An Enhanced Motif Graph Clustering-Based Deep Learning Approach for Traffic Forecasting

Chenhan Zhang, Shuyu Zhang, James J.Q. Yu
Department of Computer Science and Engineering
Southern University of Science and Technology
Shenzhen, China
Email: {zhangch, 11712122}@mail.sustech.edu.cn, yujq3@sustech.edu.cn

Shui Yu
School of Computer Science
University of Technology Sydney
Sydney, Australia
Email: Shui.Yu@uts.edu.au

***Abstract*—Traffic speed prediction is among the key problems in intelligent transportation system (ITS). Traffic patterns with complex spatial dependency make accurate prediction on traffic networks a challenging task. Recently, a deep learning approach named Spatio-Temporal Graph Convolutional Networks (STGCN) has achieved state-of-the-art results in traffic speed prediction by jointly exploiting the spatial and temporal features of traffic data. Nonetheless, applying STGCN to large-scale urban traffic network may develop degenerated results, which is due to redundant spatial information engaging in graph convolution. In this work, we propose a motif-based graph-clustering approach to apply STGCN to large-scale traffic networks. By using graph-clustering, we partition a large urban traffic network into smaller clusters to prompt the learning effect of graph convolution. The proposed approach is evaluated on two real-world datasets and is compared with its variants and baseline methods. The results show that graph-clustering approaches generally outperform the other methods, and the proposed approach obtains the best performance.**

***Index Terms*—smart city, intelligent transportation system, graph clustering, traffic speed prediction**

I. Introduction

In the past decade, the development of Intelligent Transportation System (ITS) has been increasing rapidly since it can significantly solve many traffic issues in modern cities [1]. Among the various problems in ITS, Traffic Speed Prediction (TSP) is among the fundamental ones, which plays a vital role in supporting ITS services such as route guidance, navigation and flow control [2].

Achieving accurate prediction has always been challenging due to many obstacles, such as the large traffic network size, the complex traffic conditions in the real world, and the massive traffic data with noise. To overcome these problems, researchers proposed various advanced models using advanced deep learning techniques [3]. The graph-based deep learning methods that process the traffic data on graphs have been seen in the emerging trend due to their successful application on TSP, which shed light on the traffic spatial feature exploiting [4]. Graph Convolution Network (GCN) is an important development of graph-based deep learning, which generalize convolution operation to the graph domain [3]. In TSP community, Spatio-Temporal Graph Convolutional Networks (STGCN) is one of the most advanced extensions of GCN-based models that have demonstrated outstanding performance [5]. However, for large-scale traffic network, massive irrelevant spatial information may be involved in graph convolution calculations which degenerate the model performance [6], [7]. In this paper, to address this problem, we propose a graph-clustering based approach to apply STGCN for large-scale urban TSP. By using graph clustering, a traffic graph can be partitioned into several subgraphs. Each subgraph is expected to contain less redundant spatial information where the GCN model can develop better performance.

The main contributions of this paper are as follows:

The remainder of this paper is organized as follows. Section II presents the development of TSP models. In Section III, we elaborate on the mechanism of the proposed graph-clustering approach and the overall architecture. A series of case studies are performed in Section IV to demonstrate the efficacy of the proposed mechanism. Finally, this paper is concluded in Section V with a discussion of potential future studies.

II. Literature Review

A significant amount of effort in traffic prediction domain have been made. In this section, we briefly summarize the literature related to this work.

With the rapid development of traffic data collection techniques, a significant portion of popular traffic prediction approaches are data-driven-based. Existing data-driven approaches can be generally divided into two main groups: parametric and non-parametric approaches [3]. Parametric approaches include Autoregressive Integrated Moving Average (ARIMA), Kalman Filter and their extensions, which have been widely applied in the time series community [8]. Compared with parametric approaches, non-parametric approaches are more flexible and sophisticated since their structures and parameters are not fixed, where K-Nearest Neighbor (KNN), Support Vector Regression (SVR) and Neural Networks (NN) are among the most representative ones. Their applications in traffic prediction have shown outstanding performance, especially in handling non-linear traffic data [9], [10].

