, 2021.

[18] T. Xia, J. Lin, Y. Li, J. Feng, P. Hui, F. Sun, D. Guo, and D. Jin, "3dgcn: 3-dimensional dynamic graph convolutional network for citywide crowd flow prediction," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 15, no. 6, pp. 1–21, 2021.

[19] J. Hu and L. Chen, "Multi-attention based spatial-temporal graph convolution networks for traffic flow forecasting," in *2021 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2021, pp. 1–7.

[20] P. Veličković, A. Casanova, P. Lio, G. Cucurull, A. Romero, and Y. Bengio, "Graph attention networks." 6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings, 2018.

[21] S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan, "Attention based spatial-temporal graph convolutional networks for traffic flow forecasting," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 922–929.

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

[23] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," *arXiv preprint arXiv:1709.04875*, 2017.

[24] Z. Fang, Q. Long, G. Song, and K. Xie, "Spatial-temporal graph ode networks for traffic flow forecasting," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2021, pp. 364–373.

[25] S. Lan, Y. Ma, W. Huang, W. Wang, H. Yang, and P. Li, "Dstagnn: Dynamic spatial-temporal aware graph neural network for traffic flow forecasting," in *International Conference on Machine Learning*. PMLR, 2022, pp. 11 906–11 917.

[26] L. Li, R. Jiang, Z. He, X. M. Chen, and X. Zhou, "Trajectory data-based traffic flow studies: A revisit," *Transportation Research Part C: Emerging Technologies*, vol. 114, pp. 225–240, 2020.

[27] B. Lu, X. Gan, H. Jin, L. Fu, and H. Zhang, "Spatiotemporal adaptive gated graph convolution network for urban traffic flow forecasting," in *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, 2020, pp. 1025–1034.

[28] H. Peng, H. Wang, B. Du, M. Z. A. Bhuiyan, H. Ma, J. Liu, L. Wang, Z. Yang, L. Du, S. Wang *et al.*, "Spatial temporal incidence dynamic graph neural networks for traffic flow forecasting," *Information Sciences*, vol. 521, pp. 277–290, 2020.

[29] Z. Kang, H. Xu, J. Hu, and X. Pei, "Learning dynamic graph embedding for traffic flow forecasting: A graph self-attentive method," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2019, pp. 2570–2576.

[30] J. Liu and W. Guan, "A summary of traffic flow forecasting methods [j]," *Journal of highway and transportation research and development*, vol. 3, pp. 82–85, 2004.

[31] B. M. Williams and L. A. Hoel, "Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: Theoretical basis and empirical results," *Journal of Transportation*

