# Learning Latent Road Correlations from Trajectories

Zheng Dong[1], Quanjun Chen[2]†, Renhe Jiang[2,3]†, Huanchen Wang[1], Xuan Song[1,2], Hao Tian[4]
[1]Department of Computer Science and Engineering, Southern University of Science and Technology, China
[2]Center for Spatial Information Science, The University of Tokyo, Japan
[3]Information Technology Center, The University of Tokyo, Japan
[4]Transport Bureau of Shenzhen Municipality, China
Email: zhengdong00@outlook.com; chen1990@iis.u-tokyo.ac.jp; jiangrh@csis.u-tokyo.ac.jp;
wanghc1999@gmail.com; songx@sustech.edu.cn; 1846789534@qq.com

2022 IEEE International Conference on Big Data (Big Data) | 978-1-6654-8045-1/22/$31.00 ©2022 IEEE | DOI: 10.1109/BigData55660.2022.10020759

***Abstract*—A core component of the Intelligent Transportation System (ITS) is road network, which forms the most basic transport infrastructure, and becomes widely applied in many traffic applications. In most traffic models, the spatial representation of road network is learned only through static graph connection while dynamic driver preference and traffic conditions in the real world are ignored. Therefore, in this paper, a novel trajectory-based road network representation is proposed. By mining vehicle trajectories, our proposed method can learn dynamic route choice through embeddings of each road in a next-hop prediction model. Then road correlations are calculated by the embeddings to build a latent correlation graph that can be applied in various traffic-related applications. Extensive experiment results prove the effectiveness and rationality of our proposed approach.**

***Index Terms*—Trajectory, Road Network, Road Correlation, Representation Learning**

## I. Introduction

With the great development of modern cities, the rapid growth of population and the acceleration of urbanization has made transportation systems essential infrastructure. In the meantime, transportation systems are becoming more and more complex, which causes great pressure on urban traffic management. As a result, it is important to develop the Intelligent Transportation System (ITS) [1] for efficient traffic management. A core component of ITS is the road network, which forms the most basic transport infrastructure, and becomes widely applied in many traffic applications.

Considering its importance, lots of methods are developed to characterize and model road networks. Early research mainly adopt graph structure on roads. To represent a road network, a graph is constructed where each node in the graph stands for a road segment, and edges represent the relationship between them. In most traffic models, the edges are pre-defined, such as road connectivity. It cannot capture the long-range dependency among roads and only reflects the structural characteristics. Recent studies start to leverage graph representation learning techniques to obtain representations over road networks. Representation learning methods have the ability to extract the underlying characteristics of road networks. However, considering dynamic driver preference and traffic conditions in the real world, it is not easy to design an effective representation learning framework. The traffic condition in the real world is much more complicated rather than simple road connections. For example, the main roads in a city are often congested during peak hours. Although it is usually the shortest path to travel through main roads, commuters will probably prefer a father but clearer path. That is, simply considering the structural information is not enough to characterize the transfer preference by real drivers and pedestrians. Despite that it is impractical to collect all the traffic patterns, the vehicle trajectories reflect them well and thoroughly.


Fig. 1: Learn road correlations from trajectories.

Trajectories will provide vehicle transition information, and

† Corresponding Author.

the relationship between roads can be better captured by learning real-world road correlations from huge vehicle paths. Therefore, we take advantage of trajectories and design a road network representation learning framework. In this paper, our main purpose is to design a general method to extract road correlations through trajectories. As shown in Figure 1, based on the learned correlations, a latent road correlation graph $\mathcal{G}_C$ is proposed to represent the real road relationship. It can be used in various traffic-related applications.

The contributions of our paper are:

- We propose a procedure to learn road correlation through trajectories and generate a latent graph representation for various downstream traffic-based application tasks.
- We evaluate our proposed latent graph on traffic flow prediction as a downstream task, and prove it can improve prediction accuracy through experiments on a real-world traffic dataset.

The rest of this paper is organized as the following. Section II introduces related work. Section III provides the notations and definitions. The adopted methodology and model structure will be given in section IV. In section V, the proposed model is verified by our dataset, and the experiment results demonstrate its practicability. A brief conclusion and several potential directions are provided in the last section VI.

## II. Related Work

**Graph Representation Learning.** The problem of how to make a numerical representation of a graph has long been a hot research area in graph theories. Instead of an adjacency matrix, to represent the non-euclidean information, many approaches have been proposed. Classical graph embedding methods are targeted at reducing the dimension of high-dimensional graph data into a lower-dimensional representation while preserving the desired properties of the original data. For example, Principal Component Analysis (PCA) [2], Linear Discriminant Analysis (LDA) [3], and Multidimensional Scaling (MDS) [4]. Later, statistical models are used to represent the road network, such as Hidden Markov Models [5] that are used to model the location transitions over the road networks. The random-walk-based graph embedding algorithms including DeepWalk [6] and node2vec [7] utilize random walks to learn node representations. They sample many random paths over the graph. These paths indicate the context of connected nodes. The randomness of walks gives the ability to explore the graph and capture both the global and the local structural information by walking through neighbors. Then, probability models like skip-gram [8] can be applied to learn node representations. The growth of deep learning models makes it possible to model more complex road networks and learn graph representation efficiently, including GCN [9], GraphSAGE [10], GAT [11] and other types of networks. They perform message passing from neighboring nodes and apply various aggregation strategies. Based on the core architecture, hierarchical models such as HRNR [12] are proposed to capture both structural and functional characteristics, resulting in an effective graph representation.

**Road Network Modeling.** Road network is the basic component of the urban traffic system. There are a variety of applications that develop on it, such as travel time estimation [13], travel route recommendation [14], and destination prediction [15]. In traffic prediction tasks, the early methods, i.e. statistical algorithms, machine learning techniques, and RNNs do not take the road network into consideration. Instead, these models treat traffic sequences from different roads as independent data streams, being unable to utilize the spatial information in the road network graph. Therefore, the GCN-based models utilize the adjacency matrix of the road network to represent spatial relations. However, the adjacency matrix cannot always precisely reflect the real-world dependencies of the road network. Therefore, other types of matrices are proposed. The (1) distance-based matrix is widely used whose values are the geometrical distance between nodes. The (2) similarity-based matrix represents the correlations between traffic states, e.g. similarities of traffic flow patterns. And there is also (3) dynamic matrix constructed along the training process of traffic prediction models. For example, Graph WaveNet [16] learns an adaptive adjacency matrix through the end-to-end stochastic gradient descent. Furthermore, for the reason that single graph representation may be insufficient to model complex spatial relations, multi-graph models [17]–[20] have also been proposed by some researchers recently.

