GraphTTE: Travel Time Estimation Based on Attention-Spatiotemporal Graphs

Qiang Wang, *Member, IEEE*, Chen Xu, Wenqi Zhang, and Jingjing Li

***Abstract*—This letter proposes a new travel time estimation model based on graph neural network (GraphTTE) to improve the accuracy of travel time estimation. We design a Multi-layer Spatiotemporal Graph frame (MSG), which consists of static network and dynamic networks, to fully consider the influence of traffic temporal characteristics and road network topological characteristics on travel time. Moreover, we design an Attention Graph Nodes Impact Index algorithm (AGNII) to score the impact of each node on travel time. In particular, the dynamic networks utilize the graph convolution network and gate recurrent unit to obtain the traffic characteristics, the static network utilizes graph convolution network to obtain the road basic attributes. We combine the real paths sequence with the impact score of nodes to extract the subgraph with a great impact on the trajectory. After the graph representation learning and deep residual network, the estimated time is obtained. A simulator was designed to train and test our model in Chengdu and Xi'an datasets, the results show that the mean absolute percent error (MAPE) is 12.58% and 14.01%, which is 1.54% and 1.78% lower than the baselines.**

***Index Terms*—Attention mechanism, gate recurrent unit (GRU), graph convolutional network (GCN), spatiotemporal graph, travel time estimation.**

I. Introduction

ABUNDANT and accurate sensor signals have promoted the development of intelligent transportation systems (ITS) [1], [2]. Meanwhile, accurate travel time estimation [3], [4] [5] is the more concerned part of ITS. Nowadays, the increasing number of cars and complex road network increase the uncertainty of travel time and make it more difficult to estimate. Therefore, a more accurate travel time estimation method is of great significance for society.

In recent years, many contributions for travel time estimation have emerged, but it is very challenging to achieve high estimation accuracy. The connectivities of the road network and different time periods have a great impact on the road traffic condition, which leads to uncertainty of travel time. The basic characteristics of the road, such as the number of lanes, also affect the travel time. Besides, the multiple connectivities of roads result in the travel time being affected by both the traffic condition of the road on which the vehicle is located and its adjacent roads.

At present, there are two main ways to estimate travel time. One is path-based approaches [6], [7]. In [8], the Matrix Factorization techniques were utilized to estimate the travel time, which can estimate the travel time of all roads at different time periods. Some works utilized Bayesian networks [9] and designed a framework that combines Gaussian copulas and network inference to estimate the distributions of travel times. In [10], the work solved the traffic delays, which include congestion time and stopping time at signalized intersections, for each road on the trajectory. In [4], [5], each road was represented as a vector, and the adjacent relation of the road was reflected by vector correlation, the travel time was obtained by multi-task learning combined with other characteristics. The piecewise calculation method of travel time is intuitive, but the error increases with the length of the travel. They do not match the road network to the specific trajectory and lack the consideration of other roads besides the trajectory.

The other is trajectory-based approaches [11]. Some of them only utilized the origin point and destination of the travel to form origin-destination (OD) pairs and estimate the time by nearest neighbors matching [12]. In [13], the work added other factors on the basis of OD pair and proposed a Wide-Deep-Recurrent (WDR) learning model to predict travel time. In [14], the works utilized convolutional neural networks to process map image information of the trajectory and used recurrent neural networks to obtain the travel time. In [15], ConSTGAT integrated traffic conditions and contextual information of the trajectory to estimate travel time. In [16], the multi-task learning model MTLM integrated transportation-mode recommendation and travel time estimation to recommends the appropriate transportation mode and estimated the travel time. In [17], DeepOD embedded the road segments, time periods and other characteristics, and estimated travel time by the embedding vectors and spatio-temporal representation of the trajectory. These methods overcome the shortcomings of error accumulation and some of them consider the topology of the road network, but they do not extract the real road network subgraph associated with the trajectory and lack the consideration of other roads besides the trajectory.