This work was supported by the General Program of Guangdong Basic and Applied Basic Research Foundation No. 2019A1515011032, by the Department of Education of Guangdong Province, China No. SJJG201901, and by the Guangdong Provincial Key Laboratory No. 2020B121201001. James J.Q. Yu is the corresponding author.

Deep Learning (DL) has been a popular technique for traffic prediction in the past few years [11]. The "black box" nature of DL models renders them more robust to data noise and perturbations when compared with classical statistical methods [12]. Long Short-Term Memory networks (LSTM) and Gated Recurrent Unit (GRU) are two variants of Recurrent Neural Networks (RNN). In traffic prediction field, they were applied to learn temporal dependency in traffic data to improve the model performance, especially on mid- and long-term prediction [13]. Convolution Neural Networks (CNN) have witnessed an overwhelming success in computer vision. Some researchers adopt CNN to explore spatial correlation among the traffic network [14]. Moreover, to jointly learn both spatial and temporal features, a group of hybrid models that integrate both CNN and RNN are introduced in the literature [15], [16]. However, CNN is restricted to only process the regular grid-like data such as images, while the traffic data sampled in the road network (which is in the non-Euclidean space) is irregular. In some of the current approaches, the traffic data is sampled as grids to drive CNN, which may result in loss of the spatial information [17]. To overcome this problem, researchers start to adopt graph-based deep learning approaches.

Graph Convolutional Network (GCN) is one of the most prominent graph-based deep learning approaches [18], which aims to generalize CNN to graphs. It has been introduced in many graph-based applications, including urban traffic prediction. Li et al. [19] proposed a hybrid GCN-based model, Diffusion Convolutional Recurrent Neural Network (DCRNN), that captures the spatial dependency with random walks on the traffic network. Different from the spectral-based approaches used in [19], Cui et al. [20] proposed a traffic flow prediction model based on non-spectral GCN. Yu et al. [5] proposed a Spatio-Temporal Graph Convolutional Network (STGCN) that apply convolutional structures on both spatial and time axis, which enables faster model training speed and convergences. Moreover, STGCN employs fewer parameters to achieve better scalability. These traits make STGCN grossly practical on large-scale traffic network traffic prediction.

III. Method

A. *Traffic Speed Prediction Problem on Road Graphs*

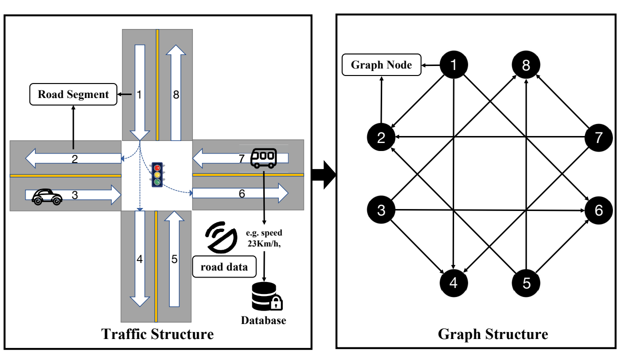


Fig. 1: Graph representation of road network

In this work, we represent the traffic networks as graphs to focus on the spatial structure of traffic data. In particular, different from the typical graph embeddings of traffic network, which models the road segments as edges and their crossroads as nodes, we regard the road segments as the graph nodes in this representation. The traffic speed observation of each road segment, x_t, is not independent but correlating pairwise in the graph. A directed edge is generated if there exists a connection between any two road segments. Fig. 1 shows an example of this graph representation, where eight road segments are labelled from 1 to 8, and they are defined as the nodes of the graph. For node 1 (i.e. road segment 1), there are three nodes, i.e., 2, 4, 6, connecting to it directly. Therefore, there are edges (1, 2), (1, 4), (1, 6), respectively. Furthermore, for bidirectional roads, we define the two directions as two nodes (e.g. nodes 2 and 3).

