Origin-Destination Matrix Prediction via Hexagon-based Generated Graph

Yixuan Yang, Shiyao Zhang, *Member, IEEE*, Chenhan Zhang, and James J.Q. Yu, *Senior Member, IEEE*

***Abstract*— With the rapid development of online ride-hailing services, people have increasingly relied on platforms providing these services to travel. The corresponding companies need to accurately obtain passengers' travel demand to allocate orders and drivers to regions. Therefore, traffic demand prediction is a critical problem of Intelligent Transportation Systems (ITS). Origin-Destination Matrix Prediction (ODMP) is a challenging extension of traffic demand prediction that needs to consider the temporal and spatial dependence of traffic data and predict the relationship between origin and destination of passengers' demand. In this paper, we proposed a method to convert order paths of passenger demand to the hexagon-based path graph. The path graph shows the origin and the destination of the paths of a period. Specifically, considering that traffic flows are time-varying, we generate different hexagon-based path graphs for different time periods. Then, we propose a Hexagon-based Dynamic-Graph Convolutional Network (Hex D-GCN) to make the GCN suitable for dynamic graphs, in which graph connections are different in time series. Furthermore, We evaluate our model on the Didi Chuxing KDD CUP 2020 dataset and get the state-of-art performance. It is shown that our method combines the spatial correlation and temporal correlation well and also captures the passenger's demand pattern.**

I. INTRODUCTION

In the past few years, with the increase in demand for taxi travel and information communication technology development, emerging online ride-hailing services have become increasingly convenient for urban travel. Online car-hailing companies, such as Didi Chuxing and Uber, greatly satisfy people's travel needs. In this context, to better dispatch online ride-hailing drivers and understand passengers' travel needs, using historical data for order demand forecasting is a beneficial method for these companies. The order demand analysis can lead to proper resource allocations (i.e., avoid unnecessary empty drives) and find the optimal paths to maximize profit. The distribution of origin and destination can be obtained from the historical order demand data in the transportation field. The main method used to describe a traveling demand is the origin-destination (OD) matrix, representing the travel demand from point A to point B in a specific time period. Understanding the origin and destination of passenger travel can well describe the distribution of travel demand, which represents the traffic flow patterns in the city.

The origin-destination matrix can well reflect the direct relationship between the origin and destination and further explain passengers' specific needs in different areas. Based on the ride-hailing company's massive data, the origin-destination matrix is becoming predictable. In this paper, we apply the and periodic historical data of passenger's demand pattern to predict the passenger demand at a particular time in the future, which is defined as Origin-Destination Matrix Prediction (ODMP) problem.

Traditional traffic forecasting methods focus more on the temporal features of OD data, e.g., autoregressive integrated moving average (ARIMA) [1], support vector regression machine (SVM) [2], and the Bayesian model [3]. But these models lack attention on the spatial semantic relationship feature of the map data. Thus, the researchers introduced the convolutional neural networks (CNN) to the traffic forecasting tasks to consider the spatial feature [4]. However, due to convolution computation's structural characteristics, CNN is best suitable for processing data in Euclidean space, such as images and videos. More methods came out with the development of related deep learning models, such as graph convolutional networks (GCN). All the traffic prediction tasks are based on the geographic information map [5]. This kind of map is Euclidean data, but the traffic flow and the real-world paths are the natural non-Euclidean data, which are suitable for being processed by GCN. Moreover, time-series data is also natural in the real world. To handle this kind of data, researchers adopted recurrent neural networks (RNN). Traffic data have a time-series relationship, i.e., the previous state determines the current traffic state. RNN can extract the traffic data temporal characteristics effectively.

In terms of ODMP, researchers need to consider the traffic order demand data's both temporal and spatial correlations. Some researchers partition the map into square areas [6], and other researchers partition the map according to the administrative areas [7] to mark the origin and destination of the orders' paths to show the spatial feature. And splitting the time-series data into several time slots to show the temporal feature. However, as traditional graph neural network techniques require the graph nodes' spatial characteristics to be fixed, they cannot be directly implemented on dynamic graph structure data; however, the traffic flow changes in real-time and is time-correlated. For ODMP, the number of orders and path distribution in each time state is different, and the orders' paths have different origins and destinations in each period. When the paths abstract to the graph data

This work is supported by the Stable Support Plan Program of Shenzhen Natural Science Fund No. 20200925155105002, by the General Program of Guangdong Basic and Applied Basic Research Foundation No. 2019A1515011032, and by the Guangdong Provincial Key Laboratory (Grant No. 2020B121201001). The authors are with the Guangdong Provincial Key Laboratory of Brain-inspired Intelligent Computation, Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen 518055, China. James J.Q. Yu is the corresponding author.

structure, the nodes' connections in the graph of different time periods need to be updated. Thus, dynamic input of traffic demand graph data is vital for the prediction.