In this letter, we propose a multi-layer travel time estimation model based on the graph neural network (GraphTTE), which takes into account simultaneously the temporal characteristics and the impact of road network topology on traffic, to get a more accurate travel time estimation. In particular, we map the road network to graphs which can effectively reflect its connectivity, so it is more convenient to study its topology structure. Besides, to enlarge the sensitivity field of the model and consider

Manuscript received October 27, 2020; revised December 24, 2020; accepted December 27, 2020. Date of publication January 5, 2021; date of current version February 5, 2021. This work was supported in part by the Beijing Municipal Natural Science Foundation Haidian Joint under Grant L182037 and in part by the Beijing Municipal Science and Technology Project under Grant Z181100003218015. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Jianxin Li. *(Corresponding author: Qiang Wang.)*

The authors are with the National Engineering Laboratory for Mobile Network, School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China (e-mail: wangq@bupt.edu.cn; xuchen@bupt.edu.cn; zwqximo@bupt.edu.cn; li961121@bupt.edu.cn).

Digital Object Identifier 10.1109/LSP.2020.3048849

the impact of nodes around the trajectory, we utilize attention mechanism to select other influential nodes besides the trajectory nodes. The attention mechanism selects the nodes by calculating the impact score according to the node information, thus making the model analysis scope not limited to the trajectory itself. The details of the contributions of this letter are as follows:

1) We designed a Multi-layer Spatiotemporal Graph frame (MSG), which consists of static network layer and dynamic network layers, to comprehensively consider the impact of time periods, road network topology and basic road attributes on traffic. The dynamic network layers of the frame are extensible, so we can increase the number of layers to process different temporal characteristics.
2) We proposed an Attention Graph Node Impact Index algorithm (AGNII) to score the impact of each node on travel time. The algorithm considers each node's own attributes and the network connectivity between nodes and trajectory, it scores the nodes through the graph convolutional neural network to obtain the node's impact index on the travel time.
3) We designed a simulator to train and test our model on Chengdu and Xi'an taxi datasets, the results showed that the mean absolute percent error (MAPE) of our model reaches 12.58% and 14.01%, which is better than the baselines of 1.54% and 1.78%. At the same time, we tested the effect of AGNII in Chengdu dataset, and the results showed that MAPE was reduced by 0.58% by employing AGNII.

The rest of this letter is organized as follows. In Section II, we make a formal description of the problem to be solved. In Section III, we elaborate on the structure and function of our algorithms. In Section IV, we train and test our model with Chengdu and Xi'an datasets and compare it with other baselines. Section V is the conclusion of this letter.

II. Problem Description

In this section, we describe the problem to be solved in this letter. Briefly, given a travel trajectory T_q, and we can get the start time t_q, the place of departure o_q, the destination d_q and the path sequence P_q from the trajectory. Our goal is to estimate the travel time y_p from o_q to d_q via P_q by comprehensively considering the dynamic characteristics (traffic flow, speed, etc) and the static characteristics (Road width, length, etc.) of the road network.

In particular, we use an undirected graph $G = (V, E)$ to represent the road network. Each road in the network is treated as a node and $V = \{v_1, v_2, \ldots, v_N\}$ is a set of nodes, where N is the number of roads. E is a set of edge and an edge $e_{ij} = (v_i, v_j) \in E$ indicates the vertex $v_i \in V$ links to the vertex $v_j \in V$. The adjacency matrix $A \in R^{N\times N}$ represents the connection relationship of the nodes, which is composed of 0 and 1. The trajectory T mentioned above is a sequence of consecutive GPS points, i.e. $T_1 = \{p_1, p_2, \ldots, p_n\}_1$. Each GPS point p_i contains three dimensions of information: the latitude, the longitude and the timestamp. By mapping the GPS points to the road network, we can get the sequence of paths covered by the trajectory, which is a series of nodes, and we call it the real paths in the following discussion. From the latitude and longitude information, we can get the distance of the trajectory, and from the timestamp, we can extract the start time and the travel time label information.

We minimize the loss function as follows:

$$\mathcal{L}(y_q; T_q, G_q, \theta) = \frac{|y_q - y_p|}{y_q} + \lambda \left\|\theta\right\|, \tag{1}$$

where y_p is predictive value and y_q is the time label.