Therefore, we can represent the traffic network as a weighted directed traffic graph $G_t = (V_t, E, W)$ at time step t, where V_t is a finite set of nodes $|V_t| = N$, E is the set of edges and W is the weighted adjacency matrix of the graph. $X_t = \left\{x_t^1, \ldots, x_t^N\right\}$ denotes the graph signal including traffic observations from the N road segments at time t. The traffic speed prediction on it can be represented by

$$[X_{t-T+1}, ..., X_t; \mathscr{E}; G_t] \xrightarrow{f(\cdot)} [\hat{X}_{t+1}, ..., \hat{X}_{t+H}], \quad (1)$$

where $f(\cdot)$ is the prediction function that the model aims to learn to predict the traffic speed.

B. *Enhanced Motif-based Graph Clustering*

In this subsection, we first introduce a basic motif-based graph-clustering method. Subsequently, an enhanced method based on this motif-based graph-clustering method is elaborated.

1) Motif-based Graph Clustering: First, we construct a motif-based hypergraph based on the original graph. Motif is a high-order connectivity pattern of the graph. Given a graph $G = (V, E, W)$, the motif-based hypergraph can be represented by

$$G^M = (V, E^M, W^M), \quad (2)$$

where V is the node set which is the same as the original graph, W^M is the motif adjacency matrix (note that we choose four-node motif [21] to follow the similar connection pattern as shown in Fig. 1), E^M is the edge set containing m weighted edges which is generated by

$$E^M = ((i, j, \omega)_k), \quad (3)$$

where $i, j \in V$ are two nodes connected by edge i ($i \in \mathbb{R}^m$) and ω is the weight of the edge i. Subsequently, we can identify a group of connected motif-like subsets of the hypergraph as

$$\Psi = \{\psi_1, \ldots, \psi_p\}, \quad (4)$$

where ψ is the motif-like subset and p is the number of the subsets. Filtering out the isolated nodes in the hypergraph, we obtain the top Q largest connected subsets $\Psi_Q \subseteq \Psi$ where

$Q < p$ (note that we set $Q = 4$ referring to [22] to obtain the best results in our tests). Then, Metis [23] is used to partition these subsets into different communities. Gathering the partitioning results of all the top Q largest subsets, we harvest a community set $M = \{M_1, \ldots, M_{p_m}\}$ where p_m is the number of communities.

2) Edge Enhancement for Traffic Data: The above motif-based graph clustering approach can reveal only higher-order community topology of the graph since only the top Q largest connected subsets are considered in the graph partition. The subsets or isolated nodes which are not connected to these largest connected subsets are not taken into account.

Considering the compact connections among different nodes (road segments) in real traffic networks, it is expected to completely involve all the nodes when processing corresponding traffic graphs, which may encode more influential information of the traffic data. Therefore, we introduced an edge enhancement approach based on [22] to reconstruct the graph. First, we intensify the connectivity among the nodes which have already been clustered into the same cluster. For cluster $M_i \in M$, an edge is constructed pairwisely for all nodes in M_i. By this procedure, a new group of edges E_{new} is generated. Now, the nodes in each cluster are interconnected in a strong pattern which is difficult to be fragmented by partitioning. Then, to consider the isolated nodes, we rewire the graph by involving both the new generated edge set E_{new} and the original edge set E. In this way, a rewired graph can be obtained by

$$G^M_{rewired} = (V, E^M_{rewired}, W^M_{rewired})), \tag{5}$$

where $E^M_{rewired} = E \cup E^M_{rewired}$ is the edges set of the rewired graph and $W^M_{rewired}$ is its adjacency matrix.

Subsequently, we use Metis method again to partition the rewired graph $G^M_{rewired}$. We define each of the new partitioned communities as a subgraph. Finally, a set of subgraphs is obtained as $\{{G^*}_1, ..., {G^*}_S\}$ where S is the number of subgraphs.

To summarize, the procedure of the proposed graph-clustering approach is presented as below:

1) Convert the original graph G into a motif-based hypergraph G^M.
2) Identify a group of motif-like subsets Ψ and obtain the top Q largest subsets Ψ_Q.
3) Partition each subset $\psi \in \Psi_Q$ by Metis and obtain a community set M.
4) Interconnect each pair of nodes in $M_i \in M$ and rewiring isolated nodes to M_i by following the original graph to construct the new graph $G^M_{rewired}$.
5) Partition $G^M_{rewired}$ into S clusters to obtain the final subgraph set $\{{G^*}_1, ..., {G^*}_S\}$.