To address the above problems, we propose a method to partition the map into hexagonal regions and then generate the graph structure data according to the orders' paths in each time period. The hexagon is one of three regular polygons that can tile the Euclidean space. Compared with triangles and squares, hexagons have an unambiguous neighborhood definition, smaller edge-to-area ratio, and multiple isotropic properties [8]. Each single regular hexagon has six equivalent regular hexagons, so this representation has better connectivity than others, potentially revealing more data correlation. In this paper, each hexagonal grid represents the vertex of the hexagon-based graph, which are generated by the orders' paths in a time slot. In this way, the graph data and temporal relationship can be better used for modeling. Inspired by T-GCN [9], we propose the Hexagon-based Dynamic-Graph Convolutional Network (Hex D-GCN) model for ODMP problem, which uses the hexagon-based dynamic graph to update the adjacent to improve the prediction performance of the model.

Our main contribution can be summarized as follows:

- Unlike existing methods for OD prediction, we divide the map space into hexagonal grids instead of square grids. Based on the hexagonal grid map, we convert the traffic demand orders' path of a time period to the hexagon-based graph path. Additionally, we use the graph path to generate the adjacent matrix, the in-degree, out-degree feature matrix, and the origin-destination matrix as the input.
- We propose a Dynamic-GCN model to address the ODMP problem. For the ODMP problem, different time slots have different hexagonal grids graphs, which means that the graph is in changing real-time. We refer to the GraphSAGE [10] model to improve the GCN effects for the real-time changing graph.
- The Hex D-GCN achieve great performance on the Didi Chuxing KDD CUP 2020 dataset. In addition, this method has good universality in analyzing traffic flow patterns and capturing the passenger's demand pattern.

The rest of our paper is organized as follows. Section II reviews the literature on the traffic prediction model and ODMP problem. Section III defines the abstract map method and give the definition of the origin-destination matrix prediction problem. Section IV presents the network framework from spatial dependence perspective and temporal dependence perspective. Section V discusses the results of our approach and compares them with other methods. Finally, concluding remarks are given in Section VI.

II. Related Work

A. Traffic Demand Prediction

The prediction of the passenger demand and traffic conditions from diverse areas and periods can contribute to taxi companies' operational strategy design. In previous OD prediction literature, the researchers pay more attention to the traffic demand prediction. Traditional time-series models are adopted to solve this problem, e.g., ARIMA [1], Kalman filter [11], and their variants. However, these models often lack the ability to capture non-linear temporal and spatial correlations in data for prediction.

With the improvement of computing efficiency, machine learning methods are getting increasingly prevalent in demand prediction tasks. RNN [12] and long short-term memory model (LSTM) [13] are good at capturing the temporal feature of the data. This kind of model looks for the time-series relationship of the data to make predictions but ignores the spatial characteristics. The convolutional neural network (CNN) can capture the data's spatial feature and combine it with LSTM or RNN to capture the temporal feature [8], [14]. For instance, [8] used the hexagonal-based CNN to capture the spatial feature and proposed the SRCN to predict the short-term and long-term traffic flow. These models always have too many model parameters and not concise enough.

For the OD prediction task, how to divide the map to generate input data is a critical problem. Some researchers divide the space into square grids [7]. This segmentation is natural for CNN to perform convolution operations. Other researchers use the administrative planning areas [6] for the OD prediction task. This segmentation has good semantic meaning, but it is not in the regular areas, which is not suitable for the OD matrix. The administrative planning areas are always the big areas that are not proper to make fine-grained predictions. As for the hexagon grids, some researchers proposed the hexagon convolution method [8], but this method is not universal to the normal data. In the ODMP problem, the paths in every period can compose of the natural graph structure data. We proposed a universal method to abstract the paths of a period to the hexagon-based path graph in Section III.