III. Proposed Algorithms for More Accurate Travel Time Estimation

In this section, we describe the structure of GraphTTE. As shown in Fig. 1, GraphTTE is mainly composed of three parts: Multi-layer Spatiotemporal Graph frame(MSG), Attention Graph Node Impact Index algorithm(AGNII), and graph representation learning. MSG consists of static network layer and dynamic network layers. The static network layer processes static data (basic attribute of the road) to obtain the static characteristics of the road. The dynamic network layers composed of GCN and GRU to obtain the traffic dynamic characteristics at different frequencies, such as hourly, daily, weekly. AGNII scores the impact of nodes and extract nodes with high score to expand the model receptive field.

A. Multi-Layer Spatiotemporal Graph Frame

We design a framework, which includes a one-layer of static network and a multi-layer dynamic network, to obtain the spatial and temporal characteristics of the traffic network.

1) Static Network Layer: Static characteristics, such as road width, road level and speed limit, are significant factors that can impact the travel time. Therefore, we build a separate network to deal with such characteristics, as shown in Fig. 1. In particular, we employ k hops iteration of the graph convolutional network(GCN) [18] to update the vector of each node by obtaining the characteristics of its neighbors in k hops, so as to consider the topology of the road network:

$$X^k = \sigma(\hat{A}X^{k-1}W^k), \tag{2}$$

where $X^k \in R^{N\times h}$ represents the graph nodes characteristics matrix after k hops of iteration, N is the number of nodes, h is the dimension of the characteristic vector, $\hat{A} = \tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}$, $\tilde{A} = A + I_N$ is a matrix with self-connection structure, $\tilde{D} = \sum_j \tilde{A}_{ij}$ is degree matrix, σ is the activation function. We get the static characteristic matrix $C_S \in R^{N\times h_s}$ after k hops iterations of GCN, and each row of the matrix $C_i \in R^{h_s}$ is a new static characteristic vector of a node in the graph.

2) Dynamic Network Layers: The traffic condition is influenced by multiple time frequencies, such as the time periods of the day (morning peak periods, evening peak periods), and the days of the week (weekdays, weekends). Therefore, We use multiple networks to process the time characteristics of different frequencies respectively. Meanwhile, we still consider the impact of road network topology on dynamic characteristics. So we use GCN and recurrent neural network to obtain spatial and temporal characteristics in each layer of dynamic network.

Spatial characteristics: The impact of traffic congestion will spread according to the road connection. Therefore, the topology structure of the road network is important when we study dynamic traffic characteristics. We employ k hops iteration GCN to obtain the dynamic characteristics of the neighbors in k hops. So the new characteristic vector of each node contains the characteristics of its neighbors.

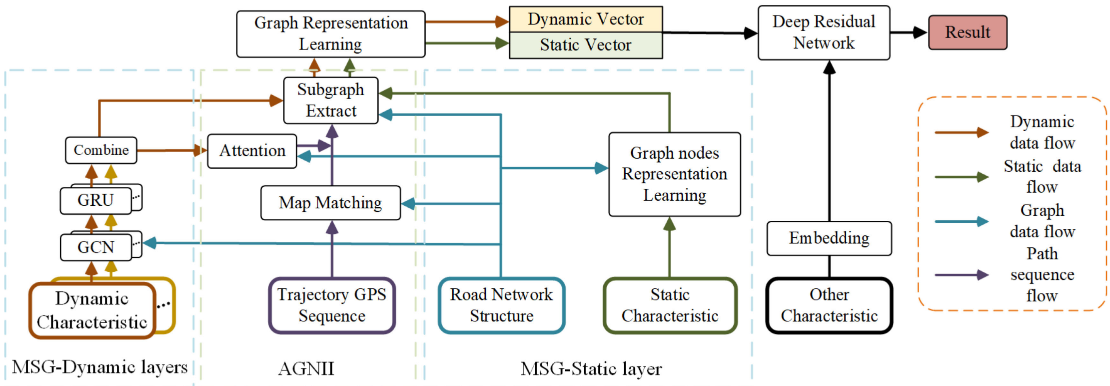


Fig. 1. GraphTTE model structure diagram. Dynamic characteristic refers to traffic speed and traffic flow, different colors represent different time frequencies. Static characteristic refers to lane number and speed limit, etc. Other characteristic refers to start time and travel distance, etc. Road network structure is represented by a graph.

