Origin-Destination Convolution Recurrent Network: A Novel OD Matrix Prediction Framework

Jiayu Chang[1], Tian Liang[1], Wanzhi Xiao[1], and Li Kuang[1(✉)]

Central South University, South Lushan Road 932, Changsha Hunan 410083 China
{8209200123,leungtien,wanzhixiao,kuangli}@csu.edu.cn

Abstract. Origin-Destination(OD) Matrix Prediction is an important part of public transportation service which aims to predict the number of passenger demands from one region to another and capture the passengers' mobility patterns. This problem is challenging because it requires forecasting not only the number of demands within a region, but the origin and destination of each trip as well. To address this challenge, we propose an effective model, ODCRN(**O**rigin-**D**estination **C**onvolution **R**ecurrent **N**etwork) which incorporates traffic context and bi-directional semantic information. First, we obtain the semantic embedded features of the region as the static traffic context by the Node2vec algorithm, and the traffic flow of the region is counted as the dynamic traffic context. Second, we construct two adjacency matrices which represent *origin-destination* and *destination-origin* travel demands within urban areas respectively based on the OD matrices of each time slot, and use the graph convolutional network to aggregate traffic context information of the semantic neighbors in both directions. Then, we use a unit constructed by GRU and the graph convolution network to capture the spatial-temporal correlations of the input data. Finally, we use those correlations and traffic contexts to predict the OD matrix for the next time slot. Our model is evaluated on *TaxiNYC* and *TaxiCD* datasets, and experimental results demonstrate the superiority of our ODCRN model against the state-of-the-art approaches.

Keywords: OD matrix prediction · Graph diffusion convolution · Spatial-temporal data.

1 Introduction

Recently, ride-hailing applications are becoming prevalent choices for daily commutes, such as Didi, UCAR, and Uber, which aim to provide passengers with convenient ride services and improve the efficiency of public transportation. To provide high-quality services and achieve company profits, ride-hailing platforms need to fully understand the passenger demands in real-time, which helps avoid empty drives (i.e., driving without passengers). Therefore, instead of merely forecasting the possible number of passenger demands within a region, it is rather

important to gain knowledge of passenger demands in terms of the origin and destination of each trip. Because the demand quantity between two regions at different time slots also takes mining useful mobility patterns into consideration. If Origin-Destination (OD) travel demand can be found earlier, popular destinations and travel routes can be provided for travel service providers, and vehicles can be scheduled in advance.

OD matrix prediction has received increasing attention because of its importance. Existing studies [1,14,21] predict the OD travel demand between all regions of the city by constructing an OD matrix. As shown in Figure 1, we regard the urban region as a large rectangle, and then divide it into $m \times n$ regions, making the dimension of the OD matrix in each time slot reach $N \times N(N = m \times n)$. And then form an OD matrix sequence according to a temporal order, where $M_{i,j}^{(t)}$ in the OD matrix represents the number of trips from region r_i to region r_j from time $t - 1$ to time t. OD matrix prediction aims to predict the OD matrix of a future moment based on the OD matrix of multiple historical moments.
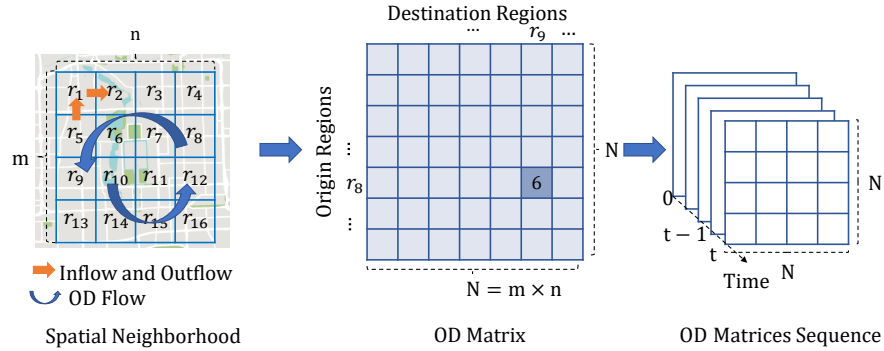

Fig. 1. Spatial neighborhood and OD matrix.

Although recent works consider combining traffic flows to utilize mobility patterns into OD matrix prediction, it is still a challenging problem, affected by the following aspects:

(1) Directed semantics of travel demand. There is a directed semantic relationship between OD travel demands among urban regions. In the left diagram of Figure 2, *A, B, and C* represent the *office region, residential region, and residential region*, respectively. During the morning rush hour, there are similar travel demands from *B to A* and *C to A*. There are more travel demands with *A as the destination*, but less travel demand with *A as the origin*, which indicates that the semantic neighbors in the two directions between departure and arrival of *A* is different in the same period. The right diagram of Figure 2 shows the *10-day OD demand curves between A and B* revealing a large number of taxi

demands *from B to A*. At the same time, due to the marginality of region *B*, the demand for taxis *from A to B* appears to be random. This non-periodic and non-stationary curve increases the difficulty of forecasting.
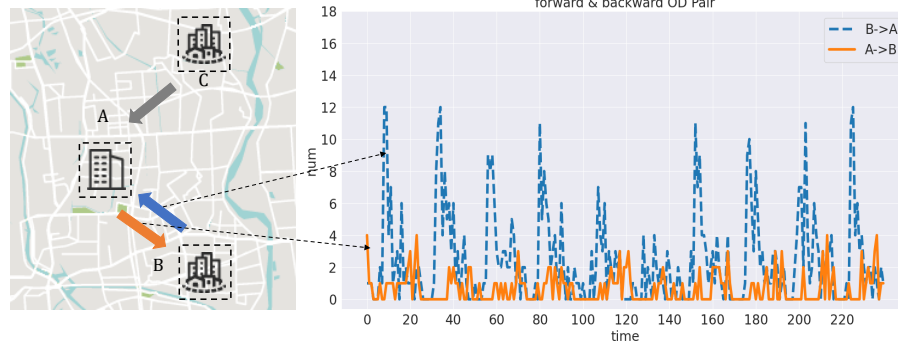
Fig. 2. OD spatial correlation (left) and OD flow time series (right) among different urban regions

(2) Various Traffic Context. In the existing OD matrix prediction studies, in addition to the number of travel demands between regions, other information, such as the traffic context of the origin and destination regions, has not been fully considered. Traffic context includes *static traffic context information* and *dynamic traffic context information*. Different traffic contexts contain different contextual information. For example, static POI information reflects the function of the region while the dynamic traffic context of the region, such as inflow and outflow traffic, reflects the dynamic traffic conditions of the region.