B. Graph Convolutional Network

Data are often sampled in non-Euclidean spaces (e.g., graphs) [15]. Compared to the CNN that captures the Euclidean data feature, [16] proposed the GCN to deal with graph data such as social networks and protein-protein connection networks. The traffic network is the natural graph data structure, and many researchers introduce GCN to the traffic prediction task [4], [9], [17]. For example, STGCN [17] models the traffic network as the graph, and adopts GCN and RNN to capture the latent spatial and temporal dependence respectively for traffic forecasting. This kind of model always fixes the adjacent matrix of GCN without considering the dynamic change of the paths; however, the graph dynamic change of the paths brings the update of the adjacent matrix. In this paper, we introduce the dynamic adjacent matrix to GCN to improve the performance of the ODMP.

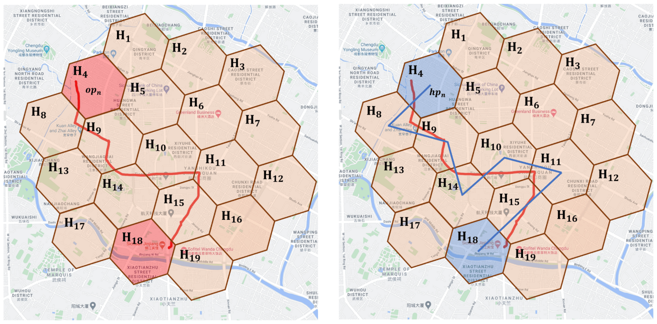
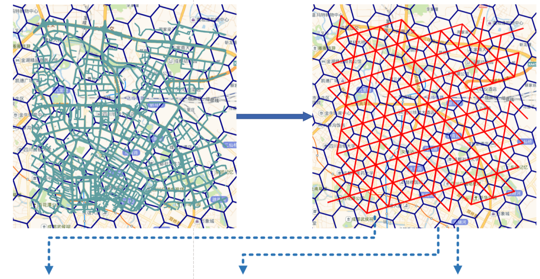


Fig. 1. The left figure shows the hexagon-based map and one order path (red path). The right figure shows the hexagon-based polyline order path (blue path).



$$A_t = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{12} & \cdots & a_{1n} \\ \vdots & \vdots & \cdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} X_t = \begin{bmatrix} x_{11} & x_{12} \\ x_{21} & x_{12} \\ \vdots & \vdots \\ x_{n1} & x_{n2} \end{bmatrix} OD_t = \begin{bmatrix} od_{11} & od_{12} & \cdots & od_{1n} \\ od_{21} & od_{12} & \cdots & od_{1n} \\ \vdots & \vdots & \cdots & \vdots \\ od_{n1} & od_{n2} & \cdots & od_{nn} \end{bmatrix}$$

Fig. 2. The process of converting the orders' path to the graph path based on the hexagonal map in one time slot $Slot_t$. The adjacent matrix A_t, the in-degree, out-degree feature matrix X_t, and the Origin-Destination matrix OD_t can be abstracted from the graph path map.

III. PRELIMINARIES

A. Abstract Map

We first discuss the abstraction of the 2D Euclidean space map to non-Euclidean graph structure data. Previous researchers primarily partition the city map into multiple square cells. This method is easy for CNN to capture the spatial correlation, and grids can be easily converted to the input matrix of CNN just as typical Euclidean space data. Despite this, we partition the city map into regular hexagons. Hexagon segmentation has the tessellation property of planes. Compared with the square grids segmentation, the hexagon segmentation has an unambiguous neighborhood definition, smaller edge-to-area ratio, and multiple isotropic properties [8]. This representation has better connectivity than squares and triangles because every single regular hexagon has six equivalent neighbor regular hexagons. So hexagon segmentation can easily capture the traffic flow feature among different hexagon areas. Compared with the circle space segmentation, hexagons can split the space without overlapping area while reducing the computational complexity.