However, few existing works consider real-world traffic in road network modeling. On the contrary, our work attempts to learn a latent road network representation from trajectories, in order to capture the road correlations under real-world traffic conditions. The latent representation is designed for general instead of a specific traffic-related task, which can be used in various downstream applications. To the best of our knowledge, it is the first time that a purely trajectory-based graph representation learning model has been proposed for road networks.

## III. Preliminaries

In this section, we will introduce the notations used in this paper, as well as definitions and problems in our task.

### A. Notations

Table I gives the frequently used notations and their definitions.

TABLE I: Frequently used notations.

| Notation | Definition |
|---|---|
| $n_{(\cdot)}$ | number of some entities |
| $d_{(\cdot)}$ | vector dimension |
| $r$ | a single road in the road network |
| $A$ | adjacency matrix |
| $T$ | trajectory |
| $w$ | window size |
| $E$ | road embedding matrix |
| $\mathbf{e}_{r_i}$ | embedding vector for road $r_i$ |
| $C$ | road correlation matrix |
| $t$ | time interval (or time step) |
| $X$ | traffic flow matrix |

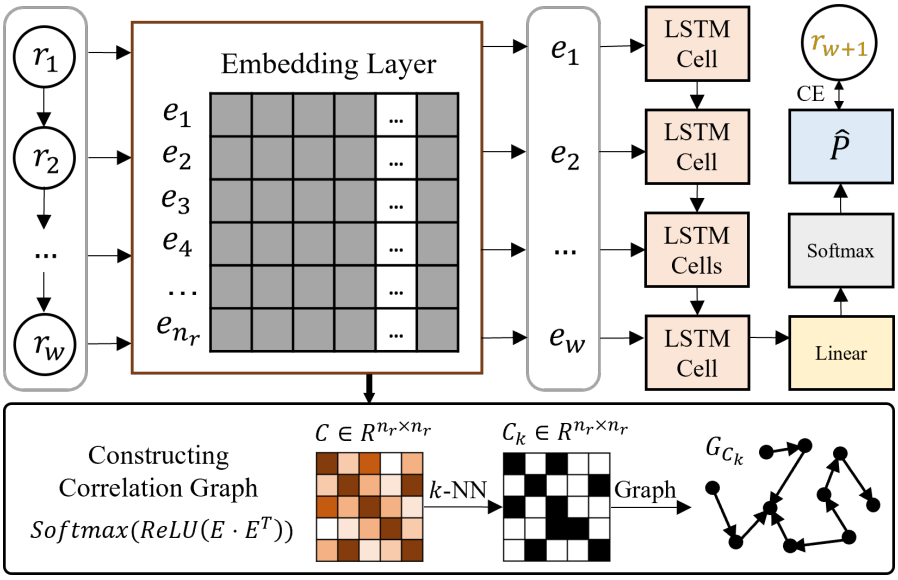



Fig. 2: Model architecture for road correlation learning.

### B. Problem Definition

This subsection gives the definitions of the concepts and tasks in this paper.

*Definition 1 (Road Network Graph):* The road network can be represented by a directed graph $\mathcal{G} = (\mathcal{R}, \mathcal{E}, A)$, where $\mathcal{R} = \{r_1, r_2, \ldots, r_{n_r}\}$ is a finite set of roads that each $r_i$ stands for a real road in the road network, which is an integer ID in practice. $\mathcal{E}$ is the set of directed edges where $(r_i, r_j) \in \mathcal{E}$ indicates that there is a directed edge from $r_i$ to $r_j$, i.e. $r_j$ is the downstream road in the road network. $A \in [0,1]^{n_r \times n_r}$ is the adjacency matrix whose entry $A_{i,j}$ is a binary value that indicates whether there exists an edge $(r_i, r_j) \in \mathcal{E}$.

*Definition 2 (Trajectory):* Given a road network graph $\mathcal{G} = (\mathcal{R}, \mathcal{E}, A)$, a trajectory $T = [(r_1, ts_1), (r_2, ts_2), \ldots, (r_l, ts_l)]$ is a sequence of (road, timestamp) tuples. Each tuple $(r_i, ts_i)$ specifies that the vehicle is driving on road $r_i$ at timestamp $ts_i$. Each consecutive pair of tuples are connected in the road network graph, i.e. $\forall i = 1, 2, \ldots, l-1$, $r_i \neq r_{i+1}$ and $(r_i, r_{i+1}) \in \mathcal{E}$.

*Definition 3 (Traffic Flow):* Traffic flow is defined as the number of vehicles passing by the road during a time interval. For a road graph $\mathcal{G} = (\mathcal{R}, \mathcal{E}, A)$, we use the traffic flow matrix $X \in \mathbb{R}^{n_r \times n_t}$ to record the traffic flow of each time interval. For time interval $t$, $\mathbf{x}_t = X_{:,t} \in \mathbb{R}^{n_r}$ represents the traffic flow of all roads during $t$.

*Problem 1 (Next-hop Prediction):* Given a road network graph $\mathcal{G} = (\mathcal{R}, \mathcal{E}, A)$ and the road sequence in a trajectory $T^r = [r_1, r_2, \ldots, r_l]$ with length $l$, the trajectory next-hop prediction is to obtain a probability distribution $\hat{P}$ on road set $\mathcal{R}$ that best predicts the next step $r_{l+1}$. In conclusion, our goal is to build a model with parameters $\Theta^*$ that satisfies

$$\Theta^* = \underset{\Theta}{\operatorname{argmin}}\, CrossEntropy(\hat{P}, P) \tag{1}$$

*Problem 2 (Road Correlation):* Given a road network graph $\mathcal{G} = (\mathcal{R}, \mathcal{E}, A)$ and a trajectory set $\mathcal{T}$, find a road correlation function $f_C$ with respect to $\mathcal{T}$ which takes two roads as input and returns a real number $f_C(r_i, r_j)$ to quantify the spatial correlation between two roads $r_i$ and $r_j$. The value is bigger if the two roads have a stronger correlation. The road correlation matrix $C \in \mathbb{R}^{n_r \times n_r}$ stores all the correlation values s.t. $C_{i,j} = f_C(r_i, r_j)$.

## IV. Methodology

### A. Learning Road Correlations

To deal with trajectories, a simple way is to consider them as first-order Markov processes [21], and calculate the Markov transition probability as road correlation value by iterating on all trajectories. But the Markov process is short in modeling high-order transition, since the next step is only related to the previous one, by its definition. Therefore, instead of statistical methods that result in a fixed probability value, we take advantage of deep learning to let the machine automatically learn the transition process and acquire the high-order dependencies. To be specific, our idea is to build a trajectory next-hop prediction model and dynamically learn a vector representation of each road. Then compute vector similarity as road correlation.