To tackle these challenges, we propose an effective model named ODCRN, to collectively predict the OD matrix in the following time slot more accurately and more efficiently. The primary contributions of this paper can be summarized as follows:

(1) In order to capture the directional information of travel demand, we construct two OD matrices from the origin to the destination(*forward*) and from the destination to the origin(*backward*) to capture the forward and backward information by *bi-directional graph diffusion convolution network*. Then we combine them together to integrate the bi-directional semantic information.

(2) We comprehensively consider static and dynamic context information by taking POI information which reflects regional functions as static traffic context information by establishing a bootstrap to better represent origins and destinations, and traffic flow information between regions as dynamic traffic context information. Then we capture the features through dynamic

and static traffic context learning networks to collaborate with OD matrix prediction.

(3) Extensive experiments on two open datasets, *TaxiNYC* and *TaxiCD*, demonstrate the proposed ODCRN model outperforms baselines.

2   Preliminaries and Related Work

2.1   Definitions

We first introduce several basic concepts to formulate the OD matrix prediction problem. We assume the target city is partitioned into subareas as regions and regard them as graph nodes.

Definition 2-1 OD Matrix We define an OD matrix at time slot t as a 2-dimensional tensor $M_t \in \mathbb{R}^{N_l \times N_l}$, where N_l is the total number of regions in the city. It can also be defined as $M_t = \{m_{i,j}, 1 \le i, j \le N_l\}$, where $m_{i,j} \in \mathbb{R}$ represents specific figure of traffic flow from region i to region j at the time slot t. OD Matrices Sequence is defined as a 3-dimensional tensor $M = [M_1, \cdots, M_{Nt}] \in \mathbb{R}^{N_t \times N_l \times N_l}$, where N_t is the total number of slots of historical traffic data.

Definition 2-2 Semantic Neighbors $m_{i,j}$ represents the number of travels from region v_i to region v_j. v_i and v_j are the OD semantic neighbors in this particular travel demand. v_i and v_j needn't be geographically adjacent.

Definition 2-3 Static Traffic Context We define the traffic graph of city as $G = \{V, A\}$ where $V = [v_1, \cdots, v_{N_l}]$ and $|V| = N_l$. $v_i \in V$ is a node on the traffic graph which represents a region in the city. 2-dimensional tensor $S \in \mathbb{R}^{N_l \times F}$ represents the static traffic context on all regions, such as POIs and embedding features.

Definition 2-4 Dynamic Traffic Context We define the dynamic traffic context in regions as a 3-dimensional tensor $X = [X_1, \cdots, X_{Nt}] \in \mathbb{R}^{Nt \times Nl \times Dt}$, where the D_t represents the features' dimension of dynamic traffic context. For example, when we use inflow and outflow as features of context, features' dimension $D_t = 2$.

2.2   Origin-Destination Matrix Prediction

For a given time slot t, using OD matrices for the past p time slots $[M_{t-p+1}, \cdots, M_t] \in \mathbb{R}^{P \times N_l \times Nl}$, dynamic traffic contexts of every region for the past p time slots $[X_{t-P+1}, \cdots, X_t] \in \mathbb{R}^{P \times N_l \times D_t}$, static traffic context of every region $S \in \mathbb{R}^{N_l \times F}$ and adjacent matrix which represents the region's geographic proximity $A \in \mathbb{R}^{N_l \times N_l}$ as inputs, to predict OD matrix $M_{t+1} \in \mathbb{R}^{N_l \times N_l}$ of one-time slot in the future.

Generally, OD Matrix prediction task [2,3,4] can be divided into two categories, static approaches and dynamic approaches. Static approaches [5,6] consider the traffic flow as independent time slots and ignore the spatial-temporal information by calculating the average traveling demand to predict the OD matrix over a long period of time. Dynamic approaches [7] take the time-variant traffichttps://www.overleaf.com/project/6424167639f3dccab22016fa flow information under consideration so they can be used for the management of the traffic flow and the dynamic path induction.

In recent years, an increasing number of neural network approaches, especially Graph Neural Networks(GNN) [9] have been applied to OD matrix prediction tasks. Since urban transportation networks can be seen as graph structures, GNN can naturally deal with the spatial relation of traffic flows in non-Euclidean city traffic networks. Those approaches consider sensors as graph nodes [12], the data at fixed time intervals as node features and the geographical proximity, distance, traffic similarity or road connectivity between sensors or regions as edges between two nodes. Researchers such as Wang and Ke [1,14] proposed graph embedding methods to embed the grid and combine it with LSTM to predict the OD matrix. Multi-task learning traffic prediction approaches [15,16,17,18] were proposed for both regional traffic flow and OD matrix prediction tasks. Liu et al. [19] constructed station distance and similarity in traffic patterns for predicting subway OD trip traffic based on a combination of physical and virtual approaches. Zheng et al. [1] proposed a method to predict the OD matrix via graph convolution and Liu et al. [8] combine the spatial-temporal context to predict taxi OD demand.

We found that they rely heavily on the construction of traffic graphs to capture the spatial correlation of flows between geo-locations. Moreover, due to the functional differences between locations and the real-time mobility and variability of traffic flows, most graph neural networks fail to capture the dynamic spatial correlations between urban locations, as well as the hierarchical spatial correlations of both local traffic correlations and global traffic similarities.

3   The ODCRN Framework

In this section, we present the detailed architecture of the proposed model OD-CRN. As the Figure 3 shows, the model consists of three parts. The first part is the spatial-temporal information capture model, which is constructed by the Bi-GDC-GRU unit based on the architecture of RNN. As Figure 4(a) shows, each GRU unit integrates the bi-direction diffusion graph convolution (in Figure 4(b)) to capture the temporal and spatial correlations of the OD matrices. The other two components are dynamic and static traffic context learning networks, which are used to capture the traffic context features associated with each time slot and input them as node features TC_t into the Bi-GDC-GRU unit. We use CNN with the kernel size of 1×1 to predict the OD matrix of the next time slot M_{t+1} by using the last hidden state h_t in the prediction phase.

The inputs to the model include the OD matrix of the historical p time segments $[M_{t-p+1}, \cdots, M_t]$ and traffic context information for historical p time slot $[TC_{t-p+1}, \cdots, TC_t]$. As Figure 3 shows, the traffic context learning network module learns regionally relevant traffic contextual information and consists of two sub-networks, including a static context learning network and a dynamic context learning network. We use Node2vec [20] embedding features which illustrate the regional structural features as input of the static traffic context learning network. To predict the number of passengers from one region to another region, we use the inflows and outflows of all regions in each time slot as inputs of the dynamic traffic context learning network.

