Transportation Research Part C 122 (2021) 102858

Contents lists available at ScienceDirect

Transportation Research Part C

journal homepage: www.elsevier.com/locate/trc

Predicting origin-destination ride-sourcing demand with a spatio-temporal encoder-decoder residual multi-graph convolutional network

Jintao Ke [d], Xiaoran Qin [a,*], Hai Yang [a], Zhengfei Zheng [a], Zheng Zhu [a], Jieping Ye [b,c]

[a] Department of Civil and Environmental Engineering, The Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong, China
[b] Department of Computational Medicine and Bioinformatics, University of Michigan, Ann Arbor, United States
[c] AI Labs, Didi Chuxing, Beijing, China
[d] Department of Logistics and Maritime Studies, The Hong Kong Polytechnic University, Kowloon, Hong Kong, China

ARTICLE INFO

Keywords:
OD demand prediction
Correlation adjacent matrix
Multi-graph convolutional neural network
Spatio-temporal feature
Deep learning model

ABSTRACT

With the rapid development of mobile-internet technologies, on-demand ride-sourcing services have become increasingly popular and largely reshaped the way people travel. Demand prediction is one of the most fundamental components in supply-demand management systems of ride-sourcing platforms. With an accurate short-term prediction for origin-destination (OD) demand, the platforms make precise and timely decisions on real-time matching, idle vehicle reallocations, and ride-sharing vehicle routing, etc. Compared to the zone-based demand prediction that has been examined in many previous studies, OD-based demand prediction is more challenging. This is mainly due to the complicated spatial and temporal dependencies among the demand of different OD pairs. To overcome this challenge, we propose the *Spatio-Temporal Encoder-Decoder Residual Multi-Graph Convolutional network* (ST-ED-RMGC), a novel deep learning model for predicting ride-sourcing demand of various OD pairs. Firstly, the model constructs OD graphs, which utilize adjacent matrices to characterize the non-Euclidean pair-wise geographical and semantic correlations among different OD pairs. Secondly, based on the constructed graphs, a residual multi-graph convolutional (RMGC) network is designed to encode the contextual-aware spatial dependencies, and a long-short term memory (LSTM) network is used to encode the temporal dependencies, into a dense vector space. Finally, we reuse the RMGC networks to decode the compressed vector back to OD graphs and predict the future OD demand. Through extensive experiments on the for-hire-vehicles datasets in Manhattan, New York City, we show that our proposed deep learning framework outperforms the state-of-arts by a significant margin.

1. Introduction

Ride-sourcing service provided by transportation network companies (TNCs) such as Uber, Lyft, and DiDi, has experienced rapid growth since its emergence in 2009. It is reported that Uber has expanded its business to 700 cities and 24 countries around the world, while DiDi is providing service for over 25 million trips per day in 400 cities in China. With their 24-hour-a-day availability and

* Corresponding author.
E-mail address: xqinad@connect.ust.hk (X. Qin).

https://doi.org/10.1016/j.trc.2020.102858
Received 29 October 2019; Received in revised form 20 October 2020; Accepted 23 October 2020
Available online 3 December 2020