In our method, we split the city map domain into the hexagon grids $\mathcal{H} = \{H_1, H_2, \cdots, H_n\}$. The order path $\mathcal{OP} = \{op_1, op_2, \cdots, op_k\}$ can be converted into the polyline path based on the hexagonal map $\mathcal{HP} = \{hp_1, hp_2, \cdots, hp_k\}$. This kind of path go through hexagon grid from $H_i \rightarrow H_j$. Fig. 1. shows part of a origin path of Chengdu, China, and the hexagon-based path of an order. A part of the op_i from $H_4 \rightarrow H_8$ converted into the connecting-line of the center points of two regular hexagons H_4 and H_8.

We abstract the normal traffic path of the demand map by the above method. Hexagon-based polyline paths in a time period can be generated as a graph of non-Euclidean paths, which will be introduced in Section IV-A.

B. Origin-Destination Matrix Prediction Problem

With the development of the riding source of the online platform, researchers can have enough data to predict the origin-destination distribution of passengers' demands by ODMP. In this part, we give the definition of ODMP.

Definition 3.1. ***Time Slot***: We divide a time period into multiple time fragments evenly, and each fragment becomes one time slot, represented as $\{Slot_1, Slot_2, \cdots, Slot_t\}$. Meanwhile, there is no overlap between neighboring time slots.

Definition 3.2. ***Origin-Destination Matrix***: For every order, there is a origin hexagon cell and a destination hexagon cell of the path. In each time slot, the total number of orders from $H_i \rightarrow H_j$ can be denoted as m_{ij}. For the whole city hexagonal-based map, the OD matrix can be represented as $M \in N^{H \times H}$, where $m_{ij} \in M$ denotes the number of demands from H_i to H_j.

Definition 3.3. ***Origin-Destination Matrix Prediction*** : For t time slots, we input a sequence of prior observed OD matrix $M_1, M_2, \cdots, M_n$ and a prior information feature X, e.g., the input-output degree of one hexagon grid, to predict the OD matrix M_{t+1} in $Slot_{t+1}$.

IV. METHODOLOGY

A. Spatial Dependence Modeling

In traffic prediction problems, the spatial relationship between the road and area is the key that needs to be considered. In Section III-A, we partition the city map into hexagon grids and convert the order paths into polygonal paths based on the hexagon-based map. There are n order paths $\{op_1, op_2, \cdots, op_n\}$ in time slot t, and the multiple paths can compose a complete paths map G_t. Every path in G_t can be converted to the hexagon-based polygonal path and then compose the new hexagon-based graph HG_t in time slot t.

The hexagon-based graph HG_t of time t is the natural graph structure. Traditional CNN is efficient in obtaining the features in Euclidean space data such as images. But for non-Euclidean space data, GCN is more suitable to capture the feature. It is easier to get the relationship between nodes in the graph or the regional relevance of areas of traffic prediction tasks.

In the traffic prediction graph-based model, researchers need to capture the temporal and spatial features of the traffic data. Inspired by T-GCN [9], our model has two parts, the Dynamic-GCN part for capturing the spatial feature and the

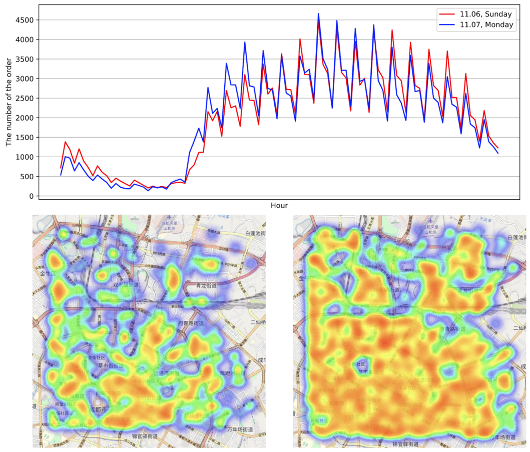


Fig. 3. The distribution of order quantity in time (above) shows that the distribution of orders on weekends and weekdays is different. The spatial distribution of order is shown in heatmaps (below). The below-left heatmap is the order distribution is from 0:00 to 1:00 at midnight on November 7th, and the below-right heatmap is from 16:00 to 17:00 on November 7th.