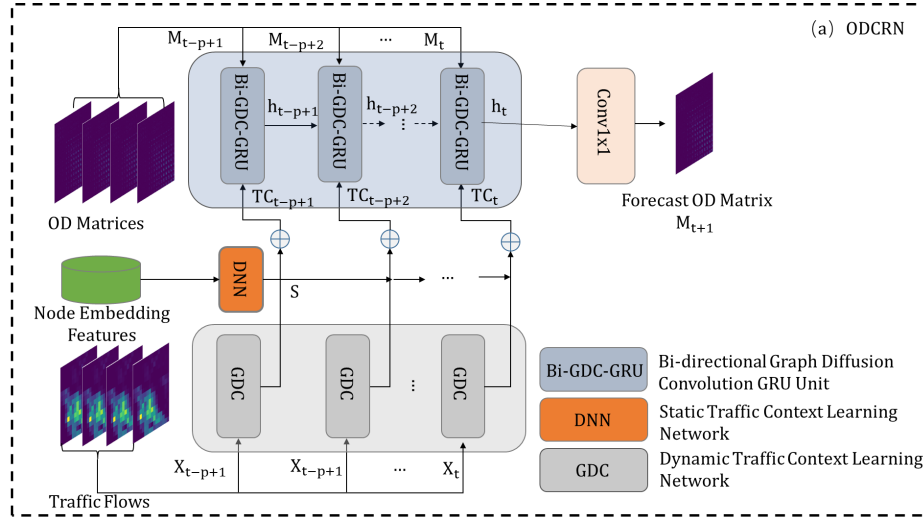### 3.1   Constructing Traffic Context from Historical Data

Fig. 3. ODCRN architecture

Static Traffic Context Learning Network Module The attributes of the region itself, such as POI, population and other information, affect its traffic flow and influence its fluency as the origin or destination for commutes. We pretrain the embedding features of each node with Node2vec algorithm as the static traffic contexts for each node. Node embedding features represent the structural properties of the nodes on the graph and the similarity between nodes.

After constructing the static traffic context, we use a deep neural network DNN as a Static Context Learner (SCL) to map the static traffic context into

node hidden states as follows:

$$h_1 = SCL\,(S) = DNN\,(S) \tag{1}$$

In Equation1, $h_1 \in \mathbb{R}^{N_l \times F_1}$ is the output of the static traffic context learning network SCL and the DNN is a multi-layers deep neural network.

Dynamic Traffic Context Learning Network Module We count the inflow and outflow traffic in each subarea of the city for each time segment from the historical travel records. We use $X = [X_1, \cdots, X_{Nt}] \in \mathbb{R}^{N_t N_l D_t}$ as regional dynamic traffic context. Since the inflow and outflow traffic of a region are affected by adjacent regions, we use a graph convolution network as a Dynamic Context Learner (DCL) network to capture spatial correlations. DCL maps the dynamic traffic context of all nodes in each time slot into hidden state representations:

$$h_2 = \ DCL\,(X_t) = GDC\,(A, X_t, W) = \sum_{k=0}^{K} A^k X_t W_k \tag{2}$$

In Equation2, GDC denotes the graph diffusion convolution [13] operation, which will be introduced in the following section 3.2. $X_t \in \mathbb{R}^{N_l \times D_t}$ is the dynamic traffic context at time slot t of the input, and W is the graph convolution parameters.

For each time slot t, we combine those two types of traffic context as shown in the Equation3 :

$$\text{TC}_t = SCL\,(S)\ \ ||\ \ DCL\,(X_t) = \ h_1\ ||\ h_2 \tag{3}$$

3.2   Capturing Spatial-temporal Correlations by Bi-GDC-GRU Unit

We propose Bi-directional Graph Diffusion Convolution GRU (Bi-GDC-GRU) unit to capture correlations of spatial-temporal information. Demands of different OD travel directions might be different at the same time. For example, office regions have more arrival demand in the morning and less departure demand. Figure 5(a) and Figure 5(b) show that the OD semantic neighbors departing from and arriving at region r2 at the same time are different. Therefore, in order to partially solve the sparsity problem of the OD matrix, we construct the semantically adjacent edges of r_i in two directions(both in and out), which can aggregate more OD semantic neighbor node information.

Specifically, for each input time t, we use the OD matrix M_t to construct two directed weighted adjacency matrices as shown in Figure 4(b) and Figure 4(c) as graphs to aggregate neighbor node information. The forward adjacency matrix is expressed as $A_f^{(t)} \in \mathbb{R}^{N_l \times N_l}$ and the backward adjacency matrix is expressed as $A_b^{(t)} \in \mathbb{R}^{N_l \times N_l}$:

$$A_f^{(t)} = Norm\,(M_t) \tag{4}$$

$$A_b^{(t)} = A_f^{(t)}.T \tag{5}$$
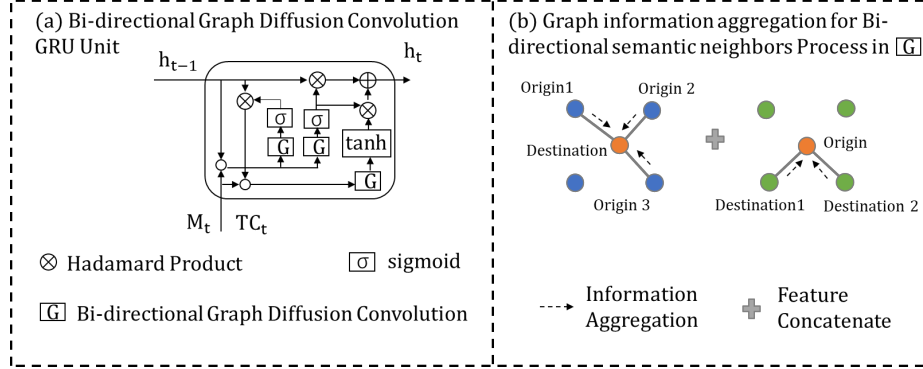
Fig. 4. Details of Bi-GDC-GRU unit (in Figure 4(a)) and an example of graph information aggregation process in Bi-GDC-GRU(in Figure 4(b))