Temporal characteristics: We employ Gated Recurrent Unit(GRU) [19] to process the temporal characteristics of traffic conditions at different time frequencies. For example, when considering the time periods in a day, we input the traffic dynamic characteristic data $X_R = [X^{t_q-M+1}, X^{t_q-M+2}, \ldots, X^{t_q}] \in R^{N\times M}$, where X is the dynamic characteristic data after k hops of GCN, t_q is the current period, M is the number of past time periods. It outputs a dynamic characteristic matrix $C_p \in R^{N\times h_d}$. For the daily time characteristics, we input the data $X_D = [X^{t_q-K\times T_D+1}, X^{t_q-(K-1)\times T_D+1}, \ldots, X^{t_q-T_D+1}] \in R^{N\times K}$, where T_D is the number of time periods in a day, K is the number of days. It outputs another dynamic characteristic matrix $C_d \in R^{N\times h_d}$. Then we combine the two results to form a traffic dynamic characteristic matrix $C_D \in R^{N\times (h_d+h_d)}$ and each row of the matrix $C_i \in R^{h_d+h_d}$ is a new dynamic characteristic vector of a node in the graph.

B. Attention Graph Node Impact Index Algorithm

A graph has a large number of nodes, but not every node has a significant impact on a particular trajectory. Therefore, in order to improve the effectiveness of the model, we need to extract the nodes that have a greater impact on the trajectory to constitute a subgraph. So we proposed an Attention Graph Node Impact Index algorithm (AGNII) to score the impact of each node on the trajectory as shown in Fig. 1.

First, we calculate the number of connections between each node within k hops [20], $L = \sum_{i=1}^{k} A^i$, where A is the adjacency matrix. Each element l_{ij} in the matrix L represents the number of ways to go from node i to node j in k hops, the greater the value of l_{ij}, the better the connectivity from node i to node j. Then we calculate the sum of the connectivity of each node in the graph to the real paths of the trajectory, $D_i = \sum_{j\in V_t} l_{ij}$, where i represents the i-th node in the graph, V_t represents the real paths. Finally, D_i combines with the dynamic vector or static vector of the i-th node to calculate the impact score s by the graph convolution network:

$$s = \sigma(\hat{A}\hat{C}W), \tag{3}$$

$$\hat{C} = C \parallel D \in R^{N\times (h+1)}, \tag{4}$$

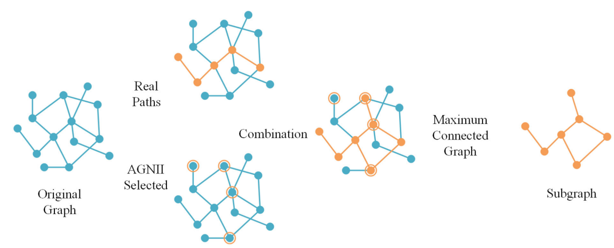


Fig. 2. Process of subgraph extraction.

where $\parallel$ represents the combination of vectors, $C \in R^{N\times h}$ represents the dynamic or static characteristic matrix, $D \in R^N$ represents the connectivity vector.

According to the pick ratio $r \in (0, 1]$, we keep $r \times N$ nodes with the highest scores [21] by $C_{extract} = top-rank(s, [rN])$, which can select the $r \times N$ nodes with the highest score. Then we extract the subgraph by combining the real paths and $C_{extract}$ as shown in Fig. 2.