C. Spatio-Temporal Graph Convolutional Network

In this work, we apply our graph-clustering approach to Spatio-Temporal Graph Convolutional Network (STGCN) model. STGCN model is composed of several spatial blocks and temporal data processing blocks, which are used to exploit spatial and temporal dependency, respectively.

The spatial block employs a graph convolution operation which follows the spectral GCN method that convolves in the spectral domains. To process the graph data in the spectral domain, we first compute the normalized graph Laplacian matrix given the adjacency matrix W as

$$L = I_N - D^{-\frac{1}{2}} W D^{-\frac{1}{2}}, \tag{6}$$

where $I_N \in \mathbb{R}^{N*N}$ is the identity matrix and $D = diag\left(\Sigma_j W_{ij}\right) \in \mathbb{R}^{N*N}$ is the diagonal degree matrix. Subsequently, we decompose L and have $L = U \Lambda U^T$ where $U \in \mathbb{R}^{N*N}$ is the matrix of eigenvectors of L and $\Lambda \in \mathbb{R}^{N*N}$ is the diagonal matrix of eigenvalues λ with $\Lambda = diag\left(\lambda\right)$. Then we can operate convolution in the spectral domain the graph by introducing a graph convolution operator $*$ as

$$g_\theta * X = g_\theta\left(U \Lambda U^T\right) X = U g_\theta\left(\Lambda\right) U^T X, \tag{7}$$

where g_θ is the kernel with convolution parameter $\theta \in \mathbb{R}^N$. By this filtering operation, an updated feature of the graph signal X can be computed by the multiplication between g_θ and $U^T X$. Moreover, to cut down the computational expense of the decomposition process of the normalized Laplacian matrix in (7), an approximation model is further constructed by utilizing Chebyshev polynomials referring to [24]. Finally, the graph convolution model can be formed as

$$X' = \sigma\left(\widetilde{D}^{-\frac{1}{2}} \widetilde{W} \widetilde{D}^{-\frac{1}{2}} X \theta\right), \tag{8}$$

where X' is the updated graph signal, $\widetilde{W} = W + I_N$ is the adjacency matrix with added self-connections, $\widetilde{D} = diag\left(\Sigma_j \widetilde{W}_{ij}\right)$ is the diagonal degree matrix, θ is the shared parameter, and $\sigma\left(\cdot\right)$ is the sigmoid function.

The temporal block incorporates a 1-D CNN and a Gated Linear Units (GLU). A gated structure is adopted since it can control the input of relevant dynamic variances, which may contribute to the performance on long-term prediction. The temporal gated convolution procedure can be formulated as

$$X'_\tau = \left(\Theta\left(X_\tau\right) + X_\tau\right) \odot \sigma\left(\Theta\left(X_\tau\right)\right), \tag{9}$$

where X_τ and X'_τ are the input and updated graph signals of the temporal convolution layer, respectively; $\Theta(\cdot)$ denotes the convolution operation, and $\odot$ denotes the Hadamard product.

D. Framework of EMGC-STGCN

In this paper, we leverage a graph-clustering method to divide the traffic graph into multiple subgraphs. Thus, a large task can be solved by solving several subtasks. We adopt a multiple-model training strategy to handle the subtasks. For example, if S subgraphs are obtained by graph-clustering, they will be fed into S independent models, and each model will be trained independently and parallelly. The framework of our proposed approach is shown in Fig. 2.