In the NLP area, how to obtain effective representations of text words has long been a research focus. One-hot embedding is a simple solution to represent each word with a one-hot vector whose dimension is equal to the size of the vocabulary. The difference between these vectors is the word index. However, one-hot embedding suffers from dimension curse, making the embedding vectors very large and sparse. More importantly,

the vectors cannot reflect the semantic relationships between words. Therefore, word embedding technologies are proposed to efficiently learn a fixed-length real-value vector representation for each word. And the most important advantage is that it can catch the contextual similarity of words. If two embedding vectors are close, which means they have high similarity, then the two corresponding words also tend to have similar semantic meanings.

In our work, a trajectory contains a sequence of different road IDs, which is similar to a sentence. Therefore, it is natural to treat each road as a word and use an embedding vector to represent it [22]–[24]. What's more, the road correlation can be modeled by the contextual similarity of embedding vectors, which can reflect the low and high-order dependency among roads. As a result, for road $r_i$, $r_j$ and their embedding vectors $\mathbf{e_{r_i}}$ and $\mathbf{e_{r_j}}$, we define the road correlation as the dot-product similarity of the embeddings, i.e.

$$f_C(r_i, r_j) = \mathbf{e_{r_i}} \cdot \mathbf{e_{r_j}} \tag{2}$$

To learn the embedding vectors, we propose a simple LSTM-based model to predict the next hop of a trajectory. The model structure is given in Figure 2. Firstly, to utilize the trajectories, we use a sliding window strategy to generate training samples. For trajectory $T^r = [r_1, r_2, \ldots, r_l]$, we can obtain $l - w$ fragments by sliding a window with size $w$. A fragment is denoted as $[r_1, r_2, \ldots, r_w]$, as well as its corresponding next hop $r_{w+1}$. Next, we feed the fragments into an embedding layer that uses an embedding matrix $E \in \mathbb{R}^{n_r \times d_r}$ to map each road ID to a vector of continuous values, where $n_r$ is the number of roads and $d_r$ is the embedding dimension. The second layer is an LSTM to capture dependencies in the sequence. LSTM encodes the embedded fragment into a single hidden vector with dimension $d_h$ after $w$ steps' iteration. The last layer is a fully connected linear layer that converts the hidden vector to an output length-$n_r$ vector for classification. Then use *Softmax* function to generate the probability distribution. Cross entropy is served as the loss function. The procedure of forward propagation is given in the following equations.

$$\begin{aligned} \mathbf{e} &= \text{Embedding}(r_1, r_2, ..., r_w) \\ h &= \text{LSTM}(\mathbf{e_{r_1}}, \mathbf{e_{r_2}}, ..., \mathbf{e_{r_w}}) \\ o &= W^\top h + b \\ \hat{P} &= Softmax(o) \\ \mathcal{L} &= CrossEntropy(\hat{P}, P) \end{aligned} \tag{3}$$

where $W$ and $b$ are parameters of the fully connected linear layer. After training, we take out the embedding matrix $E$ in the embedding layer and calculate the road correlation matrix $C$:

$$C = Softmax(ReLU(E \cdot E^\top)) \in \mathbb{R}^{n_r \times n_r} \tag{4}$$

where $C_{r_i, r_j} = \mathbf{e_{r_i}} \cdot \mathbf{e_{r_j}}$. Its corresponding graph is the proposed latent road correlation graph $\mathcal{G}_C$, which is undirected.

Furthermore, we propose to refine the correlation matrix via the idea of $k$-nearest neighbors ($k$-NN) that builds the connections between highly correlated roads and removes the low correlations to guarantee the sparsity. The $k$-adjacency matrix $C_k \in [0, 1]^{n_r \times n_r}$ will be determined as follows:

$$C_{k_{r_i, r_j}} = \mathbb{I}(r_j \in k\text{-NN}(r_i)) \tag{5}$$

where $\mathbb{I}$ stands for the indicator function, $k$-NN generates a set of $k$ nearest neighbor roads. The corresponding graph to $C_k$ is denoted as $\mathcal{G}_{C_k}$.

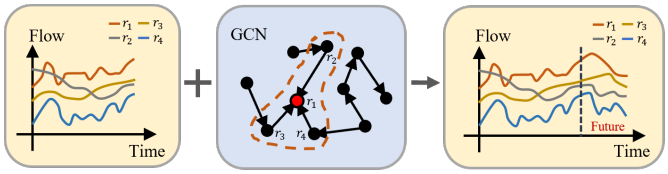


Fig. 3: Downstream task: traffic flow prediction.

### B. Downstream Task

To illustrate the effectiveness of our proposed $\mathcal{G}_C$ and $\mathcal{G}_{C_k}$, we take traffic flow prediction as a downstream task. Traffic flow prediction is the process of analyzing urban road traffic conditions, mining traffic patterns, and predicting road traffic flow trends. Graph Convolutional Networks (GCNs) have been widely applied in traffic flow prediction in recent years for their ability to efficiently capture spatiotemporal dependencies. There are two main types of GCN models: spectral-based and spatial-based. In spectral-based GCNs, the graph signal is first transformed to the spectral domain and then transformed back after convolution. The convolution [25] operation is defined as:

$$\mathbf{g} \star \mathbf{x} = \mathbf{U}\mathbf{g}\mathbf{U}^\top\mathbf{x} \tag{6}$$

where $\mathbf{g}$ is the filter, $\mathbf{x}$ is the graph signal and $\mathbf{U}$ is the matrix of eigenvectors of the normalized graph Laplacian. In practice, the filter is approximated by the Chebyshev polynomials of the diagonal matrix of eigenvalues. Another approach is spatial-based GCN, where the convolution is defined by information propagation. The information of a node comes from its neighboring nodes and is aggregated by different strategies. The method is modeled by diffusion, message passing, and attention, etc. As shown in Figure 3, the GCN prediction models take two inputs: (1) road network, which is a graph representation, and (2) traffic flow, which is the flow series of each road. The output is the predicted future traffic flow.

Therefore, we select four GCN-based traffic prediction models:

- **STGCN** [26] (Spatio-Temporal Graph Convolutional Networks). STGCN is proposed in 2018, which is one of the earliest spatiotemporal GCN models for traffic prediction. It uses gated TCN [27] to capture temporal dependencies and applies spectral graph convolution [9] on the graph to model the spatial dependencies.
- **DCRNN** [28] (Diffusion Convolutional Recurrent Neural Network). DCRNN is another typical GCN model that is also proposed in 2018. Instead of spectral graph convolution, it implements diffusion convolution through

bidirectional random walks to capture the transition information. Temporally, it integrates diffusion convolution into GRU and proposed an encoder-decoder structure to enable multistep prediction.