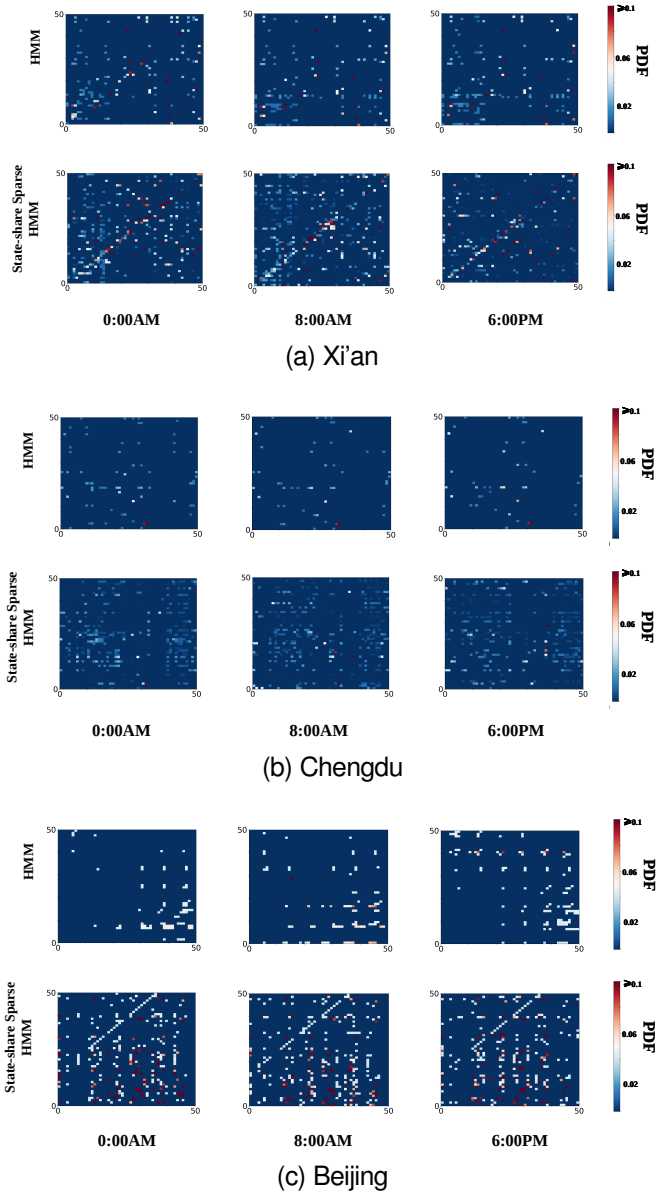


Fig. 9. Visualization of Laplacian Matrices generated by HMM and State-share Sparse HMM.

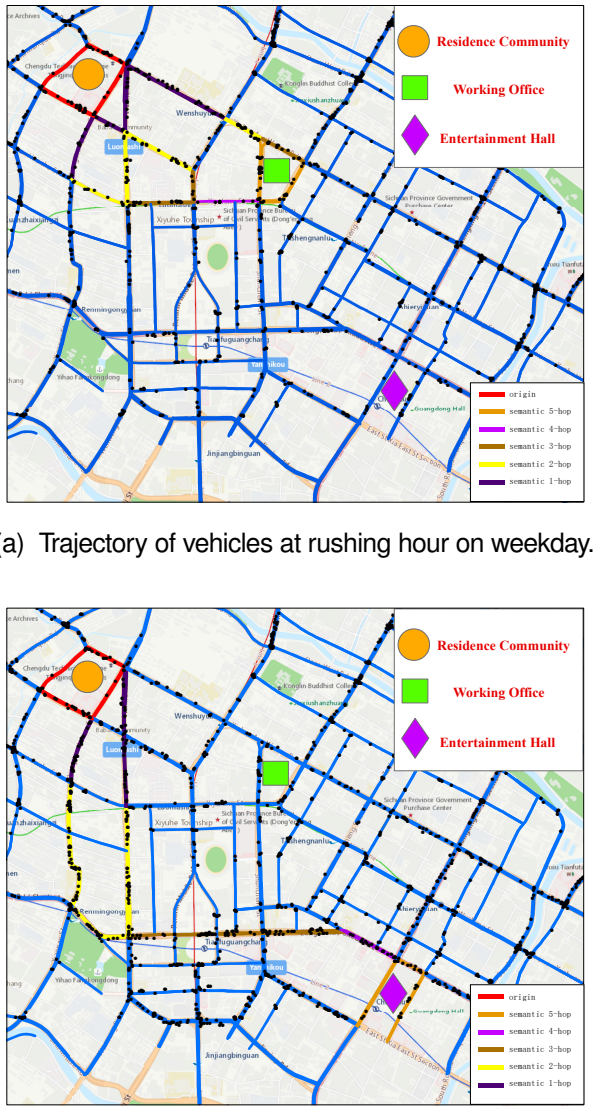


Fig. 10. Interpretation of real traffic graph.

Engineering, vol. 129, no. 6, pp. 664–672, nov 2003.

[32] E. Zivot and J. Wang, "Vector autoregressive models for multivariate time series," *Modeling Financial Time Series with S-Plus®*, pp. 385–429, 2006.

[33] J. Van Lint and C. Van Hinsbergen, "Short-term traffic and travel time prediction models," *Artificial Intelligence Applications to Critical Transportation Issues*, vol. 22, no. 1, pp. 22–41, 2012.

[34] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statistics and Computing*, vol. 14, no. 3, pp. 199–222, aug 2004.

[35] J. Li, D. Fu, Q. Yuan, H. Zhang, K. Chen, S. Yang, and F. Yang, "A traffic prediction enabled double rewarded value iteration network for route planning," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 5, pp. 4170–4181, 2019.

[36] P.-Y. Ting, T. Wada, Y.-L. Chiu, M.-T. Sun, K. Sakai, W.-S. Ku, A. A.-K. Jeng, and J.-S. Hwu, "Freeway travel time prediction using deep hybrid model–taking sun yat-sen freeway as an example," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 8, pp. 8257–8266, 2020.