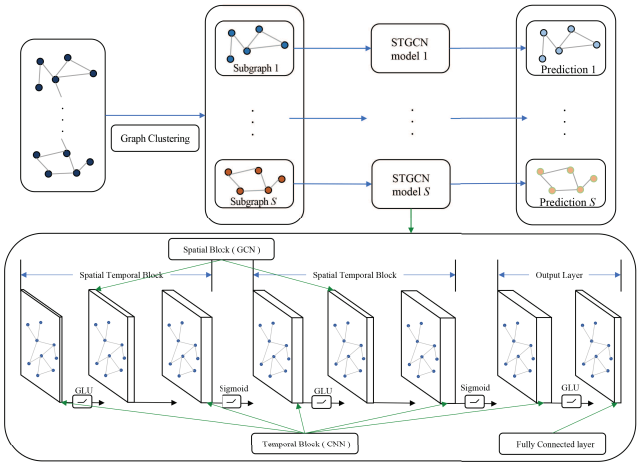


Fig. 2: The framework of the proposed EMGC-STGCN

1) Objective Function: In each training process, the aim is to minimize the error between the prediction value of the traffic speed and the ground truth. We use Mean Squared Error (MSE a.k.a. L2 loss) as the loss function of the training as follows:

$$\min L\left(\hat{Y}_t;\theta\right)=\sum_t\left\|\hat{Y}_t-Y_t\right\|, \tag{10}$$

where Y_t and $\hat{Y}_t$ are the ground truth and predictions, respectively; θ denotes the involved learnable parameters.

IV. Experiments

A. System Configuration

1) Dataset Description: In this work, two real-world network-wide traffic speed datasets are utilized. The first dataset **Nav-BJ** is from NavInfo [1], which contains data collected from 1159 sensor stations deployed in the different road segments of Beijing city. The period of this dataset is from March 1st to March 31st of 2019. The second dataset is **PeMSD7(M)** which is a public dataset collected from Caltrans Performance Measurement System (PeMS) by 228 sensor stations in District 7 of California. The data collection period is from May 1st to June 30th of 2012 (without weekends). In both of the datasets, we aggregate traffic speed observations into 5-minute interval and apply Z-Score normalization to the data. Additionally, we apply linear interpolation to recover missing data points. The training, testing, and validation sets are correspondingly generated, each of which contains 60%, 20%, and 20% of all data.

The adjacency matrices of the two datasets are constructed in two different ways. In Nav-BJ, the topology of the road graph follows the diagram as shown in Fig. 1, which is a typical urban traffic pattern. Furthermore, the generated graph is directed since each road segment has a attribute of origin-destination. In PeMSD7(M), there is no topological connections in the original graph. Therefore, the adjacency matrix is generated based on the Euclidean space distances among sensor stations using thresholded Gaussian kernel method [25]. This procedure can be represented by

$$w_{ij}=\begin{cases}1,\ \text{if } i\neq j \text{ and } \exp\left(-\dfrac{\operatorname{dist}(i,j)}{\sigma^2}\right)\geqslant\varepsilon \\ 0,\ \text{otherwise.}\end{cases}, \tag{11}$$

where w_{ij} represents the edge weight between node i and node j, $\operatorname{dist}(i,j)$ denotes the Euclidean space distance from node i to node j; ε and σ^2 are user-controlled thresholds that control the sparsity of the matrix, whose values are 10 and 0.5 in the tests.

2) Compared Methods: To verify the effectiveness of our proposed approach, we compare our approach with the following baselines: (1) HA: Historical Average; (2) ARIMA: Auto-Regressive Integrated Moving Average model; (3) GRU: A GRU-based model [26]. Additionally, the variants of our proposed approach are also introduced to verify the efficacy of our proposed graph-clustering method, including (4) **STGCN**: Our proposed approach without graph-clustering (i.e. original STGCN model in [5]); (5) **Random-STGCN**: Our proposed approach using a random graph-clustering method; (6) **Metis-STGCN**: Our proposed approach using naive Metis method. These methods are evaluated by three metrics, namely, Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE) and Root Mean Squared Errors (RMSE).

3) Experiment Setting: All neural network-based approaches are developed using PyTorch and trained using Adam optimizer for 50 epochs with a learning rate of $1e^{-4}$. We adopt grid search strategy to locate the best hyperparameters on the validation dataset. The past time window is an hour (i.e. 12 observed data points), and the covered historical data are used to predict the traffic speed in the next 15, 30, and 45 minutes.