- **GWNET** [16] (Graph WaveNet). Graph WaveNet is a GCN model proposed in 2019, which remains one of the best SOTA models nowadays. It adopts an adaptive/learnable graph rather than a static one that is used in STGCN and DCRNN. By combining the original adjacency matrix and the learned graph, it can rapidly boost the prediction accuracy of traffic flow. But the adaptive graph functionality is disabled in the experiments for the comparison of different input graphs.
- **ASTGCN** [29] (Attention Based Spatial-Temporal Graph Convolutional Networks). ASTGCN is proposed in 2019. The model combines the spatial-temporal attention mechanism and the spatial-temporal convolution, including graph convolutions in the spatial dimension and standard convolutions in the temporal dimension, to simultaneously capture the dynamic spatial-temporal characteristics of traffic data.

We will use these models to compare the performance of our proposed latent graph with other road graphs.

TABLE II: Dataset description.

| | |
|---|---|
| **Region** | Shenzhen CBD |
| **Time Range** | June 2020 |
| **#GPS Points** | 25,828,330 |
| **#Trajectories** | 1,751,602 |
| **#Time Intervals** | 8064 |
| **#Roads** | 492 |
| **Length of Time Interval** | 5 minutes |

TABLE III: Model parameters.

| Notation | Parameter | Value |
|---|---|---|
| Latent Correlation Graph Learning Model | | |
| $w$ | Window size | 5 |
| $d_r$ | Road embedding dimension | 64 |
| $d_h$ | LSTM hidden size | 256 |
| $d_o$ | Linear output dimension | 492 |
| | Batch size | 256 |
| | Early stopping epochs | 10 |
| | Learning rate | 0.0001 |
| Traffic Flow Prediction Model | | |
| $w_{in}$ | Input steps | 12 |
| $w_{out}$ | Prediction steps | 12 |
| | Batch size | 32 |
| | Learning rate | 0.001 |
| | Early stopping epochs | 10 |
| | Max training epochs | 200 |

## V. Experiments

### A. Settings

Our dataset consists of taxi GPS points in Shenzhen, where the main attributes are latitude, longitude, and timestamp. Considering the size of the graph, we manually selected a rectangular area in the CBD of Futian district. The road network data is retrieved from OSM [30] and simplified. The trajectories were generated by performing Fast Map Matching (FMM) [31]. After that, the traffic flow for each road is aggregated per 5 min interval. The description of our dataset is given in Table II. Our dataset contains 1,751,602 trajectories in total. After removing too long and too short ones, there were 1,351,700 remaining. Then we put the trajectories into 24 bins according to their starting time and randomly sampled 80% data in each bin to combine as the whole dataset. Finally, there were 1,076,886 trajectories.

For the latent road correlation learning model, the window size for trajectory fragments generation was set as $w$ = 5. The data ratio for training, validation, and testing was set as 7:1:2. Finally, we got 10,228,578 trajectory fragments for training, and 1,455,100 for validation. For model parameters, $d_r$ = 64, $d_h$ = 256, batch size was 256, and learning rate was 0.0001. Adam [32] algorithm was employed to control the overall training process, and the loss function was *Cross Entropy Loss*. We performed a thorough search on vector dimensions, and the complete parameter selection is given in Table III.

As mentioned above, we use traffic flow prediction as a downstream task for the experiments. For traffic flow prediction models, the choices of $k$ for the generation of $\mathcal{G}_{C_k}$ for the four GCN prediction models were 25, 8, 6, and 15, respectively. The experiments were performed on DL-Traff [33], an open-source benchmark platform. As for the parameters, the input steps and prediction steps were both set to 12. The data split ratio was also 7:1:2. The optimizer was Adam where the learning rate was set as 0.001, and the loss function was *Mean Squared Error (MSE) Loss*. The batch size was set to 32. The training process would be early-stopped if the validation loss was not decreasing for 10 epochs, then the best model on validation data would be saved. Details are provided in Table III.

Our experiments were performed on a server equipped with an Intel(R) Xeon(R) Silver 4216 CPU @ 2.10GHz and an NVIDIA GeForce RTX 2080Ti graphics card. The PyTorch version is 1.7.1 with Python 3.7.11.

### B. Baseline

To compare the effectiveness of the latent correlation graph with other graph representations, we selected several statistical graph representations that are frequently used in traffic prediction models. We evaluate them on the above GCN models and show the advantage of the proposed latent graph.

- $\mathcal{G}_A$. The 0-1 adjacency matrix of the graph.
- $\mathcal{G}_{OD}$. The Origin-destination graph. For a trajectory $T$, we set the starting road $r_s$ as the origin and the ending road $r_e$ as the destination. Then we iterate on all trajectories and increase the value of the corresponding $OD_{r_s,r_e}$ in the matrix representation of $\mathcal{G}_{OD}$.
- $\mathcal{G}_{COS}$. Cosine similarity graph. For each pair of roads, the cosine similarity of their traffic flow series can be used to measure their correlation. For road $r_i$ and $r_j$, the corresponding value in its representation $COS$ matrix is

TABLE IV: Performance evaluation results.