GRU part for capturing the temporal feature. The normalized adjacent matrix $\tilde{A}$ and the nodes' feature matrix X are the input of the GCN model. Specifically, the $\tilde{A}$ is the adjacent matrix calculated by $\tilde{A} = \hat{D}^{-\frac{1}{2}}(A+I)\hat{D}^{-\frac{1}{2}}$ [16], where the $\hat{D}$ is the degree matrix of $A+I$, $\hat{D} = \sum_{ij}(A+I)_{ij}$. Each layer of the GCN can be expressed as:

$$f(X^{(l)}, \tilde{A}) = \sigma(\tilde{A}X^{(l)}W^{(l)}) \tag{1}$$

In the previous GCN-based approaches of the traffic demand prediction task, the graph of the paths is based on the fixed areas. The nodes' connection in the graph is regarded as fixed, and thus the adjacent matrix is fixed. For our graph-generated method, in every time slot $Slot_t$, we can abstract the single adjacent matrix A_t, which nodes' connection update by changing the orders' paths. That means our input graph is dynamic updating for the model. Besides the matrix A_t, the feature X_t of the graph is defined as the in-degree and the out-degree for one node. Besides, we can get the OD matrix M_t in each time slot. The process illustrates in Fig. 2.

GraphSAGE [10] is a method to process large graphs data. The fundamental idea of the method is to aggregate the neighbor nodes' information to the center nodes. In this method, the graph can dynamic updates such as adding nodes or reducing nodes or change the connection relationship between nodes. In GCN approaches, the element-wise multiplication with one row i of the A_t and one column of the X_t, to get the aggregation in the dimension j of the node i. This operation can be regarded as the aggregation operation in GraphSAGE. As in [10], the researchers proposed that the mean-aggregator method is similar to the GCN. For our method, the nodes' update means the connection between nodes in the graph needs to be updated, and the number of the nodes in the graph does not change. Thus, we only need to update the adjacent matrix A_t with the time change.

To predict the OD matrix, we introduce the transition

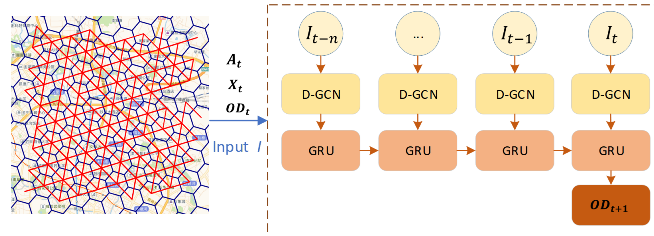


Fig. 4. Hexagon-based Dynamic-GCN model.

matrix W_{tr} with orthogonal initialization as the parameter in the first layer of the model (Eq. 2a). The complete two-layer Dynamic-GCN is defined as Eq. 2b,

$$f_1(A_t, X_t, W_{tr}) = \sigma(\hat{D}^{-\frac{1}{2}}(A_t+I)\hat{D}^{-\frac{1}{2}}X_tW_{tr}), \tag{2a}$$

$$f_2(\tilde{A}, f_1) = \sigma(\tilde{A}f_1W_0), \tag{2b}$$

where σ is the linear function and $f_2(\tilde{A}, f_1)$ is the output result of the current GCN Block.

B. Temporal Dependence Modeling

Traffic prediction has a strong time dependence. With the progress of time, taxi orders OD also has periodic changes. The distribution of taxi OD varies at different times of weekends and weekdays. Also, the OD distribution at midnight and in the rush hour is different (Fig. 3). How to capture the temporal feature is the key problem in traffic prediction. RNN [18] is a widely recognized deep learning model to deal with the sequence problem. However, too many parameters may cause the gradient explosion issue, and RNN cannot store long-term memory information well, and long-term memory will be overridden by short-term memory. LSTM [19] and GRU [20] can circumvent these problems. LSTM uses a gated mechanism to avoid gradient explosion and uses a cell state to store long-term memory. Then, it cooperates with the gate mechanism to filter information to achieve control of long-term memory information. Compared with LSTM, GRU has similar performance and has fewer parameters which are easier to train [21] and converge faster. Hence, we followed T-GCN [9] to chose GRU as our model to capture the temporal feature of the OD traffic data.