B. Experiment Results

Table I demonstrates the results of our proposed approach and the aforementioned baselines on Nav-BJ. With simpler frameworks compared to advanced deep learning models, it is not surprising to see that HA, ARIMA and GRU have larger prediction errors. ARIMA performs well on short-term prediction (i.e. 15 minutes); however, poor on long-term prediction (i.e. 45 minutes). Comparatively, GRU presents an excellent performance on relatively long-term prediction due to its RNN nature on temporal dependency learning. STGCN generally has better performance compared to the other baselines, which is explained by its spatial feature learning powered by GCN. Three graph-clustering-based approaches considerably surpass the methods above. Notably, our proposed approach EMGC-STGCN obtains the best results, whose MAPE outperforms STGCN by 1.61% (15 min), 1. 58% (30 min), and 1.62% (45 min). It implies that graph-clustering methods, especially our proposed enhanced motif-based graph-clustering approach, are effective for STGCN to learn spatial dependency of traffic graph to further improve its prediction capacity. For reference of the audience, the results of the graph-clustering are visualized as shown in Fig. 3, where we demonstrate four

[1] http://www.nitrafficindex.com

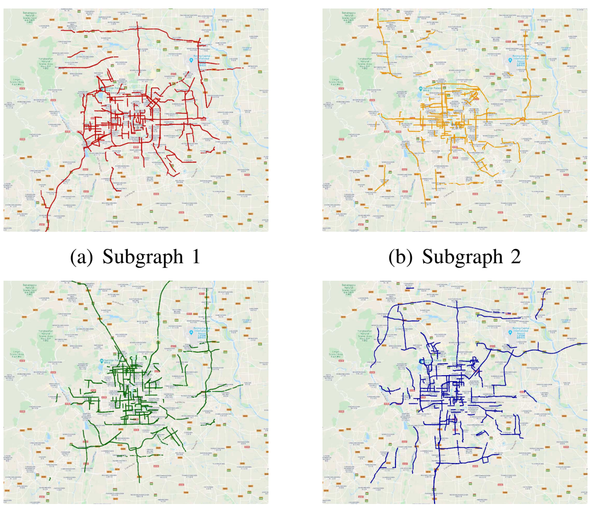
TABLE I: Performance comparison of EMGC-STGCNN and baselines on Nav-BJ dataset.

Model	15 min			30 min			45 min		
	RMSE	MAE	MAPE (%)	RMSE	MAE	MAPE (%)	RMSE	MAE	MAPE (%)
HA	7.54	4.19	12.28	7.54	4.19	12.28	7.54	4.19	12.28
ARIMA	8.03	4.39	10.88	12.21	5.91	12.66	19.07	8.85	18.56
GRU	7.34	4.02	13.56	7.61	4.23	14.32	7.56	4.32	14.67
STGCN	4.53	3.10	11.13	4.84	3.28	11.84	5.02	3.40	12.30
Random-STGCN	4.23	2.89	9.90	4.48	3.05	10.60	4.68	3.19	11.36
Metis-STGCN	4.20	2.84	9.72	4.41	3.02	10.38	4.56	3.14	10.97
EMGC-STGCN	**4.11**	**2.81**	**9.52**	**4.33**	**2.99**	**10.26**	**4.46**	**3.08**	**10.68**

TABLE II: Performance comparison of EMGC-STGCNN and baselines on PeMSD7(M) dataset.

Model	15 min			30 min			45 min		
	RMSE	MAE	MAPE (%)	RMSE	MAE	MAPE (%)	RMSE	MAE	MAPE (%)
HA	7.20	4.01	10.61	7.20	4.01	10.61	7.20	4.01	10.61
ARIMA	9.00	5.55	12.92	9.13	5.86	13.94	9.38	6.27	15.20
GRU	4.15	2.35	7.25	5.36	3.04	9.12	6.19	3.52	10.14
STGCN	3.55	2.02	4.82	4.91	2.85	7.10	5.45	3.14	7.67
Random-STGCN	3.52	2.06	4.74	4.73	2.67	6.51	5.37	3.04	7.52
Metis-STGCN	3.49	1.99	4.67	4.59	2.58	6.19	5.34	3.03	7.45
EMGC-STGCN	**3.47**	**1.98**	**4.56**	**4.62**	**2.58**	**6.20**	**5.40**	**3.03**	**7.42**

subgraphs among all the eight subgraphs. The colored lines denote the trajectories of all the road segments included in a subgraph.