| Model | Graph Type | 3 Steps / 15 min | | | 6 Steps / 30 min | | | 12 Steps / 60 min | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | RMSE | MAE | MAPE | RMSE | MAE | MAPE | RMSE | MAE | MAPE |
| STGCN | $\mathcal{G}_A$ | 4.44 | 3.46 | 33.00% | 4.64 | 3.57 | 33.64% | 5.07 | 3.88 | 36.07% |
| | $\mathcal{G}_{OD}$ | 4.42 | 3.44 | 33.08% | 4.72 | 3.61 | 33.73% | 5.43 | 4.12 | 37.90% |
| | $\mathcal{G}_{COS}$ | 5.43 | 4.08 | 37.40% | 5.74 | 4.39 | 40.47% | 6.35 | 4.75 | 45.24% |
| | $\mathcal{G}_{DTW}$ | 4.40 | 3.41 | 32.48% | 4.63 | 3.57 | 33.70% | 5.03 | 3.82 | 35.73% |
| | $\mathcal{G}_C$ | **4.38** | **3.40** | **32.41%** | **4.56** | **3.52** | 33.33% | 5.03 | 3.82 | 35.87% |
| | $\mathcal{G}_{C_k}$ | 4.42 | 3.43 | 32.70% | **4.56** | **3.52** | **33.26%** | **5.02** | **3.81** | **35.71%** |
| DCRNN | $\mathcal{G}_A$ | 4.47 | 3.46 | 33.01% | 4.66 | 3.60 | 34.06% | 5.03 | 3.85 | 35.76% |
| | $\mathcal{G}_{OD}$ | 4.49 | 3.46 | 32.92% | 4.71 | 3.61 | 34.16% | 5.15 | 3.87 | 36.19% |
| | $\mathcal{G}_{COS}$ | 4.42 | 3.42 | 32.71% | **4.60** | **3.53** | **33.46%** | 5.01 | 3.79 | 35.35% |
| | $\mathcal{G}_{DTW}$ | 4.54 | 3.50 | 33.01% | 5.07 | 3.78 | 35.30% | 5.75 | 4.19 | 38.21% |
| | $\mathcal{G}_C$ | 4.51 | 3.84 | 32.92% | 4.71 | 3.63 | 34.09% | 5.14 | 3.91 | 36.25% |
| | $\mathcal{G}_{C_k}$ | **4.40** | **3.39** | **32.38%** | 4.63 | 3.54 | 33.60% | **4.98** | **3.77** | **35.26%** |
| GWNET | $\mathcal{G}_A$ | 4.45 | 3.45 | 32.66% | 4.69 | 3.61 | 33.97% | 5.21 | 3.94 | 36.46% |
| | $\mathcal{G}_{OD}$ | 4.51 | 3.48 | 32.76% | 4.78 | 3.65 | 34.04% | 5.45 | 4.07 | 36.81% |
| | $\mathcal{G}_{COS}$ | 4.48 | 3.47 | 33.12% | 4.86 | 3.72 | 34.94% | 5.48 | 4.12 | 37.78% |
| | $\mathcal{G}_{DTW}$ | 4.61 | 3.54 | 33.30% | 5.06 | 3.82 | 35.41% | 5.72 | 4.22 | 38.26% |
| | $\mathcal{G}_C$ | 4.48 | 3.45 | 32.82% | 4.68 | 3.58 | 33.79% | 5.13 | 3.85 | 35.71% |
| | $\mathcal{G}_{C_k}$ | **4.41** | **3.41** | **32.61%** | **4.57** | **3.51** | **33.31%** | **4.98** | **3.76** | **35.25%** |
| ASTGCN | $\mathcal{G}_A$ | 4.65 | 3.57 | 33.76% | 4.84 | 3.67 | 34.59% | 5.21 | 3.90 | 36.38% |
| | $\mathcal{G}_{OD}$ | 4.65 | **3.54** | 33.52% | 4.86 | 3.67 | 34.68% | 5.30 | 3.96 | 37.80% |
| | $\mathcal{G}_{COS}$ | 4.76 | 3.65 | 33.89% | 4.90 | 3.73 | 34.35% | 5.40 | 4.08 | 36.96% |
| | $\mathcal{G}_{DTW}$ | 6.30 | 4.29 | 38.54% | 6.61 | 4.47 | 39.77% | 7.01 | 4.85 | 42.77% |
| | $\mathcal{G}_C$ | 4.67 | 3.58 | 33.92% | 4.88 | 3.70 | 34.94% | 5.30 | 3.95 | 36.87% |
| | $\mathcal{G}_{C_k}$ | **4.61** | **3.54** | **33.46%** | **4.75** | **3.62** | **33.93%** | **5.11** | **3.83** | **35.72%** |

calculated by the angle cosine of the $r_i$-th and $r_j$-th row in the traffic flow matrix $X$, i.e.

$$COS_{r_i,r_j} = \frac{X_{r_i,:} \cdot X_{r_j,:}}{|X_{r_i,:}| \, |X_{r_j,:}|} \tag{7}$$

- $\mathcal{G}_{DTW}$. Dynamic time warping [34] graph. For each pair of roads, the DTW distance can measure the correlation of their traffic flow time series. For road $r_i$ and $r_j$, the corresponding value in the $DTW$ matrix is calculated by the DTW distance of the $r_i$-th and $r_j$-th row in the traffic flow matrix $X$, i.e.

$$DTW_{r_i,r_j} = DTWDistance\left(X_{r_i,:}, X_{r_j,:}\right) \tag{8}$$

### C. Evaluation Metrics

Following previous studies, we use *Root Mean Square Error (RMSE)*, *Mean Absolute Error (MAE)*, and *Mean Absolute Percentage Error (MAPE)* as the metrics to show the performance of different methods. Lower errors indicate better performance. The definitions of them are shown as the following.

$$\begin{aligned} MAE &= \frac{1}{n}\sum_{i=1}^{n}|\hat{y}_i - y_i| \\ MAPE &= \frac{1}{n}\sum_{i=1}^{n}\left|\frac{\hat{y}_i - y_i}{y_i}\right| \\ RMSE &= \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2} \end{aligned} \tag{9}$$

where $y_i$ is label, $\hat{y}_i$ is prediction and $n$ is the number of samples.

### D. Performance Analysis

The comparison for the performance of traffic flow prediction is given in Table IV. To demonstrate how the prediction accuracy varies with time, we report the metrics for 15, 30, and 60-minute ahead prediction. From the table, we obtain the following observations:

- Among all graphs, our proposed $\mathcal{G}_C$ and $\mathcal{G}_{C_k}$ achieve the best performance in all cases, especially a big improvement compared to the original adjacency graph $\mathcal{G}_A$. This illustrates that the latent road correlations can better capture the real-world spatial correlation of the road network.
- With the increasing of prediction steps, all models tend to perform worse because of the difficulty in long-range prediction. However, our proposed latent graphs still have a considerable improvement on long-range prediction. Especially for GWNET and ASTGCN, the improvement on 60-minute ahead prediction is larger than short-range prediction. This is because our latent correlation learning model uses a sequence learning method, which can capture the high-dimensional road dependencies.
- Comparing the four models, DCRNN and GWNET have the best overall performance. DCRNN is good at short-range prediction, where the metrics are much lower than other models. GWNET performs well in mid and long-range cases.
- Comparing $\mathcal{G}_C$ and $\mathcal{G}_{C_k}$, it can be found that $\mathcal{G}_{C_k}$ has a better performance in almost all cases. This proves the effectiveness of $k$-NN filtering. However, speaking to $\mathcal{G}_C$, it is still better than other baseline graphs. For example,

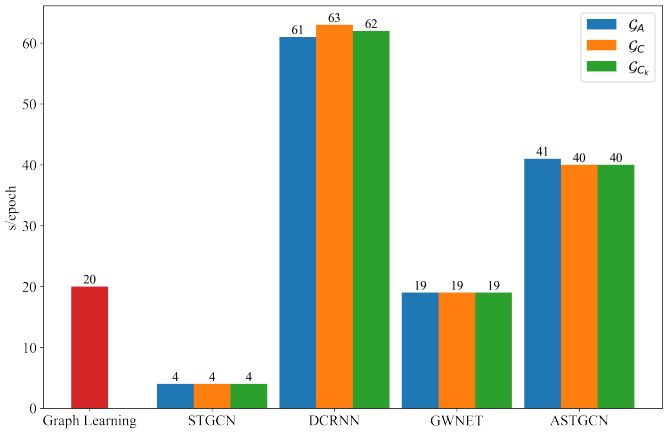



Fig. 4: Training time for each model.