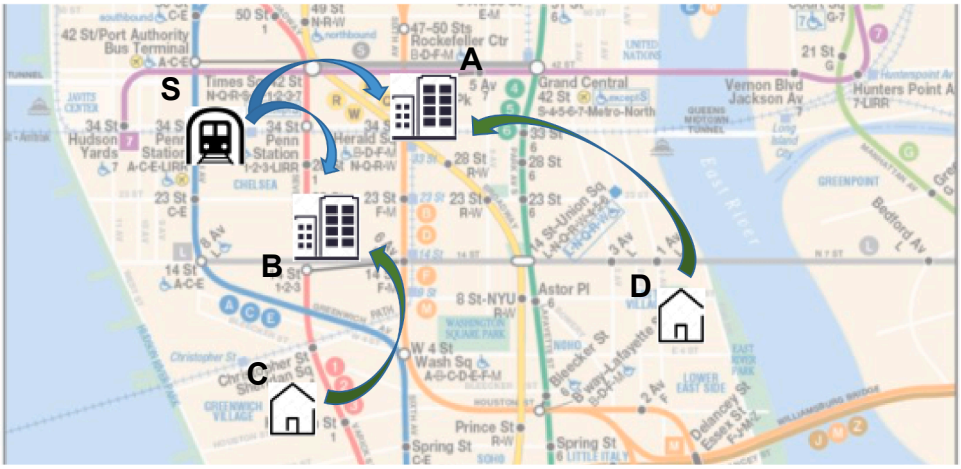


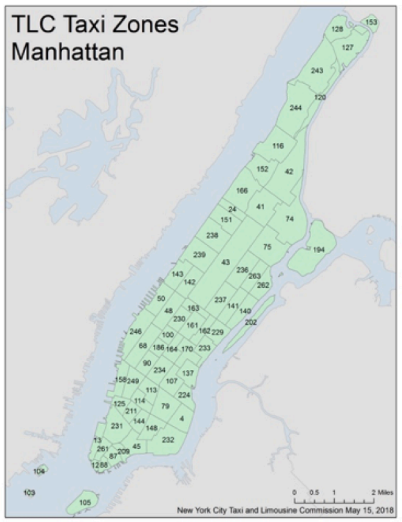
Fig. 1. An example of different correlations among OD pairs.

capacity to serve door-to-door on-demand requests, ride-sourcing service is becoming an important and indispensable component in urban transportation systems. The major challenges in the operational management of ride-sourcing services are how to address supply-demand imbalance across space and time, and how to satisfy as many passenger requests as possible with a limited vehicle fleet size. To address these issues, prior studies have proposed a series of approaches, including surge pricing in which prices are raised to suppresses passenger demand in peak-hours, idle vehicle reallocations in which idle vehicles are moved from regions with excessive supply to regions with excessive demand (Lin et al., 2018), efficient order dispatching strategies (Zha et al., 2018), rush hour supply management (Su and Wang, 2019) and shared ride-sourcing services that allow one vehicle to serve two or more passengers/requests in each ride to improve vehicle usage (Ke et al., 2020a, 2020b; Li et al., 2019; Dong et al., 2018; Liu and Li, 2017; Nourinejad and Roorda, 2016), etc.

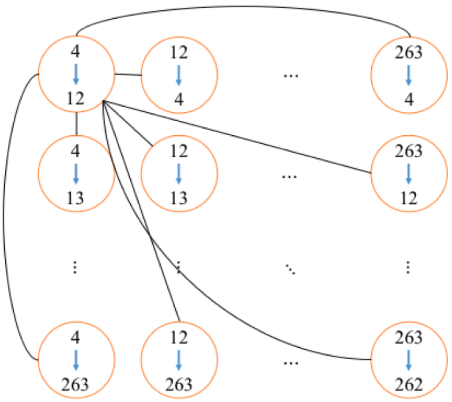
Many of these strategies rely on accurate real-time demand forecasting, especially the real-time origin-destination (OD) demand forecasting that predicts not only the potential demand originating from one region but also their destinations. For example, being aware of many passengers requesting rides from region A to region B, the platform can reallocate idle vehicles to region A in advance, such that passengers will not experience a long waiting time. In particular, OD demand prediction plays a critical role in the operations of shared ride-sourcing services. To attract passengers to use shared ride-sourcing services, the platform usually introduces a discounted fare for these services compared with normal solo ride services; however, passengers who opt for shared rides may experience extra trip time caused by vehicles' detour to pick-up and drop-off other passengers. Generally, as the platform decides to dispatch a vehicle to pick-up the first passenger, it does not know whether the vehicle can easily pick-up a second passenger along the way. This uncertainty makes it difficult for the platform to determine the upfront trip fares and dispatching strategies (whether and when to dispatch an en-route vehicle to pick-up a second passenger). An accurate OD demand forecasting can help the platform alleviate these uncertainties and estimate the probability that an en-route vehicle meets a second passenger, which can further guide the upfront pricing and matching decisions. As the market share of shared ride-sourcing services increases rapidly, for example, Lyft is said to have 50 percent of its rides being shared by 2022 (Schaller, 2018), there is a pressing need for accurate prediction of real-time OD demand.

Most of the existing studies focus on the prediction of passenger demand originating from each region or zone (Ke et al., 2017; Zhang et al., 2017; Yao et al., 2018a; Zhang et al., 2019a,b,c,d). However, only limited efforts are made so far towards predictions of OD passenger demand (Liu et al., 2019; Wang et al., 2019). One possible reason is that OD-based demand forecasting is much more challenging than zone-based demand forecasting. One of the major challenges is how to capture the spatial-temporal dependencies between each two OD pairs. On a complex and irregular network, passenger demand in different OD pairs can be correlated with each other both geographically and semantically. For example, as shown in Fig. 1, the passenger demand originating from a subway station S to a commercial zone A and the passenger demand from the station S to a commercial zone B can be positively correlated during the period such as morning rush hours, since they are both determined by the outflow of the subway station. This indicates that the prediction of demand for a specific OD pair can benefit from the information of OD pairs with nearby origins or destinations. Moreover, an OD pair may have common demand patterns with a geographically distant OD pair, as they share similar functionalities. As the example in Fig. 1, an OD pair from residential zone C to commercial zone B is similar to an OD pair from residential zone D to commercial zone A semantically, even though A and D are geographically far away from B and C, respectively. Despite their importance, the spatial-temporal dependencies among different OD pairs are not well modeled and characterized by the existing methods.