For ODMP problem, by inputing a time series $\{x_t\}_{t=1}^{T}$ to the GRU cell, GRU encode the $\{x_t\}_{t=1}^{T}$ to the hidden state $\{h_t\}_{t=1}^{T}$ via $h_t = g(x_t, h_{t-1})$, where the $g(\cdot)$ is the non-linear mapping function of GRU. Certain expression can be defined as below equations,

$$r_t = \sigma(W_r[f(A_t, X_t, W_{tr}), h_{t-1}] + b_r) \tag{3}$$

$$u_t = \sigma(W_u[f(A_t, X_t, W_{tr}), h_{t-1}] + b_u) \tag{4}$$

$$c_t = \tanh(W_c[f(A_t, X_t, W_{tr}), (r_t * h_{t-1})] + b_c) \tag{5}$$

$$h_t = u_t * h_{t-1} + (1-u_t) * c_t \tag{6}$$

where r_t and u_t is the reset gate and the update gate, c_t is the new input state. W is the weight, and b is the bias. $f(\cdot)$ is the Dynamic-GCN block output, and h_t is the output of the GRU cell.

Besides the GRU model, the dynamically updating adjacent matrices A_t also have the temporal relevance property. The change of A_t depends on the passengers' demand in different time slots $Slot_t$ as time goes by.

To conclude, we introduce a method to generate the passengers' demand traffic map's dynamic graph based on the hexagon segmentation. We propose a Hexagon-based Dynamic-GCN model to predict the passengers' origin-destination matrix with the combination of spatial and temporal features. The model is illustrated in Fig. 4.

C. Loss function

Our training goal is to minimize the error between the predicted output OD matrix M_{t+1} and the ground truth OD matrix $\hat{M}_{t+1}$ in time slot $Slot_{t+1}$. We apply the same loss function as the T-GCN [9] as Equation 7. The first item is the L2 loss function and the second item is the regularization item as the L2-norm.

$$loss = \left\| M_{t+1} - \hat{M}_{t+1} \right\| + \lambda L_{reg} \tag{7}$$

V. Experiments

A. Dataset Pre-processing and Evaluation Method

In our experiments, we select the Didi Chuxing KDD CUP 2020 dataset (`https://gaia.didichuxing.com`) to evaluate the ODMP task. This dataset comes from the trajectory and order data of the Didi ride-sharing platform in a local area of Chengdu, China. The order data include the empty car transfer rate, order cancellation probability, and hexagonal grid coordinates data. The sampling interval of trajectory points is 2–4s. The period of this dataset is from 1st to 30th in November 2016.

We focus on the trajectory data of each order and hexagonal grid coordinates data in the dataset. All order routes are located within $[30°65294'N, 104°04215'E]$ and $[30°72775'N, 104°12958'E]$. We further remove noisy data such as too-short trajectories whose number of trajectory points is less than 50 and drifting paths whose trajectory points are off the road. We set the time slot interval to 1 hour to split the dataset and using the OD matrix of 5 time slots to predict the next 1 hour OD matrix, which follows [7].

We adopt the root mean square error (RMSE) and symmetric mean absolute percentage error (SMAPE) to show the prediction accuracy:

$$RMSE = \sqrt{\frac{1}{|M_{t+1}| \times N} \sum_{n=1}^{N} \left\| M_{t+1}^{n} - \hat{M}_{t+1}^{n} \right\|} \tag{8}$$

$$SMAPE = \frac{2}{|M_{t+1}| \times N} \sum_{n=1}^{N} \sum_{m \in M_{t+1}^{n}} \frac{m - \hat{m}}{m + \hat{m} + 1} \tag{9}$$

TABLE I

Results on demand prediction of different methods.

Metrics	HA (Baseline)	GCRN	LSTM	GEML	**Ours**
RMSE	2.8382 -	4.1512 -46.3%	12.0027 -322.9%	1.6928 +40.3%	**1.0749** **+62.1%**
SMAPE	0.6199 -	0.3842 +38.0%	0.5121 +17.4%	0.2678 +56.8%	**0.2660** **+57.1%**