[37] Y. Tang, N. Cheng, W. Wu, M. Wang, Y. Dai, and X. Shen, "Delay-minimization routing for heterogeneous vanets with machine learning based mobility prediction," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 4, pp. 3967–3979, 2019.

[38] D. Zhang and M. R. Kabuka, "Combining weather condition data to predict traffic flow: a GRU-based deep learning approach," *IET Intelligent Transport Systems*, vol. 12, no. 7, pp. 578–585, mar 2018.

[39] W. Fang, W. Zhuo, J. Yan, Y. Song, D. Jiang, and T. Zhou, "Attention meets long short-term memory: A deep learning network for traffic flow forecasting," *Physica A: Statistical Mechanics and its Applications*, vol. 587, p. 126485, 2022.

[40] X. Pan, F. Hou, and S. Li, "Traffic speed prediction based on time classification in combination with spatial graph

convolutional network," *IEEE Transactions on Intelligent Transportation Systems*, 2022.

[41] X. Wang, Y. Ma, Y. Wang, W. Jin, X. Wang, J. Tang, C. Jia, and J. Yu, "Traffic flow prediction via spatial temporal graph neural network," in *Proceedings of The Web Conference 2020*. ACM, apr 2020.

[42] M. Niepert, M. Ahmed, and K. Kutzkov, "Learning convolutional neural networks for graphs," in *International conference on machine learning*. PMLR, 2016, pp. 2014–2023.

[43] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral networks and locally connected networks on graphs," *arXiv preprint arXiv:1312.6203*, 2013.

[44] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," *Advances in neural information processing systems*, vol. 29, pp. 3844–3852, 2016.

[45] B. Yu, Y. Lee, and K. Sohn, "Forecasting road traffic speeds by considering area-wide spatio-temporal dependencies based on a graph convolutional neural network (GCN)," *Transportation Research Part C: Emerging Technologies*, vol. 114, pp. 189–204, may 2020.

[46] J. Atwood and D. Towsley, "Diffusion-convolutional neural networks," in *Advances in neural information processing systems*, 2016, pp. 1993–2001.

[47] F. Zhou, Q. Yang, T. Zhong, D. Chen, and N. Zhang, "Variational graph neural networks for road traffic prediction in intelligent transportation systems," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 4, pp. 2802–2812, 2020.

[48] Y. Wang, C. Jing, S. Xu, and T. Guo, "Attention based spatiotemporal graph attention networks for traffic flow forecasting," *Information Sciences*, vol. 607, pp. 869–883, 2022.

[49] L. Zhao, Y. Song, C. Zhang, Y. Liu, P. Wang, T. Lin, M. Deng, and H. Li, "T-GCN: A temporal graph convolutional network for traffic prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 9, pp. 3848–3858, sep 2020.

[50] K. Guo, Y. Hu, Z. Qian, Y. Sun, J. Gao, and B. Yin, "Dynamic graph convolution network for traffic forecasting based on latent network of laplace matrix estimation," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–10, 2020.

[51] L. Wu, P. Cui, J. Pei, L. Zhao, and L. Song, "Graph neural networks," in *Graph Neural Networks: Foundations, Frontiers, and Applications*. Springer, 2022, pp. 27–37.

[52] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[53] G. Li, M. Muller, A. Thabet, and B. Ghanem, "DeepGCNs: Can GCNs go as deep as CNNs?" in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, oct 2019.

[54] N. M. Nasrabadi, "Pattern recognition and machine learning," *Journal of electronic imaging*, vol. 16, no. 4, p. 049901, 2007.

[55] H. Shi, C. Zhang, Q. Yao, Y. Li, F. Sun, and D. Jin, "State-sharing sparse hidden markov models for personalized sequences," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 1549–1559.

[56] J. Yuan, Y. Zheng, C. Zhang, W. Xie, X. Xie, G. Sun, and Y. Huang, "T-drive: driving directions based on taxi trajectories," in *Proceedings of the 18th SIGSPATIAL International conference on advances in geographic information systems*, 2010, pp. 99–108.

[57] J. Yuan, Y. Zheng, X. Xie, and G. Sun, "Driving with knowledge from the physical world," in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2011, pp. 316–324.

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