To address the aforementioned challenge, this paper proposes a novel deep learning framework named *Spatial-Temporal Encoder-Decoder Residual Multi-Graph Convolutional network* (ST-ED-RMGC) to simultaneously predict ride-sourcing passenger demand in various OD pairs. First, we construct multiple OD graphs, in which each OD pair is viewed as a node, and the adjacent matrices of nodes are established to represent different aspects of the relationships among OD pairs, such as neighborhood, distance, functional similarity, and historical demand correlations, and so on. Second, to capture both spatial and temporal correlations, we use a residual



(a) Irregular zones based on zip codes



(b) Sample fully connected graph with OD pairs as nodes (e.g., the 1st node is the OD pair with origin No. 4 and destination No. 12)

Fig. 2. Zone division and fully connected graph.

multi-graph convolutional (RMGC) network to capture the spatial correlations among OD pairs in different time intervals, and a regular long-short term memory (LSTM) network to characterize the temporal correlations of each OD pair itself. In view of the different input/output formats of the RMGC network (with graphs as inputs and outputs) and LSTM network (with time series as inputs and outputs), we encode the outputs of these two networks into a dense latent vector space. Then the RMGC network is reused to decode the latent vector back to the OD graph to predict future OD demand. The main contributions of this paper are summarized as follows:

- We characterize the pair-wise relationships between different OD pairs by constructing multiple OD graphs, including origin- and destination-based neighborhood relationship graphs, origin- and destination-based functional similarity graphs, origin- and destination-based distance graphs, and mobility pattern correlation graph.
- We propose a novel deep learning model with a well-designed encoder-decoder structure to model both the spatial dependencies across different OD pairs and the temporal dependencies of the OD pairs themselves. The structure learns spatial and temporal features in an end-to-end learning framework.
- We show that the proposed model significantly outperforms the benchmark algorithms, based on the evaluations with the datasets of for-hire-vehicles (i.e., vehicles providing ride-sourcing services) in Manhattan, New York City.

The rest of the paper is organized as follows: Section 2 summarizes the recent studies on ride-sourcing demand forecasting; Section 3 gives a formal definition of the research problem and the spatio-temporal features used as inputs. Section 4 describes the proposed ST-ED-RMGC model from overall architecture to its detailed components. Section 5 presents the dataset and the experimental results, and Section 6 concludes the paper and outlooks future studies.

2. Related work

2.1. Demand forecasting

Real-time prediction of passenger demand or traffic states is a fundamental requirement for the control and operations of transportation systems. In the literature, many efforts have been directed towards the prediction of traffic flow (Zhang et al., 2019a,b,c,d; Yu et al., 2019; Wang et al., 2018; Zhang et al., 2019a,b,c,d; Li et al., 2017; Wu et al., 2018a,b; Zhu et al., 2016; Zhu et al., 2018; Zhu et al., 2019; Guo et al., 2019), as well as bike flow (Chai et al., 2018; Lin et al., 2018). Particularly, the availability of a massive amount of mobility data collected via the online e-hailing platform makes the real-time prediction of ride-sourcing demand possible, such as zone-based demand prediction examined in a few recent studies (Ke et al., 2017; Zhang et al., 2017; Yao et al., 2018a; Zhang et al., 2019a,b,c,d; Yao et al., 2018b). In these studies, a city is partitioned into various squares, and the near future (i.e., from 10 min to 2 h ahead) passenger demand originating from each square is predicted. Instead of square meshes, Ke et al. (2018) used hexagons as the basic grids for demand prediction, which resembles a circle and can better characterize the inflow and outflow between neighboring grids.