B. Experimental Setup

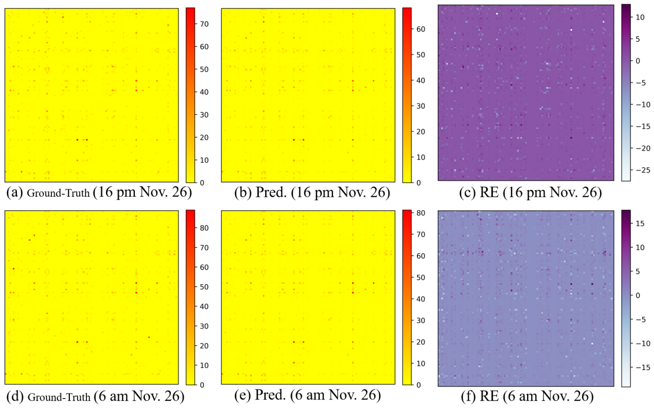
To test the model performance on the Didi dataset, we divide the whole dataset into the training dataset, validation dataset, and test dataset. There are no national holidays in November 2016, so we don't need to exclude specific dates. The data from November 1st to November 20th are used as the training set, and the data from November 21st to November 25th as the validation set, while the last five days as the test set. In ODMP, researchers typically set all the entries smaller than three to zeros in OD matrices to reduce the influence of data noise [14], which can prevent taxis from appearing at irregular locations and times to interfere with the model. We follow this paradigm to pre-process the raw data.

Our model's hyperparameters are set as follows. We use the Adam optimizer to optimize training, and the learning rate is 0.001, the batch size is 5, the training epoch is 1000. Empirically, we use 100 GRU hidden units to train our model. The experiments are executed on the server with Intel(R) Xeon(R) Silver 4210 CPU @ 2.20GHz CPU and 2080Ti GPU. And the reported values of RMSE and SMAPE are the averaged value of five times running results.

C. Experiments Results

1) OD Matrix Prediction Accuracy: In our experiments, the History Average (HA) of data is set as the baseline. Besides, we employ the classical temporal learning model LSTM, a temporal and graph data learning model GCRN [22], and the grid-embedding-based multi-task learning (GEML) [7] which considers the prior information and geographical semantic information for comparison.

From Table I, we can develop the following observations. The HA as the baseline has the stable performance of RMSE and SMAPE, and sometimes even better than the proposed models. We calculated the relative performance improvements (or degradation) of other methods based on the baseline HA. LSTM model only considers the temporal features of the traffic data and does not care about the data's spatial feature. RMSE of LSTM gets the worst performance among the models. Therefore, the model which only cares about the time-series feature is not suitable for the ODMP problem. And for the GCRN model, this model cares about the data's temporal and spatial correlations. This method has better performance on SMAPE than the baseline. GEML is a specific model for the ODMP task. This model divides the map into square grids to do embedding and do multi-task learning to predict the OD matrix. This model considers temporal and spatial features and gets satisfactory performance.



(a) Ground-Truth (16 pm Nov. 26) (b) Pred. (16 pm Nov. 26) (c) RE (16 pm Nov. 26)

(d) Ground-Truth (6 am Nov. 26) (e) Pred. (6 am Nov. 26) (f) RE (6 am Nov. 26)

Fig. 5. The ground truth and the prediction result of different time slots, and the relative prediction error between them.

Our method, Hexagon-based Dynamic-GCN, generates the state-of-the-art performance on the Didi Chuxing KDD CUP 2020 dataset and gets 1.0749 RMSE and 0.2660 SMAPE, which has a massive improvement from previous models.

2) OD Matrix Prediction Visualization: RMSE and SMAPE are the evaluation index of the model used to compare the prediction result of the different models. But they cannot show the prediction result directly. To see our prediction results intuitively, we choose two time slots, 16:00 on November 26th and 6:00 AM on November 26th, to show the predicted OD matrix and ground truth OD matrix. However, since the intensity and peak values of the ground truth and prediction results are concentrated in fixed areas, it isn't easy to compare them directly. So we also illustrate the relative prediction error distribution in Fig. 5. Intuitively, for the relative prediction error in Figs. 5 (c) and (f), the depth of color is different. That means for the different time slots, the prediction errors are different. Also, the OD demands are very imbalanced in different areas and different time periods. Since the prediction's intensity is similar to the ground truth, our model effectively captures the human mobility patterns' feature to make the prediction.