C. Graph Representation Learning

We adopt the method of multi-node aggregation for graph representation learning [22]. In particular, we use two GCN to complete the characteristic extraction and node aggregation. The first GCN is used to obtain the node characteristic and graph structural characteristic:

$$Z_l = GCN_{embed}(A_l, X_l) = \sigma(\hat{A}_l X_l W_l) \in R^{N_l\times h_{l+1}}, \tag{5}$$

where X_l and A_l represents the characteristic matrix and the adjacency matrix of the l-th layer, and h_{l+1} represents the dimension of the characteristic vectors of each node of the (l+1)-th layer. The second GCN is used to calculate a weight matrix S, which is used to determine the way of aggregation:

$$S_l = GCN_{aggre}(A_l, X_l) = \sigma(\hat{A}_l X_l W_l) \in R^{N_l\times N_{l+1}}, \tag{6}$$

where N_{l+1} represents the number of nodes at the next level, then we can get the characteristic matrix and the adjacency matrix for the next layer:

$$X_{l+1} = S_l^T Z_l \in R^{N_{l+1}\times h_{l+1}}, \tag{7}$$

$$A_{l+1} = S_l^T A_l S_l \in R^{N_{l+1}\times N_{l+1}}, \tag{8}$$

TABLE I
COMPARISON RESULTS OF SIX MODELS IN CHENGDU/XI'AN DATASET

Model	RMSE(sec)	MAE(sec)	MAPE(%)
Average	401.33 / 337.30	322.91 / 252.22	22.96 / 26.64
GBDT	344.97 ± 3.29 / 291.07 ± 3.76	255.13 ± 2.87 / 207.79 ± 3.05	19.97 ± 0.06 / 22.31 ± 0.08
DeepTTE	283.17 ± 1.83 / 261.72 ± 1.94	232.97 ± 1.56 / 170.91 ± 1.61	15.69 ± 0.05 / 18.07 ± 0.05
$GTTE-S$	264.50 ± 1.72 / 244.57 ± 1.77	194.54 ± 1.58 / 157.39 ± 1.60	14.51 ± 0.06 / 16.12 ± 0.06
DeepOD	251.26 ± 1.55 / 237.91 ± 1.62	181.39 ± 1.43 / 150.24 ± 1.55	14.12 ± 0.05 / 15.79 ± 0.04
$GTTE-D$	247.01 ± 1.43 / 219.53 ± 1.69	176.51 ± 1.41 / 143.09 ± 1.53	13.14 ± 0.05 / 15.26 ± 0.05
GraphTTE	**242.62 ± 1.34 / 201.82 ± 1.60**	**172.64 ± 1.07 / 135.14 ± 1.49**	**12.58 ± 0.04 / 14.01 ± 0.05**

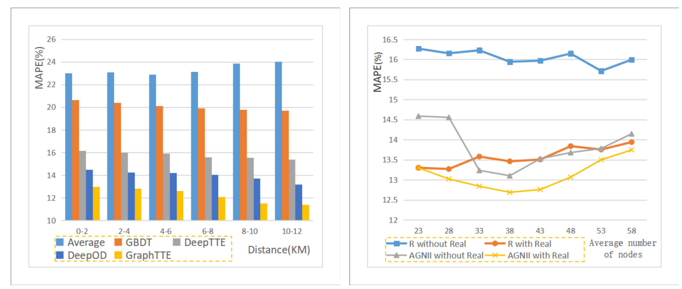
After several times of aggregation, the number of nodes is aggregated into one, we can finally obtain the dynamic vector V_d and static vector V_s. Then these two vectors and the embedded vectors of other attributes, such as distance, are input into the deep residual network to obtain the travel time estimation value, as shown in Fig. 1.