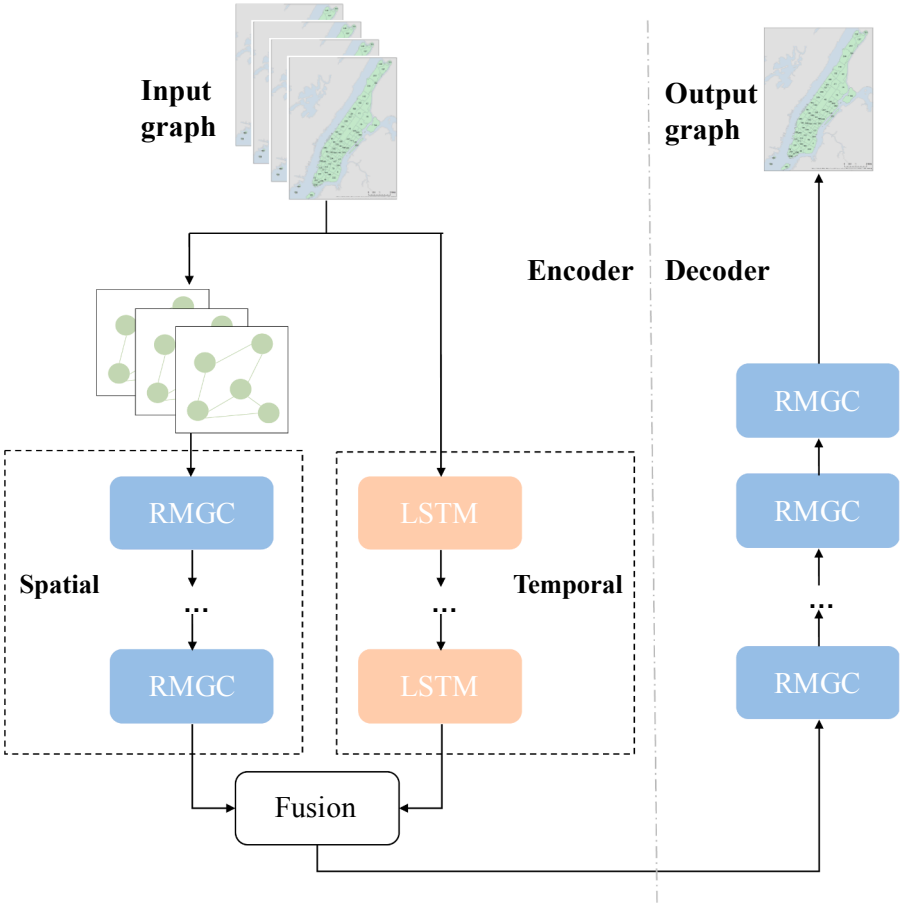
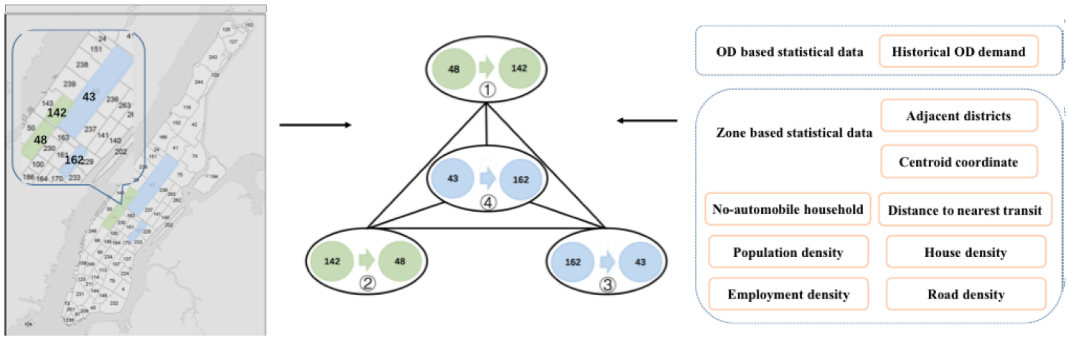


Fig. 3. Framework of ST-ED-RMGC Model.

Through this spatial discretization, the input and output data of ride-sourcing demand can be treated as images and some deep learning techniques widely used in image recognition can be adopted, such as convolutional neural networks (CNNs) and LSTM networks.

However, CNNs only capture the local spatial correlations in a geographical manner, but fails to model the semantic correlations between two zones, which are geographically far away from each other but share common functionalities. Moreover, CNNs are not well adaptable to the prediction on irregular zones without exhibiting an image-like data structure (e.g., administrative regions and Zip-code regions). To tackle this problem, some recent researches introduced graph convolutional neural network (GCN) and graph embedding techniques into ride-sourcing demand predictions. Yao et al. (2018a) proposed a deep multi-view spatial-temporal network, with three major views: spatial view, temporal view, and semantic view. The semantic view uses graph embedding techniques to incorporate the information of functionalities of various zones into the framework. Sun et al. (2019) proposed a multi-view graph convolutional network to predict inflows and outflows in irregular regions. Geng et al. (2019a) proposed an ST-MGCN model to forecast zone-based ride-sourcing demand, which first characterizes the non-Euclidean relationship among zones by designing various adjacent matrices and then applies recurrent neural networks to learn the temporal correlations. Geng et al. (2019b) further utilized a grouped GCN in lower layers and a multi-linear relationship GCN in higher layers to learn more generalized features.