(a) Subgraph 1

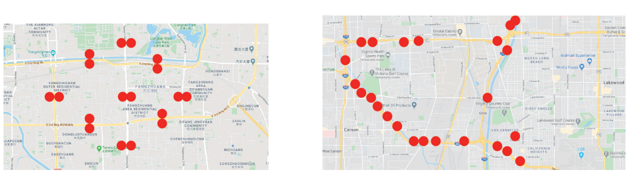
(b) Subgraph 2

(c) Subgraph 3

(d) Subgraph 4

Fig. 3: The partitioned results of our graph-clustering approach

C. Approach Scalability



(a) Nav-BJ

(b) PeMSD7(M)

Fig. 4: The diagram of the distribution of sensor stations

To verify the generalization ability of our proposed approach, we additionally test its performance on PeMSD7(M). Table II presents the results on PeMSD7(M). Overall, we can observe that the results demonstrate a similar pattern as shown in Table I. However, the performance gap between our proposed approach (i.e. EMGC-STGCN) and other graph-clustering-based approaches is not significant as significant as on Nav-BJ.

The aforementioned results may be credited to the following factors. First, the number of nodes in PeMSD7(M) (228) is much less than that of Nav-BJ (1159). Second, as shown in Fig. 4, the distribution of sensor stations (a.k.a. nodes) in the two datasets show different patterns. Recall the graph representation we define in Fig. 1, one may notice that it is more adapted to the dataset collected from some regular urban traffic networks (e.g. Nav-BJ). To conclude, the similar results imply that the performance improvement of graph-clustering is related to the size and traffic pattern of the historical dataset. A further study on their correlations will be conducted in the future.

D. Sensitivity of the Number of Clusters

We test the performance of our approach with four different numbers of clusters on dataset Nav-BJ. Specifically, these values are set to 2, 4, 8 (default) and 16. Table III demonstrates the result. We can observe that the best result is obtained using the default setting (i.e. $S = 8$). The performance degeneration exists but not significant. It can be concluded that EMGC-STGCN is not quite sensitive to the number of cluster, yet fine-tuning it can further improve the model performance.

V. Conclusion

In this paper, we propose an enhanced motif-based graph-clustering approach for large-scale urban traffic speed prediction. By adopting the proposed graph-clustering approach, we partition the large urban traffic network into a group of small networks and trained each of them on an independent STGCN model. To evaluate the performance of the proposed approach, we conduct several case studies on two real-world datasets.

TABLE III: Performance comparison of the number of clusters on Nav-BJ.

S^{a}	15 min			30 min			45 min		
	RMSE	MAE	MAPE (%)	RMSE	MAE	MAPE (%)	RMSE	MAE	MAPE (%)
2	4.21	2.83	9.65	4.48	3.04	10.60	4.66	3.16	11.39
4	4.16	2.81	9.58	4.48	3.03	10.55	4.59	3.13	11.12
8 (default)	**4.11**	**2.81**	**9.52**	**4.33**	**2.99**	**10.26**	**4.46**	**3.08**	**10.68**
16	4.22	2.84	9.55	4.40	3.00	10.35	4.54	3.09	10.78

[a] S denotes the number of clusters.

We first compare the prediction accuracy performance among our proposed approach, its variants and baseline methods on Nav-BJ dataset. The results show that our proposed approach obtains the best performance. To verify the effectiveness of the proposed approach, the PeMSD7(M) dataset is also employed, whose result shows the merit of graph-clustering. Lastly, we investigate the sensitivity of our proposed approach to the number of clusters (subgraphs).

In the future, we plan to further investigate the relevance of traffic patterns with our approach. Additionally, apart from STGCN, more state-of-the-art GCN-based approaches will be adopted to improve both short-term and long-term urban traffic speed prediction performance.

REFERENCES

[1] T. Mallick, P. Balaprakash, E. Rask, and J. Macfarlane, "Graph-partitioning-based diffusion convolution recurrent neural network for large-scale traffic forecasting," *arXiv preprint arXiv:1909.11197*, 2019.