In Equation 4, *Norm* is to normalize the OD matrix M_t to prevent the gradient explosion problem after the graph convolution operation. In Equation 5, $.T$ is the transpose operation. Since there is some correlations between the region set as the origin and set as the destination, using the forward and backward adjacency matrix to simultaneously aggregate the semantic neighbor node features on the OD pair starting from node v_i and arriving at node v_i.
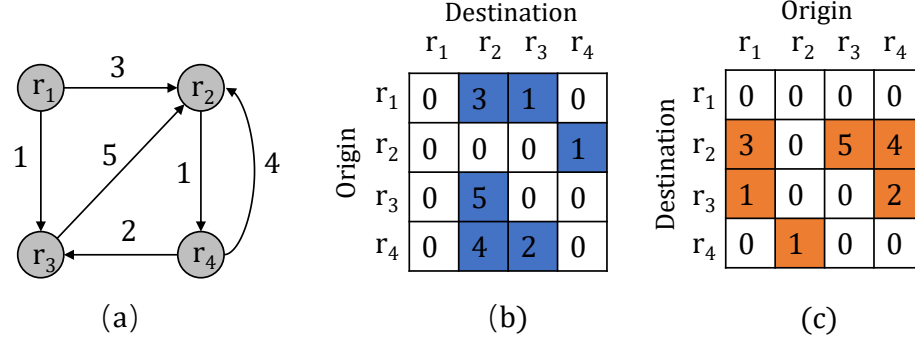

Fig. 5. An example of forward and backward OD matrix. Figure 5(a) shows the trip times and directions of the four nodes. Figure 5(b) and Figure 5(c) show the corresponding OD matrix. The travel demand relationship from r2 to r1, r2, r3, and r4 represents as [0,0,0,1], and the number of trips arriving at r2 is [3,0,5,4].

After constructing the forward and backward adjacency matrices, we use bi-directional diffusion graph convolution to capture the spatial correlation between OD pairs, as shown in Equation 6:

$$h^{(t)} = \sum_{k=0}^{K} \left(A_f^{(t)}\right)^k \mathrm{TC}_t W_{k1} + \left(A_b^{(t)}\right)^k \mathrm{TC}_t W_{k2} \tag{6}$$

TC_t represents the traffic context at time t, and W is the graph convolution kernel parameter. For time t, the diffusion graph convolution of Equation 6 is defined as a finite number of directed diffusion processes on the graph, k is the number of diffusions, and $(A_f^{(t)})^k$ represents the diffusion probability to the $k-th$ hop neighbor. Figure 4-6 shows the convolution process of the forward diffusion graph. The central node is represented by the aggregation and summation of the information of neighboring nodes of different hops. The diffusion probability $A_f^{(t)}$ indicates the strength of the edge connection between nodes. The OD traffic of nodes is bi-directional, and the forward and backward diffusion graph convolution in Equation 6 can aggregate the region as the information of multi-hop semantic neighbor nodes in the two directions of the origin and the destination.

The graph convolution diffusion process of the node in the center   Step 0   Step 1   Step 2   Step k
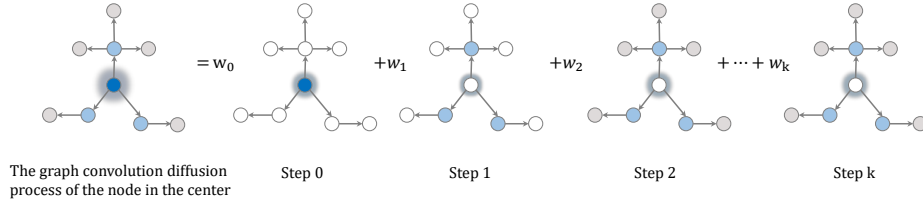
Fig. 6. An schematic of k-steps forward graph diffusion convolution

Then, the time correlations of the OD matrix at different moments are captured by the GRU units based on the recurrent neural network. Specifically, we use GRU to capture the temporal correlation between OD matrices, and replace the matrix multiplication operation in GRU with the bi-directional diffusion graph convolution operation in Equation 6.

$$r^{(t)} = \sigma(\Theta_r *_G \left[TC_t, H^{(t-1)} \right] + b_r) \tag{7}$$

$$u^{(t)} = \sigma \left(\Theta_u *_G \left[TC_t, H^{(t-1)} \right] + b_u \right) \tag{8}$$

$$c^{(t)} = tanh(\Theta_c *_G [TC_t, r^{(t)} \odot H^{(t-1)}] + b_c) \tag{9}$$

$$H^{(t)} = u^{(t)} \odot H^{(t-1)} + (1 - u^{(t)}) \odot c^{(t)} \tag{10}$$

In Equation 10, $H^{(t)}$ is the output of the GRU unit at time t. $*_G$ is the diffusion map convolution operation represented by equation (4-6). In Equation 7, 8, and 9, Θ is the graph convolution kernel parameter, σ is the Sigmoid activation function, and $\odot$ is the Hadamard product. The reset gate $r^{(t)}$ and the update gate $u^{(t)}$ of the GRU unit at time t are used to control the forgetting information and control output respectively, and the input is the traffic context TC_t at the current time t and the previous time. The hidden state $H^{(t-1)}$ output by the

GRU unit is concatenated and used as node features to input into the diffusion graph convolution represented by Equation 6. Diffusion Graph Convolution GRU units can capture both temporal and spatial correlations.

In the model prediction stage, we regard the hidden state output by the last GRU unit as the input of the prediction network. Since we need to predict the OD matrix next moment, we adopt a single-layer convolutional neural network with a 1×1 convolution kernel as the prediction network, which is shown in Equation 11:

$$\hat{M}_{t+1} = \Theta *_C \left(H^{(t)} \right) \tag{11}$$

Where $H^{(t)} \in \mathbb{R}^{N \times F}$ is the hidden state output by the last GRU unit, $*_C$ is the convolution operation, Θ is the convolution kernel parameter, $\hat{M}_{t+1} \in \mathbb{R}^{N_l \times N_l}$ is the predicted OD matrix at the next moment.

3.3   Loss Function of OD Matrix Prediction

We use Mean Square Error (MSE), a common loss function in regression tasks, to back-propagate the error and optimize the model parameters. As shown in Equation (4-12):

$$L\left(W_\theta \right) = MSE \left(M_{t+1}, \ \hat{M}_{t+1} \right) = \frac{1}{N_l * N_l} \sum_{i=1}^{N_l} \sum_{j=1}^{N_l} \left(m_{i,j} - \hat{m}_{i,j} \right)^2 \tag{12}$$

W_θ are the parameters of the whole model. $M_{t+1}, \hat{M}_{t+1} \in \mathbb{R}^{N_l \times N_l}$ are the true and predicted values of the OD matrix at the next moment, respectively. N_l is the total number of nodes, namely the number of regions in the city.

4   Experiment

4.1   Datasets and Evaluation Metrics

We use taxi order data from New York City and Chengdu for the experiments. The raw data information includes boarding time, alighting time, boarding longitude, boarding latitude, alighting longitude and alighting latitude.