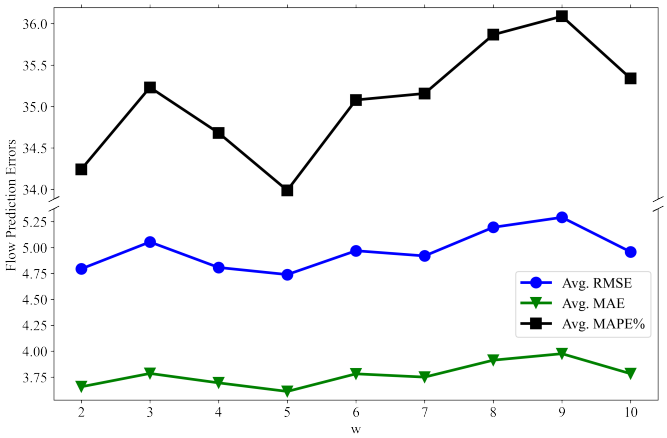



Fig. 5: GWNET-$\mathcal{G}_C$ average prediction errors w.r.t. $w$.

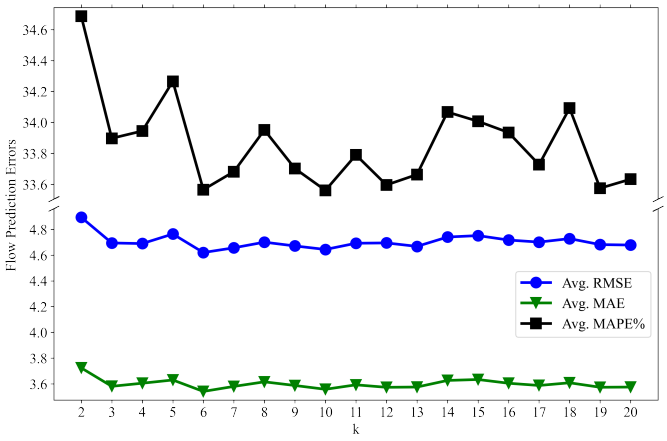



Fig. 6: GWNET-$\mathcal{G}_{C_k}$ average prediction errors w.r.t. $k$.

in STGCN and GWNET, $\mathcal{G}_C$ achieves the second lowest error compared to other graphs.

In Figure 4, we report the training time per epoch for the latent correlation graph learning model and the four GCN-based traffic prediction models. The latent graph learning model mainly consists of an LSTM layer, but it still trains fast due to its simple architecture and simple loss function. For the prediction models, we report the training time with respect to three different input graphs, which are $\mathcal{G}_A$, $\mathcal{G}_C$, and $\mathcal{G}_{C_k}$. We can observe that the training time is almost not affected by the input graphs at all. In addition, the computation time of $k$-NN filtering is too small that it can be ignored. To conclude, although our proposed latent graph $\mathcal{G}_C$ and $\mathcal{G}_{C_k}$ are more complex, they do not increase the computation cost for the downstream traffic prediction models.

### *E. Parameter Analysis*

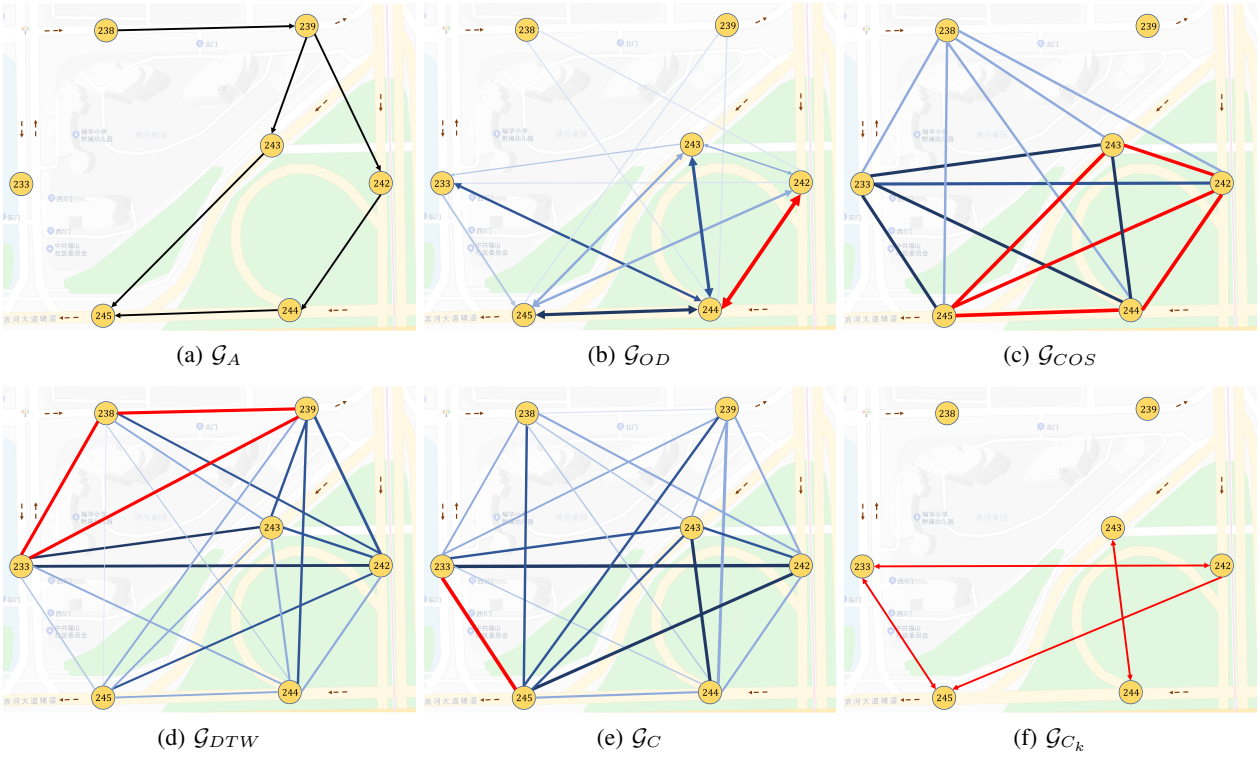
To discuss the effect of window size $w$ in the latent graph learning model, we train different $\mathcal{G}_C$ using $w$ from 2 to 10. To avoid the influence of $k$-NN filtering, we skip this step and directly use $\mathcal{G}_C$ as the input graph to compare the traffic flow prediction results w.r.t. different $w$. As shown in Figure 5, we use GWNET as an example and draw the prediction errors. The parameter $w$ controls how many historical steps will be used to predict the trajectory next-hop, leading to different road embedding matrix $E$, and then different $\mathcal{G}_C$. If it is small, the LSTM cannot learn the long-range spatial dependencies, which will lead to bad prediction results, especially for the 60-minute ahead prediction. Also, if it is too large, the simple LSTM structure cannot effectively handle such a long sequence, which will still lead to high prediction errors. According to the figure, the prediction errors reach the minimum values when $w = 5$. Thus, it is a suitable value for the latent graph learning model.