[2] M. Veres and M. Moussa, "Deep learning for intelligent transportation systems: A survey of emerging trends," *IEEE Transactions on Intelligent Transportation Systems*, 2019, in press.

[3] Y. Wang, D. Zhang, Y. Liu, B. Dai, and L. H. Lee, "Enhancing transportation systems via deep learning: a survey," *Transportation research part C: emerging technologies*, vol. 99, pp. 144–163, 2018.

[4] T. Jia and P. Yan, "Predicting citywide road traffic flow using deep spatiotemporal neural networks," *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[5] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting," in *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pp. 3634–3640, 2018.

[6] H. Yao, X. Tang, H. Wei, G. Zheng, and Z. Li, "Revisiting spatial-temporal similarity: A deep learning framework for traffic prediction," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 5668–5675, 2019.

[7] C. Zhang, J. J. Yu, and Y. Liu, "Spatial-temporal graph attention networks: A deep learning approach for traffic forecasting," *IEEE Access*, vol. 7, pp. 166246–166256, 2019.

[8] J. Guo, W. Huang, and B. M. Williams, "Adaptive kalman filter approach for stochastic short-term traffic flow rate prediction and uncertainty quantification," *Transportation Research Part C: Emerging Technologies*, vol. 43, pp. 50–64, 2014.

[9] B. Yu, X. Song, F. Guan, Z. Yang, and B. Yao, "k-nearest neighbor model for multiple-time-step prediction of short-term traffic condition," *Journal of Transportation Engineering*, vol. 142, no. 6, p. 04016018, 2016.

[10] X. Luo, D. Li, and S. Zhang, "Traffic flow prediction during the holidays based on dft and svr," *Journal of Sensors*, vol. 2019, 2019.

[11] A. M. Nagy and V. Simon, "Survey on traffic prediction in smart cities," *Pervasive and Mobile Computing*, vol. 50, pp. 148–163, 2018.

[12] J. J. Yu, "Online traffic speed estimation for urban road networks with few data: A transfer learning approach," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pp. 4024–4029, IEEE, 2019.

[13] Y. Liu, J. J. Yu, J. Kang, D. Niyato, and S. Zhang, "Privacy-preserving traffic flow prediction: A federated learning approach," *arXiv preprint arXiv:2003.08725*, 2020.

[14] 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.

[15] J. Zhang, Y. Zheng, and D. Qi, "Deep spatio-temporal residual networks for citywide crowd flows prediction," in *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[16] Y. Jia, J. Wu, M. Ben-Akiva, R. Seshadri, and Y. Du, "Rainfall-integrated traffic speed prediction using deep learning method," *IET Intelligent Transport Systems*, vol. 11, no. 9, pp. 531–536, 2017.

[17] 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.

[18] M. Henaff, J. Bruna, and Y. LeCun, "Deep convolutional networks on graph-structured data," *arXiv preprint arXiv:1506.05163*, 2015.

[19] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," *arXiv preprint arXiv:1707.01926*, 2017.

[20] Z. Cui, K. Henrickson, R. Ke, and Y. Wang, "Traffic graph convolutional recurrent neural network: A deep learning framework for network-scale traffic learning and forecasting," *IEEE Transactions on Intelligent Transportation Systems*, 2019.

[21] Z. Ning, L. Liu, S. Yu, and F. Xia, "Detection of four-node motif in complex networks," in *International Conference on Complex Networks and their Applications*, pp. 453–462, Springer, 2017.

[22] P.-Z. Li, L. Huang, C.-D. Wang, and J.-H. Lai, "Edmot: An edge enhancement approach for motif-aware community detection," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 479–487, 2019.

[23] G. Karypis and V. Kumar, "A fast and high quality multilevel scheme for partitioning irregular graphs," *SIAM Journal on scientific Computing*, vol. 20, no. 1, pp. 359–392, 1998.

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

[25] 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 processing magazine*, vol. 30, no. 3, pp. 83–98, 2013.

[26] R. Fu, Z. Zhang, and L. Li, "Using LSTM and GRU neural network methods for traffic flow prediction," in *2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, pp. 324–328, IEEE, 2016.