We divide the whole city into $m \times n$ regions. New York City is divided into $256(16 \times 16)$ regions while Chengdu City is divided into $256(16 \times 16)$ regions as well. Firstly, based on the data of orders, the number of orders corresponding to the boarding and alighting regions per hour is counted and used to construct an OD Matrix, which is arranged in chronological order to form a sequence of OD Matrices with a dimension of $T * N * N$, where the T is the total number of time periods and $N = m \times n$ is the total number of regions. Specific information on the dataset is shown in Table 1:

---

[1] https://www.nyc.gov/site/tlc/about/tlc-trip-record-data.page
[2] https://outreach.didichuxing.com/research/opendata/

Table 1. Description of Dataset

Dataset Name	TaxiNYC[1]	TaxiCD[2]
Data Type	Taxi Order	Taxi Order
City	New York	ChengDu
Longitude range	-74.02-73.95	104.02-104.12
Latitude range	40.67-40.77	30.62-30.70
Time range	2015/1/1-2015/4/30	2016/11/1-2016/11/30
Total number of time slots	2880	720
Length of unit time period	1 hour	1 hour
Number of grids	16×16	16×16
Static Traffic Context Information	Node2vec Node Embedding	Node2vec Node Embedding
Dynamic Traffic Context Information	Regional inflow/outflow	Regional inflow/outflow

At the same time, we construct two kinds of traffic context information:

(1) Static Traffic Context Information. Using the data of orders within two weeks, we construct the graph by using the starting grid and the ending grid as two nodes on an edge. The number of orders is the weight of the edge. The embedding vector for each grid was pre-trained with Node2vec.

(2) Dynamic Traffic Context Information. The number of boardings and alightings corresponding to each zone per hour is counted as the inflow and outflow of the zone.

The OD matrix prediction task uses historical data from the past 8 hours (8 unit time slots) to predict the future OD matrix of 1 hour (1 time slot). The dataset is divided in chronological order with training, validation and test set at the ratio of 8:1:1. We evaluate the performance by two commonly used regression task evaluation indicators, root mean square error(RMSE) and mean absolute error(MAE).

4.2   Parameter Setting

The hyperparameters of our model are set as follows:

(1) Dimension of the Node2vec nodes embedding feature is 10. The static context learning network consists of 2 layers neural network where the number of hidden layer units in each layer is 32. The dynamic context learning network is a 3 layers GDC network, and the number of units in each layer is 32.

(2) The ODCRN has a total of two layers of recurrent networks, where each recursive cell contains two layers of diffusion map convolutional networks. The number of hidden layer cells is 32, and the number of GDC diffusion steps is set to 2.

(3) We use Adam optimizer to train the model. The learning rate during the training stage is 0.01 with a batch size of 128. In addition, to prevent overfitting, the regularization parameter is set to 0.0001 and an early stop mechanism is used.

4.3   Spatial-temporal Data Preprocess

For OD matrix prediction, we use historical data from the traffic flows' dataset to generate static and dynamic traffic contexts as the input of traffic context, and generate the OD matrices for the past 8 hours as input of the historical OD matrices sequence.
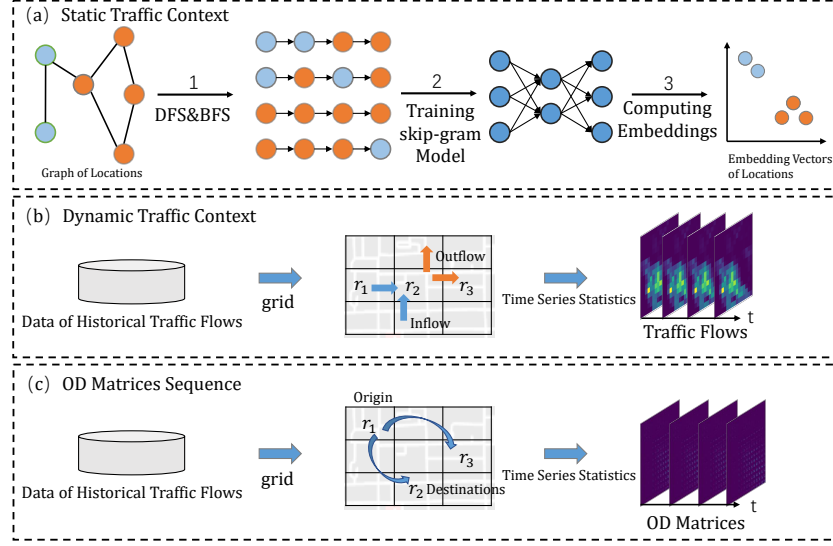
Fig. 7. Traffic context and OD matrix construction process

Static Traffic Context Data According to the statistics of the start point, end point and traveling counts between them, we use Node2vec to train the embedding features of each region as our static traffic context, including graph structural characteristics and node similarity.

Dynamic Traffic Context Data As shown in Figure 7(b), we count the inflow and outflow of each region in hourly intervals as our dynamic traffic context data. regions with high inflow flows tend to be the end of the commute, while those with high outflow flows tend to be the origin of the commute.

OD Matrices Sequence Since inter-regional commuting takes time, the OD matrices are also counted in hours. We first grid the data to find the semantic neighbors of each region. Then we perform temporal statistics on the order start and end points of each region to determine the weights of the adjacency matrix. We regard the OD matrix as the adjacency matrix and use it as our input of the Bi-GDC-GRU cell to capture the spatial association of OD semantic neighbors.

4.4   Comparison and Analysis of Model Prediction Accuracy

Table 2 shows the prediction results of each method on the TaxiNYC and TaxiCD datasets. It is clear that the traditional models HA and ARIMA have poor prediction results, due to the randomness of OD travel demand between some regions. ARIMA and the historical averaging method (HA) are simple but less effective. The deep learning method ST-ResNet, which considers the trend, period, proximity, and other temporally more relevant historical data, and deposits a residual network to enhance the network depth, improves the prediction effect. In contrast, as an improved version of ST-ResNet [11], MDL [15] model designs node network and edge network branches to extract edge flow (OD) and node flow (inflow and outflow) features respectively, and uses a multi-task learning approach to predict both flows. But its performance is inferior to that of the single-task ST-ResNet. MDL and ST-ResNet use three segments of historical data as input. GEML [1] is a GNN-based OD matrix prediction model that uses a grid embedding approach and graph convolution to aggregate information about the semantic and geographic neighbors of OD pairs to pre-weight the importance of neighbor nodes in the OD matrices, and adapts LSTM to capture temporal correlations. But the model neither considers the matrix prediction in terms of dynamic and static contextual information of the starting and ending points, nor does it explicitly distinguish the direction of the edges of the adjacency matrix.