Secondly, we analyze the choice of $k$ when calculating $\mathcal{G}_{C_k}$, using GWNET as an example. As shown in Figure 6, the change of prediction errors do not have an obvious pattern with the increase of $k$ from 2 to 20. This is because the value in $\mathcal{G}_{C_k}$ is either zero or one, and a small change can lead to totally different results. There is no general rule for us to determine the value of $k$. Therefore, according to the figure, we choose different $k$ for each GCN prediction model, which is mentioned in the experiment settings.

### *F. Case Study*

The visualization of the six graphs used in the experiment is given in Figure 7. We select seven roads $r_{233}, r_{238}, r_{239}, r_{242}, r_{243}, r_{244}, r_{245}$ to visualize the different relations among them in the graphs. Figure 7a is the original adjacency graph of the road network, where $r_{233}$ is not connected with any road. This is caused by the driving directions of the seven roads, which are marked as brown arrows. However, in $\mathcal{G}_C$, $r_{233}$ are highly correlated with other roads. The red edge color stands for a very high edge weight and the same for the next four graphs. Therefore, referring to Figure 7f, $\mathcal{G}_{C_k}$ becomes completely different to $\mathcal{G}_A$, where the connectivity is learned from trajectories. The $\mathcal{G}_{OD}$ graph in Figure 7b is a summary for origin-destination patterns, where $r_{244}$ seems to be a hot spot for taxis. For the $\mathcal{G}_{COS}$ graph in Figure 7c, the roads $r_{242}, r_{243}, r_{244}, r_{245}$ that locate on the main roads of this region are highly related, while $r_{238}, r_{239}$ have

Fig. 7: Visualization of input graphs for GCN.




(a) $\mathcal{G}_A$ (b) $\mathcal{G}_{OD}$ (c) $\mathcal{G}_{COS}$

(d) $\mathcal{G}_{DTW}$ (e) $\mathcal{G}_C$ (f) $\mathcal{G}_{C_k}$

smaller correlations with other roads. This is influenced by the similarity of their flow patterns. Figure 7d shows the DTW similarity of these roads, which is negatively related to the DTW distance. The distance among $r_{233}, r_{238}, r_{239}$ are small since they have the same upstream roads according to the driving direction. In general, the various road relations cause different performances in the experiment, and our proposed $\mathcal{G}_C$ and $\mathcal{G}_{C_k}$ can best reflect the real road correlations.

Figure 8a provides the visualization of road correlations among $r_{225}$ to $r_{245}$. We selected these roads due to space limitation. In the figure, instead of Softmax normalization, the correlation values are scaled by the MinMax scaling algorithm without the diagonal and the diagonal values are manually set as 1. A darker color represents a higher correlation. Through this figure, we provide two case studies on the learned latent road correlation, which are shown in Figure 8b, and Figure 9, respectively.

**Case 1. Road Correlations of $r_{233}$.** For the visualization on the road network, we select roads $r_{155}$ to $r_{245}$, in total 91 roads, which lie in the bottom half of the road network, to visualize the road correlations among them. The road correlation values are taken directly from correlation graph $\mathcal{G}_C$, which is also the same as row 233 in Figure 8a. As a result, the Figure 8b shows the correlations of road $r_{233}$ with other roads. The blue road at the bottom right corner stands for $r_{233}$, while other roads are drawn by the color bar according to their correlation values. A darker color means that the corresponding road has a higher correlation with $r_{233}$, and vice versa. If a road is in light yellow in the figure, then it does not correlate with $r_{233}$ at all. From the figure, we can observe that the nearby roads, which are $r_{231}, r_{245}, r_{242}, \ldots$, are highly correlated with $r_{233}$. It meets our expectations because according to the map, they have the same upstream and downstream roads as $r_{233}$, which means our model successfully captured the low-order dependencies. What's more, some long-distance roads, such as $r_{200}, r_{181}, r_{177}$, also have considerable correlations with $r_{233}$, which are the high-order dependencies captured by the model. We investigate these two locations on the real map and find that $r_{233}$ lies in a residential area, and $r_{200}, r_{181}, r_{177}$ lie in a shopping area. This indicates that the learned road correlations meet the daily travel patterns of people.

**Case 2. Surrounding POIs of Road $r_{228}$.** Referring to Figure 8a, row 228 has the biggest row sum, i.e. $r_{228}$ has the largest correlations with other roads. We draw the position of $r_{228}$ and mark the areas nearby on the map as Figure 9, from which we can see the surrounding Point-Of-Interests (POIs) of $r_{228}$ are Shenzhen Convention and Exhibition Center, LianHua Middle School, Excellence Century Center, Shenzhen ICC Tower, etc. Road $r_{228}$ is also directly connected to the main road of this region, Binhe Boulevard. These POIs demonstrate that $r_{228}$ serves an important role in the road network, causing the high correlation values of $r_{228}$. Therefore, the proposed road correlation can also help us to specify the road importance and let the traffic managers pay more attention to high-correlated roads.

In conclusion, these case experiments prove not only the

Fig. 8: Visualization of learned road correlations.


(a) Visualization for $\mathcal{G}_C$.

(b) Road correlations of $r_{233}$.

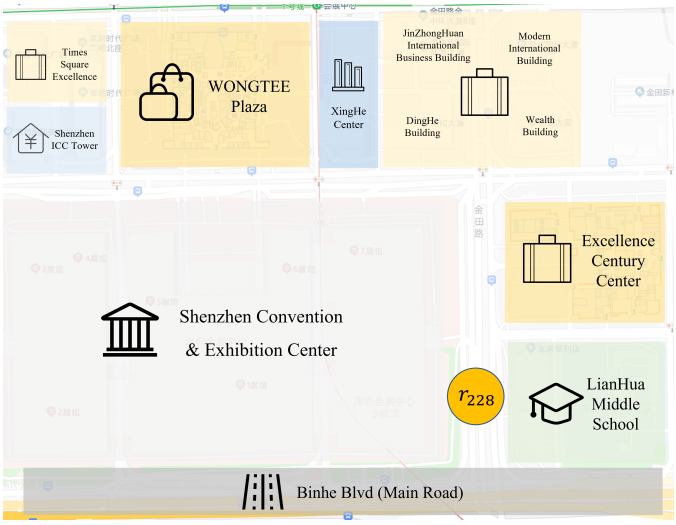



Fig. 9: Surrounding POIs of $r_{228}$.

rationality of our thesis on the latent correlation graph, but also the effectiveness of our proposed model and methods.