IV. EXPERIMENT

We evaluate our model on Chengdu and Xi'an taxi datasets from Didi Chuxing. The number of trajectories in both datasets is 50 000, time intervals are 10-20/10, 2018 and 01-20/11, 2016, the average numbers of road segments are 23 and 19, average travel times are 1247.96 s and 965.71 s, average lengths are 5010.03 m and 3622.57 m. The number of roads in two road natworks are 1891 and 1375. In the experiment, static data contains the number of lanes, road width and connectivity of road, dynamic data is traffic flow. When two-layer dynamic network is emplied to process traffic flow data of $10minutes$ frequency and $daily$ frequency and the selection rate of ANGII is 0.0095, we compare the mean absolute percent error(MAPE), root mean square error(RMSE) and mean absolute error(MAE) of GraphTTE with six baselines: **Average:** It calculates the travel time according to the average speed of road network in each time period. **GBDT [23]:** It obtains the travel time by adding the results of multiple decision trees. **DeepTTE [11]:** It uses CNN and LSTM to get characteristic of GPS points and calculate travel time. **DeepOD [17]:** It calculates travel time based on road sections embedding and trajectory spatiotemporal representation. **GTTE-D:** It only uses two layers of dynamic network of GraphTTE. **GTTE-S:** It only uses the static network of GraphTTE.

Table I shows that the MAPE of GraphTTE is at least 1.54% lower than that of baselines in Chengdu dataset and 1.78% lower in Xi'an dataset. Even if only dynamic networks is used, the results are 0.98% and 0.53% better than the baselines. Figure 3(a) shows MAPE of GraphTTE, DeeoOD, DeepTTE, GBDT and Average at different distances, GraphTTE is on average better than baselines 1.72%, 3.52%, 7.91% and 10.99%.

We also design ablation experiments to test the effect of AGNII. We keep the real paths and select [0,35] nodes by AGNII or random method to expand the model receptive field. The average number of nodes in real paths in the dataset is 23, so the range of the average number of nodes is [23,58]. Besides, we remove the real paths and only select nodes by AGNII or random method to determine the subgraph and also set the number of nodes to a range of [23,58]. As shown in Fig. 3(b), when the average number of nodes is 23 and 38, that is, the number of nodes AGNII selects is 0 and 15, the MAPE of AGNII with real paths is 13.2986% and 12.6871%, which indicates that the



(a) Comparison of five models in the range of 0-12 km.

(b) Ablation experiments associated with AGNII.

Fig. 3. Other experiments related to GraphTTE.

selection of additional nodes through AGNII reduces MAPE by 0.6115%. The MAPE of random method with real paths is higher than that of AGNII with real paths, and when the number of nodes selected is 15, their difference reaches a maximum of 0.7727%. In the absence of a real paths, we can found that MAPE decreased to 13.10% when 15 nodes are selected by AGNII. The MAPE of the random method fluctuates at 16%. Therefore, AGNII can have a positive effect on reducing MAPE by adding additional nodes, regardless of whether there are real paths.

We measure the running time of DeepTTE and GraphTTE on the CPU (Core i7-10th). DeepTTE takes 16 454.7 s in training, while GraphTTE takes 3554.6 s, which is 78.40% faster than DeepTTE. Besides, DeepTTE takes 273.5 s in testing, while GraphTTE takes 254.5 s, which is still 6.95% faster. In total time, DeepTTE takes 16 728.2 s, and GraphTTE takes 3809.1 s, which shows GraphTTE is 77.23% faster than DeepTTE. So our proposed GraphTTE has an advantage both in training and testing speed.

V. CONCLUSION

In this letter, we proposed a travel time estimation model GraphTTE to improve the accuracy of travel time estimation. To overcome the drawbacks in the existing works, we designed a Multi-layer Spatiotemporal Graph frame (MSG) to obtain the temporal and topological characteristics of traffic conditions. Besides, we designed an Attention Graph Nodes Impact Index algorithm (AGNII) to score the impact of each node on travel time, which can enlarge the sensitivity field of the model. We built a simulator and used Chengdu and Xi'an datasets to train and test our model. The results show that the accuracy of our model is superior to existing methods at least 1.54% and 1.78%, and the speed is 77.23% faster than baseline.

REFERENCES

[1] J. Sun, Z. Wu, Z. Yin, and Z. Yang, "SVM-CNN-based fusion algorithm for vehicle navigation considering atypical observations," *IEEE Signal Process. Lett.*, vol. 26, no. 2, pp. 212–216, Feb. 2019.