Table 2. Comparison Results of TaxiNYC and TaxiCD

Models	RMSE(NYC)	MAE(NYC)	RMSE(CD)	MAE(CD)
HA	1.085	0.163	0.440	0.114
ARIMA	4.027	2.713	0.813	0.328
ST-ResNet	0.647	0.195	0.325	0.146
MDL	1.081	0.993	0.413	0.319
GEML	0.865	0.156	0.332	0.090
ODCRN(ours)	**0.622**	**0.153**	**0.322**	**0.116**

Among all models, ODCRN has the lowest error on the OD matrix prediction task. The prediction performance is improved because ODCRN uses the time-varying OD matrix as the adjacency matrix, the values of the OD matrix as the weights of edges, and considers both forward and backward-directed edges to aggregate neighbor node features more effectively. In addition, ODCRN incorporates both static and dynamic traffic context information of each region, which enables the traffic conditions in the region to be captured to further improve the prediction.

4.5   Comparison of Model Complexity and Speed

We compare the number of parameters (complexity), training speed and inference speed (testing speed) of ODCRN and several deep learning-based baseline

models, where model complexity and speed are important reference metrics for model deployment. In the OD matrix prediction task, we define the OD matrix of the first 8 time slots with the OD matrix of the next 1 time slot representing the labeled data as one sample. Define the speed as the number of samples that can be processed per unit of time (samples/s).

Table 3. Comparison of the number of Parameters on the Model and Speed

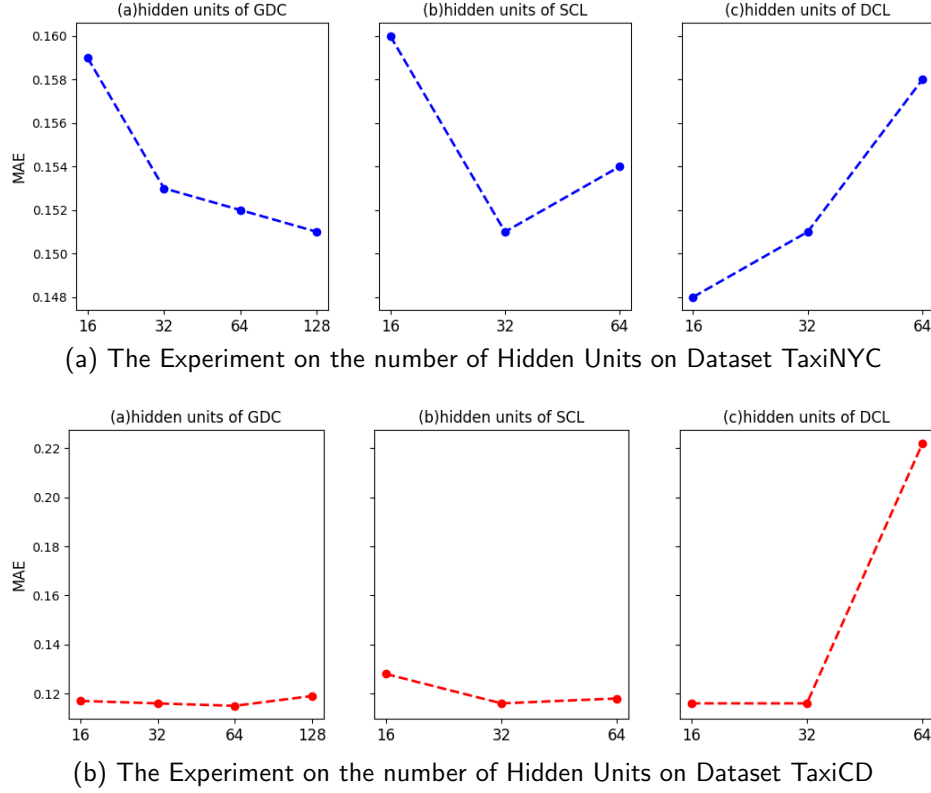| Model	The Number of Trainable Parameters	Training Speed	Reasoning Speed
MDL	7,694,106	81	238
ST-ResNet	7,202,579	93	323
GEML	395,264	53	138
ODCRN(ours)	108,640	108	324

As shown in Table 3, the MDL model uses multi-task learning to extract both regional traffic flow features and OD matrix features, and each task uses three deep convolutional networks with residual connections to extract spatial-temporal features from three highly correlated historical data, respectively, with the highest complexity.ST-ResNet only performs feature extraction on the OD matrix, and the single task makes the model parametric number and speed better.

Our ODCRN model, compared with the GEML model, is an OD matrix prediction model with an LSTM networkwhich has a much smaller number of parameters. For inputs at different time steps, the parameters in the LSTM units are shared, greatly reducing the number of parameters. In contrast, since the ODCRN model integrates both GRU and GCN, the number of parameters in the whole model is mainly the convolutional kernel parameters in the bi-direction diffusion graph convolution in the GRU units, which further reduces the number of parameters compared with the serial structure of GCN and LSTM in GEML. The design of the recurrent unit in ODCRN can capture both temporal and spatial correlations, which improves the training and inference speed while reducing the parameters.

4.6   Hyperparameter Experiments

In this section, we adjust the hyperparameters in the ODCRN model and compare the prediction errors of the model under different parameters. This experiment adjusts the hidden status dimension of the bi-directional diffusion graph convolution in GRU, the hidden state dimension of the static traffic context learning network DNN, and the hidden state dimension of the dynamic traffic context learning network GDC. The experiments are conducted by using the control variates, where one of the parameters is adjusted to observe the prediction accuracy of the model while the other parameters remain unchanged.

In Figure 8a(b) and Figure 8b(b), a hidden state of 32 in the static context network provides a better representation of spatially static information, while smaller hidden state dimensions (16 dimensions) and larger hidden state dimensions (128 dimensions) suffer from underfitting and overfitting problems. Finally, the dynamic traffic context, i.e. the GDC with input features of regional inflow and outflow traffic (2 dimensions), was adjusted for different hidden states. As shown in Figure 8a(c) and Figure 8b(c), the model over-fits as the dimensionality of the GDC hidden layer increases, indicating that the dynamic traffic context features can be better extracted by using GDCs with low-dimensional (16-dimensional) hidden layers at lower input feature dimensions.

(a) The Experiment on the number of Hidden Units on Dataset TaxiNYC

(b) The Experiment on the number of Hidden Units on Dataset TaxiCD

Fig. 8. The Influence of the Number of Hidden Units on Model Performance

4.7   Ablation Experiment

We conduct extensive ablation studies to quantify the performance benefit of each component in our model. Four variants of the ODCRN-based model are constructed and tested as follows:

(1) W/O Static Context Learner: without inputting the static traffic context information, i.e., Node2vec node embedding features.