While there is a large body of literature on zone-based demand prediction, there are only a few preliminary studies on OD-based demand prediction. Liu et al. (2019) proposed a contextualized spatial-temporal network to predict the OD taxi demand in New York. They partitioned the city into squares and used a 3D matrix (with two dimensions of height and width of the city map and one dimension of the total number of squares) to encode the origin-destination information, such that various types of CNNs can be utilized to capture the spatial dependences among the square-shape regions. Wang et al. (2019) estimated the OD matrix with grid-embedding-based multi-task learning. The grid embedding module was used to obtain the pre-weighted features by modeling the spatial mobility patterns of passengers, which were then fed into the multi-task learning module to predict future OD demand. Xiong et al. (2019) fused line GCN and Kalman filter to predict the OD demand of the whole traffic network. However, they treated the origin and destination grids separately and learned the local features around the separate grids. The adjacent matrices did not take advantage of semantic information, but only contain information about the distances and flows between any two grids.



(a) Graph construction and input information

Origin neighborhood relationship

$$A_n^O = \begin{array}{c} \\ ① \\ ② \\ ③ \\ ④ \end{array} \begin{array}{c} \begin{array}{cccc} ① & ② & ③ & ④ \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \end{array}$$

Destination neighborhood relationship

$$A_n^D = \begin{array}{c} \\ ① \\ ② \\ ③ \\ ④ \end{array} \begin{array}{c} \begin{array}{cccc} ① & ② & ③ & ④ \end{array} \\ \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

(b) Neighborhood relationship adjacent matrix

OD order quantity correlation

$$A_c = \begin{array}{c} \\ ① \\ ② \\ ③ \\ ④ \end{array} \begin{array}{c} \begin{array}{cccc} ① & ② & ③ & ④ \end{array} \\ \begin{bmatrix} 0 & cor(Q_1,Q_2) & cor(Q_1,Q_3) & cor(Q_1,Q_4) \\ cor(Q_1,Q_2) & 0 & cor(Q_2,Q_3) & cor(Q_2,Q_4) \\ cor(Q_1,Q_3) & cor(Q_2,Q_3) & 0 & cor(Q_3,Q_4) \\ cor(Q_1,Q_4) & cor(Q_2,Q_4) & cor(Q_3,Q_4) & 0 \end{bmatrix} \end{array}$$

(c) Mobility pattern correlation adjacent matrix

Origin centroid distance

$$A_d^O = \begin{array}{c} \\ ① \\ ② \\ ③ \\ ④ \end{array} \begin{array}{c} \begin{array}{cccc} ① & ② & ③ & ④ \end{array} \\ \begin{bmatrix} 0 & d_{48,142}^{-1} & d_{48,162}^{-1} & d_{48,43}^{-1} \\ d_{142,48}^{-1} & 0 & d_{142,162}^{-1} & d_{142,43}^{-1} \\ d_{162,48}^{-1} & d_{162,142}^{-1} & 0 & d_{162,43}^{-1} \\ d_{43,48}^{-1} & d_{43,142}^{-1} & d_{43,162}^{-1} & 0 \end{bmatrix} \end{array}$$

Destination centroid distance

$$A_d^D = \begin{array}{c} \\ ① \\ ② \\ ③ \\ ④ \end{array} \begin{array}{c} \begin{array}{cccc} ① & ② & ③ & ④ \end{array} \\ \begin{bmatrix} 0 & d_{142,48}^{-1} & d_{142,43}^{-1} & d_{142,162}^{-1} \\ d_{48,142}^{-1} & 0 & d_{48,43}^{-1} & d_{48,162}^{-1} \\ d_{43,142}^{-1} & d_{43,48}^{-1} & 0 & d_{43,162}^{-1} \\ d_{162,142}^{-1} & d_{162,48}^{-1} & d_{162,43}^{-1} & 0 \end{bmatrix} \end{array}$$