[2] W. Zhou, S. Lv, Q. Jiang, and L. Yu, "Deep road scene understanding," *IEEE Signal Process. Lett.*, vol. 26, no. 4, pp. 587–591, Apr. 2019.

[3] A. Prokhorchuk, V. P. Payyada, J. Dauwels, and P. Jaillet, "Estimating travel time distributions using copula graphical lasso," in *Proc. IEEE 20th Int. Conf. Intell. Transp. Syst. (ITSC).*, 2017, pp. 1–6.

[4] Y. Li, K. Fu, Z. Wang, C. Shahabi, J. Ye, and Y. Liu, "Multi-task representation learning for travel time estimation," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2018, pp. 1695–1704.

[5] R. Gao *et al.*, "Aggressive driving saves more time? Multi-task learning for customized travel time estimation." in *Proc. Twenty-Eighth Int. Joint Conf. Artif. Intell. (IJCAI-19)*, 2019, pp. 1689–1696.

[6] M. Asghari, T. Emrich, U. Demiryurek, and C. Shahabi, "Probabilistic estimation of link travel times in dynamic road networks," in *Proc. 23rd SIGSPATIAL Int. Conf. Adv. Geograph. Inf. Syst.*, 2015, pp. 1–10.

[7] X. Li, G. Cong, A. Sun, and Y. Cheng, "Learning travel time distributions with deep generative model," in *Proc. World Wide Web Conf.*, 2019, pp. 1017–1027.

[8] E. Badrestani *et al.*, "Real-time travel time estimation using matrix factorization," 2019, *arXiv:1912.00455*.

[9] A. Prokhorchuk, J. Dauwels, and P. Jaillet, "Estimating travel time distributions by Bayesian network inference," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 5, pp. 1867–1876, May 2020.

[10] Z. Meng, C. Wang, L. Peng, A. Teng, and T. Z. Qiu, "Link travel time and delay estimation using transit AVL data," in *Proc. 4th Int. Conf. Transp. Inf. Saf.*, 2017, pp. 67–72.

[11] D. Wang, J. Zhang, W. Cao, J. Li, and Y. Zheng, "When will you arrive? estimating travel time based on deep neural networks," in *Proc. AAAI Conf. Artif. Intell.*, vol. 32, no. 1, 2018, pp. 1–8.

[12] H. Wang, X. Tang, Y.-H. Kuo, D. Kifer, and Z. Li, "A simple baseline for travel time estimation using large-scale trip data," *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 2, pp. 1–22, 2019.

[13] Z. Wang, K. Fu, and J. Ye, "Learning to estimate the travel time," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2018, pp. 858–866.

[14] T.-Y. Fu and W.-C. Lee, "DEEPIST: Deep image-based spatio-temporal network for travel time estimation," in *Proc. 28th ACM Int. Conf. Inf. Knowl. Manage.*, 2019, pp. 69–78.

[15] X. Fang, J. Huang, F. Wang, L. Zeng, H. Liang, and H. Wang, "Constgat: Contextual spatial-temporal graph attention network for travel time estimation at baidu maps," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2020, pp. 2697–2705.

[16] S. Xu, R. Zhang, W. Cheng, and J. Xu, "MTLM: A multi-task learning model for travel time estimation," *GeoInformatica*, pp. 1–17, 2020.

[17] H. Yuan, G. Li, Z. Bao, and L. Feng, "Effective travel time estimation: When historical trajectories over road networks matter," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2020, pp. 2135–2149.

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

[19] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," 2014, *arXiv:1409.1259*.

[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 Trans. Intell. Transp. Syst.*, vol. 21, no. 11, pp. 4883–4894, Nov. 2020.

[21] H. Gao and S. Ji, "Graph U-Nets," 2019, *arXiv:1905.05178*.

[22] Z. Ying, J. You, C. Morris, X. Ren, W. Hamilton, and J. Leskovec, "Hierarchical graph representation learning with differentiable pooling," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 4800–4810.

[23] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Ann. Statist.*, vol. 29, no. 5, pp. 1189–1232, 2001.