(2) W/O Dynamic Context Learner: without inputting the dynamic traffic context information, i.e., The inflow and outflow for each region at each moment.

(3) W/O Forward GDC: removing the graph aggregation from the central node, the forward adjacency matrix is an $N \times N (N = m \times n)$ matrix from Origin to Destination.

(4) W/O Backward GDC: removing the graph aggregation to the central node, the backward adjacency matrix is an $N \times N (N = m \times n)$ matrix from Destination to Origin.

Table 4. Results of Ablation Experiments on Dataset TaxiNYC and TaxiCD

Model Variant	RMSE(NYC)	MAE(NYC)	RMSE(CD)	MAE(CD)
w/o static context learner	0.628	0.157	0.322	0.118
w/o dynamic context learner	0.687	0.204	0.327	0.123
w/o forward GDC	0.663	0.161	0.328	0.121
w/o backward GDC	0.631	0.159	0.322	0.119
ODCRN	0.622	0.153	0.322	0.116

The results of the model ablation experiments are shown in Table 4. The ODCRN model has the best prediction with all components retained, and the effectiveness of each component will be described in turn as follow.

Static Context Learning Network When the static context learning network is removed, RMSE and MAE increase compared to ODCRN, which indicates that adding Node2vec node embedding features helps to improve the performance of the model. Since there is no POI information of the regions, we count the starting region, the ending region, and the number of orders as the connected edges between regions (nodes) from one month of taxi orders, so that the constructed graph structure can reflect the community structure, and other characteristics of a region as a popular starting or ending region has multiple connected edges on the graph. Node2vec trains the node embedding features to allow the OD matrix prediction model to learn the prior knowledge of nodes which helps improve the prediction.

Dynamic Context Learning Network Regions with high inflow traffic tend to be set as the endpoint in OD travel demand, while regions with high outflow traffic tend to be set as the starting point in OD travel demand. Using the dynamic traffic contexts of regions as the dynamic characteristics of nodes can improve the OD matrix prediction.

Forward and Backward Diffusion Graph Convolution Due to the regularity of commuting activities, the OD travel demand has different directions at different moments, and departure and arrival are correlated. Therefore, using

the OD matrix with directions (forward) and its transposed DO matrix (backward) to construct bidirectional edges from the target region to other regions, and from other regions to the target region, the graph convolution learns more effective node representations than one-way edges and is aggregated with more information about neighbor nodes.

4.8   Comparison Experiment of Predicted Value and Actual Value

For a more intuitive view of the model predictions, we selected Region 113 and Region 198 in the TaxiNYC dataset and plotted the predicted versus actual value curves of OD travel demand between the two regions for a total of 10 days from April 21, 2015 to April 30, 2015, respectively.

(a) from region 113 to region 198   (b) from region 198 to region 113
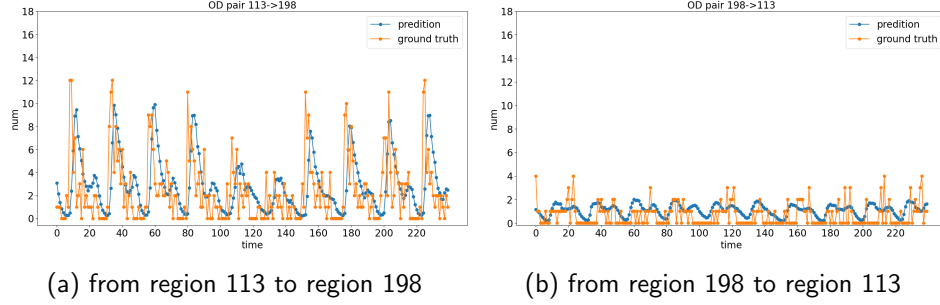
Fig. 9. Comparison between Predicted and Actual value between regions

From Figure 9, we can see that the true values of the OD curves change irregularly with time, which shows an overall cyclical nature and a more obvious morning and evening peak. The trend of the curve and the range of values are different, and the demand for taxis from *region 113* to *region 198* is more than the demand for taxis from *region 198* to *region 113*, which indicates that the travel demand between regions is asymmetric, and the forward OD matrix and backward OD matrix we constructed aggregates the characteristics of neighbor nodes with OD semantic relationship through a directed and weighted adjacency matrix. The ODCRN fits the real data well and is generally consistent with the real values in terms of trend and numerical magnitude. Secondly, we can also find from Figures 9 that the OD curves show dramatic local fluctuations due to the dynamic changes in taxi demand between regions. And this acyclic, randomly changing curve increases the difficulty of prediction, so the model fails to give a more accurate prediction in detail.

Figure 10 shows the predicted values of the OD matrix ($N \times N, N = 256$) compared to the true values at three different moments on April 30, 2015 at 8 am, 2 pm and 10 pm. It can be seen that the smaller number suburban regions keep the values of the corresponding locations of the OD matrix silent due to the
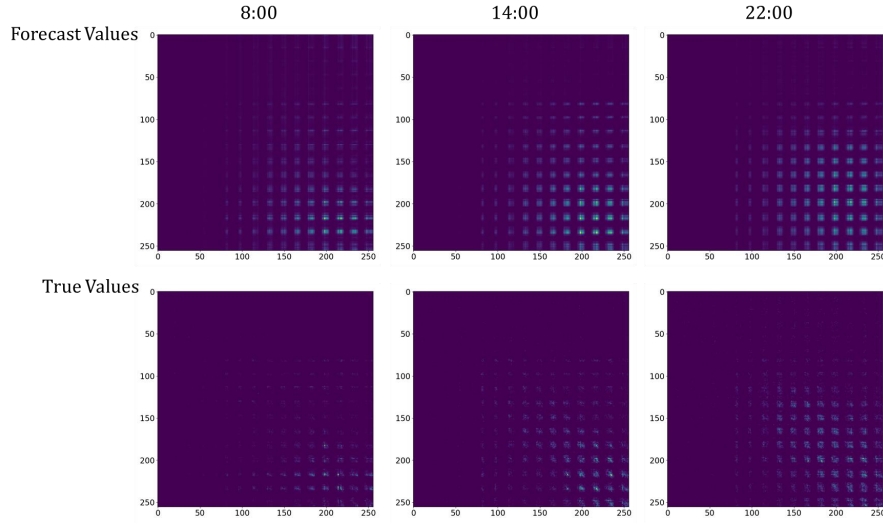
Fig. 10. Comparison of the Predicted Value and Actual Value of OD Matrix

sparse number of taking taxis. The OD matrix predicted by our model is close to the real OD matrix in terms of spatial distribution and temporal variation of ride-hirings, and can better predict the OD travel demand among urban regions.