## VI. Conclusion

In this paper, we first investigated the existing road network representations in traffic applications. The shortage of them is that the spatial dependency is expressed only by the static relationship among roads. Therefore, we proposed a novel road network representation that extracts dynamic road correlations through vehicle trajectories. The trajectory next-hop prediction model was built to learn road embeddings based on an LSTM neural network. The latent road correlation graph $\mathcal{G}_C$ was calculated by the embedding matrix $E$ in the model. Based on that, the $k$-NN approach for the computing of $\mathcal{G}_{C_k}$ was given. Traffic flow prediction was used as a downstream task to verify the proposed latent graph. Extensive experiment results proved the effectiveness and robustness of our proposed methods. Following, some case studies demonstrated the rationality of the proposed latent graphs.

The proposed latent correlation graph is still not dynamic enough, since it is learned by the trajectories over the whole time range. To strengthen its temporality, we will consider extending our current work to time-varying road correlations. In addition, it is worth trying to utilize more modern sequence prediction models for the learning of road correlations. And to improve the robustness of the latent correlation graph, we will investigate how to handle GPS-sparse regions and refine our model.

## References


[1] J. Zhang, F.-Y. Wang, K. Wang, W.-H. Lin, X. Xu, and C. Chen, "Data-driven intelligent transportation systems: A survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 4, pp. 1624–1639, 2011.

[2] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics and intelligent laboratory systems*, vol. 2, no. 1-3, pp. 37–52, 1987.

[3] J. Ye, R. Janardan, and Q. Li, "Two-dimensional linear discriminant analysis," *Advances in neural information processing systems*, vol. 17, 2004.

[4] S. L. Robinson and R. J. Bennett, "A typology of deviant workplace behaviors: A multidimensional scaling study," *Academy of management journal*, vol. 38, no. 2, pp. 555–572, 1995.

[5] P. Newson and J. Krumm, "Hidden markov map matching through noise and sparseness," in *Proceedings of the 17th ACM SIGSPATIAL international conference on advances in geographic information systems*, 2009, pp. 336–343.

[6] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2014, pp. 701–710.

[7] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, 2016, pp. 855–864.

[8] D. Guthrie, B. Allison, W. Liu, L. Guthrie, and Y. Wilks, "A closer look at skip-gram modelling." in *LREC*, vol. 6. Citeseer, 2006, pp. 1222–1225.

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

[10] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," *Advances in neural information processing systems*, vol. 30, 2017.

[11] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *International Conference on Learning Representations*, 2018.

[12] N. Wu, X. W. Zhao, J. Wang, and D. Pan, "Learning effective road network representation with hierarchical graph neural networks," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, pp. 6–14.

[13] Y. Li, K. Fu, Z. Wang, C. Shahabi, J. Ye, and Y. Liu, "Multi-task representation learning for travel time estimation," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 1695–1704.

[14] J. Dai, B. Yang, C. Guo, and Z. Ding, "Personalized route recommendation using big trajectory data," in *2015 IEEE 31st international conference on data engineering*. IEEE, 2015, pp. 543–554.

[15] V. Kostov, J. Ozawa, M. Yoshioka, and T. Kudoh, "Travel destination prediction using frequent crossing pattern from driving history," in *Proceedings. 2005 IEEE Intelligent Transportation Systems, 2005.* IEEE, 2005, pp. 343–350.

[16] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, "Graph wavenet for deep spatial-temporal graph modeling," *arXiv preprint arXiv:1906.00121*, 2019.

[17] H. Shi, Q. Yao, Q. Guo, Y. Li, L. Zhang, J. Ye, Y. Li, and Y. Liu, "Predicting origin-destination flow via multi-perspective graph convolutional network," in *2020 IEEE 36th International Conference on Data Engineering (ICDE)*. IEEE, 2020, pp. 1818–1821.

[18] K. Lee and W. Rhee, "Ddp-gcn: Multi-graph convolutional network for spatiotemporal traffic forecasting," *Transportation Research Part C: Emerging Technologies*, vol. 134, p. 103466, 2022.

[19] Z. Wang, T. Xia, R. Jiang, X. Liu, K.-S. Kim, X. Song, and R. Shibasaki, "Forecasting ambulance demand with profiled human mobility via heterogeneous multi-graph neural networks," in *2021 IEEE 37th International Conference on Data Engineering (ICDE)*. IEEE, 2021, pp. 1751–1762.

[20] D. Yin, R. Jiang, J. Deng, Y. Li, Y. Xie, Z. Wang, Y. Zhou, X. Song, and J. S. Shang, "Mtmgnn: Multi-time multi-graph neural network for metro passenger flow prediction," *GeoInformatica*, pp. 1–29, 2022.

[21] M. Li, P. Tong, M. Li, Z. Jin, J. Huang, and X.-S. Hua, "Traffic flow prediction with vehicle trajectories," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 1, 2021, pp. 294–302.

[22] R. Jiang, X. Song, Z. Fan, T. Xia, Q. Chen, Q. Chen, and R. Shibasaki, "Deep roi-based modeling for urban human mobility prediction," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 2, no. 1, pp. 1–29, 2018.

[23] Z. Fan, X. Song, T. Xia, R. Jiang, R. Shibasaki, and R. Sakuramachi, "Online deep ensemble learning for predicting citywide human mobility," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 2, no. 3, pp. 1–21, 2018.

[24] R. Jiang, X. Song, Z. Fan, T. Xia, Z. Wang, Q. Chen, Z. Cai, and R. Shibasaki, "Transfer urban human mobility via poi embedding over multiple cities," *ACM Transactions on Data Science*, vol. 2, no. 1, pp. 1–26, 2021.

[25] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, "Graph neural networks: A review of methods and applications," *AI Open*, vol. 1, pp. 57–81, 2020.

[26] 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*, 2018, pp. 3634–3640.

[27] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," *arXiv preprint arXiv:1511.07122*, 2015.

[28] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," in *International Conference on Learning Representations*, 2018.

[29] 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.

[30] M. Haklay and P. Weber, "Openstreetmap: User-generated street maps," *IEEE Pervasive computing*, vol. 7, no. 4, pp. 12–18, 2008.

[31] C. Yang and G. Gidofalvi, "Fast map matching, an algorithm integrating hidden markov model with precomputation," *International Journal of Geographical Information Science*, vol. 32, no. 3, pp. 547 – 570, 2018.

[32] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[33] R. Jiang, D. Yin, Z. Wang, Y. Wang, J. Deng, H. Liu, Z. Cai, J. Deng, X. Song, and R. Shibasaki, "Dl-traff: Survey and benchmark of deep learning models for urban traffic prediction," in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 2021, pp. 4515–4525.

[34] D. J. Berndt and J. Clifford, "Using dynamic time warping to find patterns in time series." in *KDD workshop*, vol. 10, no. 16. Seattle, WA, USA:, 1994, pp. 359–370.