(d) Centroid distance adjacent matrix

Origin functional similarity

$$A_f^O = \begin{array}{c} \\ ① \\ ② \\ ③ \\ ④ \end{array} \begin{array}{c} \begin{array}{cccc} ① & ② & ③ & ④ \end{array} \\ \begin{bmatrix} 0 & E_{48,142}^{-1} & E_{48,162}^{-1} & E_{48,43}^{-1} \\ E_{142,48}^{-1} & 0 & E_{142,162}^{-1} & E_{142,43}^{-1} \\ E_{162,48}^{-1} & E_{162,142}^{-1} & 0 & E_{162,43}^{-1} \\ E_{43,48}^{-1} & E_{43,142}^{-1} & E_{43,162}^{-1} & 0 \end{bmatrix} \end{array}$$

Destination functional similarity

$$A_f^D = \begin{array}{c} \\ ① \\ ② \\ ③ \\ ④ \end{array} \begin{array}{c} \begin{array}{cccc} ① & ② & ③ & ④ \end{array} \\ \begin{bmatrix} 0 & E_{142,48}^{-1} & E_{142,43}^{-1} & E_{142,162}^{-1} \\ E_{48,142}^{-1} & 0 & E_{48,43}^{-1} & E_{48,162}^{-1} \\ E_{43,142}^{-1} & E_{43,48}^{-1} & 0 & E_{43,162}^{-1} \\ E_{162,142}^{-1} & E_{162,48}^{-1} & E_{162,43}^{-1} & 0 \end{bmatrix} \end{array}$$

(e) Functional similarity adjacent matrix

Fig. 4. Adjacent matrixes processing.

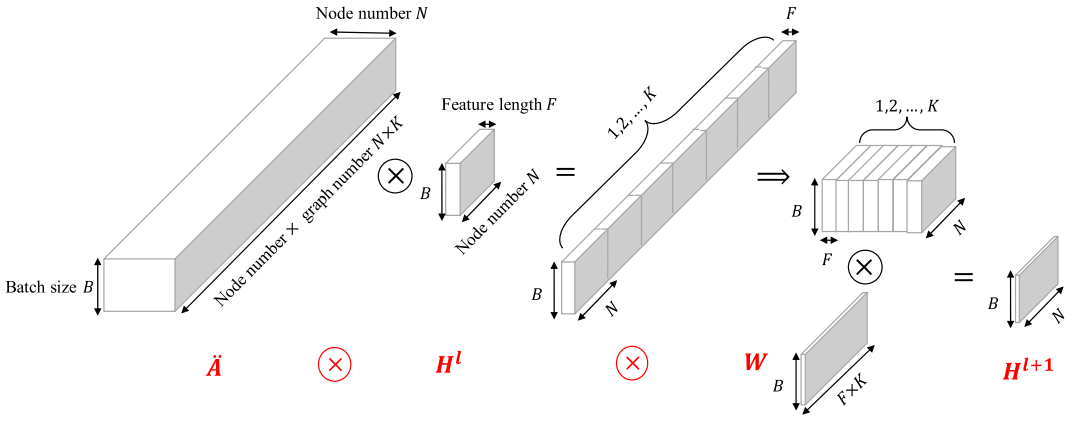


Fig. 5. Architecture of one MGC layer.

2.2. Graph convolution

CNNs have been widely adopted in the field of image processing due to its outstanding performance in capturing local spatial correlations. However, CNNs have the drawback of requiring input and output data as matrices or tensors, and thus are difficult to deal with many real-world problems that rely on arbitrary graphs with non-Euclidean correlations. To address this issue, a large number of GCNs have been proposed, in which the convolution operators are redefined for graph data, since the work by Bruna et al. (2014). Some studies (Li et al., 2018; Levie et al., 2017; Kipf and Welling, 2016) employed spectral graph theory into GCNs to transfer information from the original graph domain to the frequency domain to capture the non-Euclidean relationships among vertices. Some other studies directly perform convolution operations in the original graph domain by aggregating the features of neighboring nodes (Gao et al., 2018; Monti et al., 2017a,b). In the later methods, computations are performed in a batch of nodes instead of the whole graph to reduce computational complexity. Readers may refer to Wu et al. (2019) for a comprehensive review of GCNs. Nowadays, GCNs become popular in the transportation field due to its outperformance in many applications, such as traffic flow prediction (Wu et al., 2018a,b; Do et al., 2019), traffic speed and state prediction (Zhang et al., 2019a,b,c,d), and parking occupancy prediction (Yang et al., 2019).

3. Research problem

In this section, we introduce our research problem by giving a formal definition of the OD graph and describing the geographical, semantic, and temporal features used in our study.