5   Limitation Analysis

Although the method proposed in this paper is better than the existing traffic prediction model in terms of prediction effect, there still exist some aspects which can be further improved.

Prediction of non-stationary and non-strictly periodic traffic time series Due to the uneven development of urban regions, differences in travel demand, or uneven distribution of data collection devices, there are non-stationary and non-periodic trends in the time series of some nodes, and the prediction of such time series is more difficult than that of regular and periodically varying time series. So, a more robust and better prediction model for the input time series is needed.

Traffic flow prediction under abnormal conditions By visualizing the prediction results, we find that the model has difficulty in achieving accurate prediction for traffic flow surges or sudden decreases caused by peak hours, holidays, extreme weather, social events, traffic accidents, etc. We consider increasing the amount of training data, which can add external features containing the

above events, or time series anomaly detection methods to achieve more accurate prediction.

6   Conclusion

In this paper we propose ODCRN, a novel model that integrates traffic context and spatial-temporal information of OD matrices. We take advantage of the bi-directional semantic information of each travel demand's semantic neighbors to capture both the inflow and outflow of one region. Then we construct traffic context information by static and dynamic traffic contexts to coordinate with OD matrices in the prediction task. We conduct extensive experiments in two datasets and the results demonstrate that our method outperforms baseline methods in terms of prediction accuracy and model complexity.

In future work, we will explore predicting traffic spikes or dips caused by rush hours in the day, holidays, extreme weathers and social events more accurately. Also, we will consider integrating more traffic features into the model elegantly, which requires more experiments on model design and data mining.

References

1. Yuandong Wang, Hongzhi Yin, Hongxu Chen, Tianyu Wo, Jie Xu, and Kai Zheng. 2019. Origin-Destination Matrix Prediction via Graph Convolution: a New Perspective of Passenger Demand Modeling. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '19). Association for Computing Machinery, New York, NY, USA, 1227–1235. https://doi.org/10.1145/3292500.3330877
2. O. Tamin and L. Willumsen, "Transport demand model estimation from traffic counts," Transportation, vol. 16, no. 1, pp. 3–26, 1989.
3. E. Cascetta and S. Nguyen, "A unified framework for estimating or updating origin/destination matrices from traffic counts," Transportation Research Part B: Methodological, vol. 22, no. 6, pp. 437–455, 1988.
4. X. Zhou and H. S. Mahmassani, "Dynamic origin-destination demand estimation using automatic vehicle identification data," TITS, vol. 7, no. 1, pp. 105–114, 2006
5. O. Z. Tamin, H. Hidayat, and A. K. Indriastuti, "The development of maximum-entropy (me) and bayesian-inference (bi) estimation methods for calibrating transport demand models based on link volume information," in EASTS, vol. 4, 2003, pp. 630–647.
6. M. L. Hazelton, "Some comments on origin–destination matrix estimation," Transportation Research Part A: Policy and Practice, vol. 37, no. 10, pp. 811–822, 2003
7. X. Zhou, X. Qin, and H. Mahmassani, "Dynamic origin-destination demand estimation with multiday link traffic counts for planning applications," Transportation Research Record: Journal of the Transportation Research Board, no. 1831, pp. 30–38, 2003.

8. L. Liu, Z. Qiu, G. Li, Q. Wang, W. Ouyang and L. Lin, "Contextualized Spatial–Temporal Network for Taxi Origin-Destination Demand Prediction," in IEEE Transactions on Intelligent Transportation Systems, vol. 20, no. 10, pp. 3875-3887, Oct. 2019, doi: 10.1109/TITS.2019.2915525.

9. F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner and G. Monfardini, "The Graph Neural Network Model," in IEEE Transactions on Neural Networks, vol. 20, no. 1, pp. 61-80, Jan. 2009, doi: 10.1109/TNN.2008.2005605.

10. Xia, Jun & Zhu, Yanqiao & Du, Yuanqi & Li, Stan. (2022). A Survey of Pretraining on Graphs: Taxonomy, Methods, and Applications.

11. Zhang, J., Zheng, Y. and Qi, D. 2017. Deep Spatio-Temporal Residual Networks for Citywide Crowd Flows Prediction. Proceedings of the AAAI Conference on Artificial Intelligence. 31, 1 (Feb. 2017). DOI:https://doi.org/10.1609/aaai.v31i1.10735.

12. Z. Li, L. Li, Y. Peng and X. Tao, "A Two-Stream Graph Convolutional Neural Network for Dynamic Traffic Flow Forecasting," 2020 IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI), Baltimore, MD, USA, 2020, pp. 355-362, doi: 10.1109/ICTAI50040.2020.00063.

13. Li, Yaguang, et al. "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting." arXiv preprint arXiv:1707.01926 (2017).

14. Ke, Jintao, et al. "Predicting origin-destination ride-sourcing demand with a spatial-temporal encoder-decoder residual multi-graph convolutional network." Transportation Research Part C: Emerging Technologies 122 (2021): 102858.

15. Zhang, Junbo, et al. "Flow prediction in spatial-temporal networks based on multitask deep learning." IEEE Transactions on Knowledge and Data Engineering 32.3 (2019): 468-478.

16. Wang, Senzhang, et al. "Multi-task adversarial spatial-temporal networks for crowd flow prediction." Proceedings of the 29th ACM international conference on information & knowledge management. 2020.

17. Zhang, Kunpeng, et al. "A deep learning based multitask model for network-wide traffic speed prediction." Neurocomputing 396 (2020): 438-450.

18. Ke, Jintao, et al. "Joint predictions of multi-modal ride-hailing demands: A deep multi-task multi-graph learning-based approach." Transportation Research Part C: Emerging Technologies 127 (2021): 103063.

19. Liu, Lingbo, et al. "Physical-virtual collaboration modeling for intra-and interstation metro ridership prediction." IEEE Transactions on Intelligent Transportation Systems 23.4 (2020): 3377-3391.

20. Grover, Aditya, and Jure Leskovec. "node2vec: Scalable feature learning for networks." Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining. 2016.

21. Cheng, Yan, Xiafei Ye, and Lifeng Zhou. "Forecasting the Peak-Period Station-to-Station Origin–Destination Matrix in Urban Rail Transit System: Case Study of Chongqing, China." Transportation Research Board 97th Annual Meeting. National Academy of Sciences, 2018.

22. Yuan, Quan, et al. "Time-aware point-of-interest recommendation." Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval. 2013.