VI. Conclusion

In this paper, we propose a Hexagon-based Dynamic-GCN model to address the Origin-Destination Matrix Prediction (ODMP) problem. The OD matrix can show the traffic demand data's temporal and spatial features and show the passengers' travel patterns. But previous methods cannot consider the temporal and spatial features well. Different from the previous efforts, we divide the map into hexagon grids and propose a novel method to convert the Euclidean data of the paths of order demand graph to the hexagon-based graph data. Furthermore, considering that traffic flow is real-time changing, we improve the proposed GCN-based model to adapt to dynamic traffic graphs. To test the performance of Hexagon-based Dynamic-GCN, we evaluate our model on the Didi Chuxing KDD CUP 2020 dataset. The proposed model develops the state-of-the-art performance of ODMP on the dataset.

References

[1] M. M. Hamed, H. R. Al-Masaeid, and Z. M. B. Said, "Short-term prediction of traffic volume in urban arterials," *Journal of Transportation Engineering*, vol. 121, no. 3, pp. 249–254, 1995.

[2] C.-H. Wu, J.-M. Ho, and D.-T. Lee, "Travel-time prediction with support vector regression," *IEEE Transactions on Intelligent Transportation Systems*, vol. 5, no. 4, pp. 276–281, 2004.

[3] M. J. Maher, "Inferences on trip matrices from observations on link volumes: A bayesian statistical approach," *Transportation Research Part B: Methodological*, vol. 17, no. 6, pp. 435–447, 1983.

[4] J. Zhang, Y. Zheng, and D. Qi, "Deep spatio-temporal residual networks for citywide crowd flows prediction," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 31, no. 1, 2017.

[5] C. Zhang, S. Zhang, Y. J. JQ, and S. Yu, "An enhanced motif graph clustering-based deep learning approach for traffic forecasting," in *GLOBECOM 2020-2020 IEEE Global Communications Conference*. IEEE, 2020, pp. 1–6.

[6] C. Li, L. Bai, W. Liu, L. Yao, and S. T. Waller, "Graph neural network for robust public transit demand prediction," *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[7] Y. Wang, H. Yin, H. Chen, T. Wo, J. Xu, and K. Zheng, "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*, 2019, pp. 1227–1235.

[8] J. Ke, H. Yang, H. Zheng, X. Chen, Y. Jia, P. Gong, and J. Ye, "Hexagon-based convolutional neural network for supply-demand forecasting of ride-sourcing services," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 11, pp. 4160–4173, 2018.

[9] L. Zhao, Y. Song, C. Zhang, Y. Liu, P. Wang, T. Lin, M. Deng, and H. Li, "T-gcn: A temporal graph convolutional network for traffic prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 9, pp. 3848–3858, 2019.

[10] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," *arXiv preprint arXiv:1706.02216*, 2017.

[11] G. Vigos, M. Papageorgiou, and Y. Wang, "Real-time estimation of vehicle-count within signalized links," *Transportation Research Part C: Emerging Technologies*, vol. 16, no. 1, pp. 18–35, 2008.

[12] J. Xu, R. Rahmatizadeh, L. Bölöni, and D. Turgut, "Real-time prediction of taxi demand using recurrent neural networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 8, pp. 2572–2581, 2017.

[13] X. Ma, Z. Tao, Y. Wang, H. Yu, and Y. Wang, "Long short-term memory neural network for traffic speed prediction using remote microwave sensor data," *Transportation Research Part C: Emerging Technologies*, vol. 54, pp. 187–197, 2015.

[14] L. Liu, Z. Qiu, G. Li, Q. Wang, W. Ouyang, and L. Lin, "Contextualized spatial–temporal network for taxi origin-destination demand prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 10, pp. 3875–3887, 2019.

[15] C. Zhang, J. James, and Y. Liu, "Spatial-temporal graph attention networks: A deep learning approach for traffic forecasting," *IEEE Access*, vol. 7, pp. 166 246–166 256, 2019.

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

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

[18] T. Mikolov, M. Karafiát, L. Burget, J. Černockỳ, and S. Khudanpur, "Recurrent neural network based language model," in *Eleventh annual conference of the international speech communication association*, 2010.

[19] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

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

[21] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[22] Z. Cui, K. Henrickson, R. Ke, and Y. Wang, "Traffic graph convolutional recurrent neural network: A deep learning framework for network-scale traffic learning and forecasting," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 11, pp. 4883–4894, 2019.