3.1. OD graph

As mentioned above, previous studies usually divide the space of interest into various regular grids, such as squares and hexagons. These segmentations enable the use of standard machine learning algorithms, such as CNNs, but cannot well represent the administrative and functional properties of the regions under consideration. In this paper, we partition the examined city, Manhattan in New York City, into various irregular zones, according to the administrative zip codes as shown in Fig. 2 (a). A day is uniformly divided into several intervals (for example, 24 hours). Our target is to predict the quantity of the order requests in various OD pairs simultaneously in each time interval.

Unlike the conventional traffic network graph with each vertex representing an intersection or a zone, we construct a tailored OD graph $G = (V, E, \boldsymbol{A})$, in which each vertex of the graph refers to an OD pair, as shown in Fig. 2 (b). V is the set of OD pairs and $N = |V|$ is the number of OD pairs in each time interval, E denotes the set of edges, and $\boldsymbol{A} \in \mathbb{R}^{N\times N}$ defines the adjacent matrixes with their entries representing the connections between vertices (i.e. OD pairs). Note that the OD graph is fully connected, indicating that there exists an edge connecting any two OD pairs, although their connections could be weak due to far geographical and semantic distances.

3.2. Research problem and features

Let $x_i^{(d,t)}$ denote the passenger demand (quantity of requested orders) in the ith OD pair at the time interval t of day d, where $i \in V$(the set of OD pairs), $x_i^{(d,t)} \in \mathbb{R}^+$. Let $\boldsymbol{X}^{(d,t)}$ denote the passenger demand in all OD pairs at the time interval t of day d. To predict $\boldsymbol{X}^{(d,t)}$, all the OD demand prior to time interval t of day d can be used as features. However, feeding all of the historical OD demand into the model is unnecessary and infeasible due to the limitations of computational resources. As pointed out by Zhang et al. (2017), there are two major types of temporal dependencies: tendency (demand is affected by the historical demand in the past few intervals) and periodicity (demand repeats similar patterns over days and weeks). With this knowledge, we extract the following historical

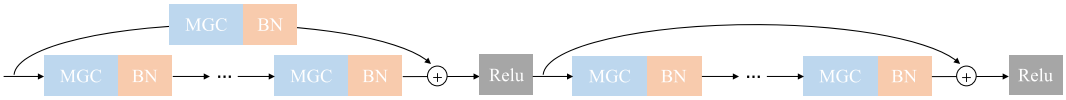


Fig. 6. Convolutional block followed by identity block in RMGC.

observations as the features:

1) Tendency-based features: the demand in the OD graph at the last two time intervals, i.e., $\boldsymbol{X}^{(d,t-1)}$ and $\boldsymbol{X}^{(d,t-2)}$;
2) Periodicity-based features over a day: the demand in the OD graph at the same time interval in the previous day, i.e., $\boldsymbol{X}^{(d-1,t)}$;
3) Periodicity-based features over a week: the demand in the OD graph at the same time interval in the same day of last week, i.e., $\boldsymbol{X}^{(d-7,t)}$.

Then the OD ride-sourcing demand prediction problem can be formulated below.

Problem 1:. *To learn a function* $f(\cdot): \mathbb{R}^{N\times T} \rightarrow \mathbb{R}^{N}$ *that maps the historical demand of all OD pairs on an OD graph to the demand of all OD pairs on the same OD graph in the next time interval*:

$$\boldsymbol{X}(d,t) = \left[\boldsymbol{X}^{(d-7,t)}, \boldsymbol{X}^{(d-1,t)}, \boldsymbol{X}^{(d,t-2)}, \boldsymbol{X}^{(d,t-1)}\right] \overset{f}{\rightarrow} \boldsymbol{X}^{(d,t)} \tag{1}$$

4. The proposed ST-ED-RMGC model

4.1. Overview of model framework

Fig. 3 briefly demonstrates the architecture of the proposed ST-ED-RMGC model, which uses an encoder-decoder framework. There are two encoders: one spatial encoder and one temporal encoder. The spatial encoder utilizes several RMGCs to model the spatial correlations between the OD pairs from different aspects (including geographical distances, functionality similarities, and mobility pattern correlations). The temporal encoder proposes a spatial LSTM model to learn the temporal dependencies of each OD pair. To fuse the spatial and temporal models in one end-to-end learning framework, we flatten the outputs of the two decoders into two dense latten vectors, which are then concatenated. Finally, in the decoder part, several RMGC networks are reused to transform the compressed vector back to an OD graph used to predict the target OD demand.

4.2. Detailed methodologies

This section presents the detailed methodologies of different modules in the framework. We will start with the design of the multiple graphs for capturing various types of spatial relationships among the OD pairs, and then present the technical details of the RMGC module. Besides, we will give a brief introduction to the spatial LSTM module, which is slightly different from the standard LSTM network. Finally, we describe the encoder-decoder architecture that combines the abovementioned modules in one end-to-end learning framework.

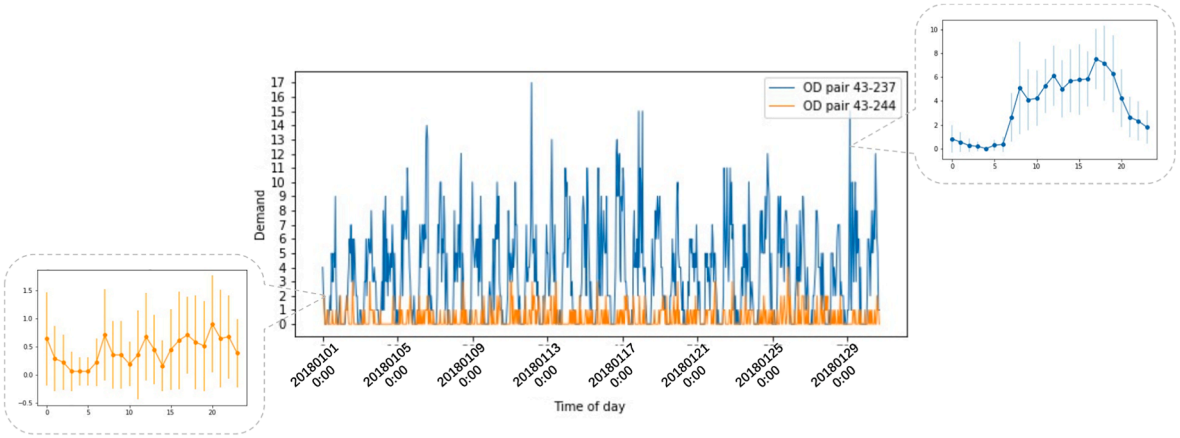


Fig. 7. OD demand distribution.

4.2.1. Modelling spatial correlations between OD pairs

This subsection models the multi-graphs and defines the corresponding adjacent matrixes. To accurately predict OD demand, the model calibrates the OD demand with the historical demand data of correlated OD pairs over space given the measurement of the OD pair correlation. This implies that it is critical to explore the inherent relationships between OD pairs both geographically and semantically. From the geographical aspect, if the origins/destinations of two OD pairs are adjacent to each other, we may naturally expect that the demands of these two OD pairs have strong correlations (since the demands may origin from a common transportation station, a shopping mall, or an estate, etc.). This motivates us to design a neighborhood relationship graph to indicate whether any two of the origins or destinations are neighbors to each other. We also expect that the demands of two OD pairs with close origins or/and destinations have relatively strong dependencies, which motivates us to construct an adjacent matrix that describes the distances between the centroids of the origins and destinations respectively. From the semantical aspect, it is generally expected that the demands of two OD pairs with similar functionality (such as commercial regions, recreational regions, residential regions) are prone to have strong relationships. In addition, it is intuitive to expect that the OD pairs with similar historical demand patterns will exhibit similar demand patterns in the future. In what follows, we will present clear definitions of these four types of adjacent matrices: (1) neighborhood relationship graphs $G_n(V,E,\boldsymbol{A}_n)$, $\boldsymbol{A}_n \in \mathbb{R}^{N\times N}$; (2) functional similarity graphs $G_f(V,E,\boldsymbol{A}_f)$, $\boldsymbol{A}_f \in \mathbb{R}^{N\times N}$; (3) centroid distance graphs $G_d(V,E,\boldsymbol{A}_d)$, $\boldsymbol{A}_d \in \mathbb{R}^{N\times N}$; and (4) mobility pattern correlation graph $G_c(V,E,\boldsymbol{A}_c)$, $\boldsymbol{A}_c \in \mathbb{R}^{N\times N}$. It is also noteworthy that other matrices, such as connectivity (measured by the number of highways or subways connecting the two zones), can be easily incorporated into our framework if the data is available.

(1) *Neighborhood relationship graphs*. As mentioned in Section 1, demands in OD pairs with nearby origin or destination are more likely to have similar patterns. We define two adjacent matrices to indicate whether two OD pairs have neighboring origins or destinations, respectively:

$$\left[\boldsymbol{A}_n^O\right]_{i,j} = \begin{cases} 1, \text{if origin of OD pair } i \text{ and OD pair } j \text{ are adjacent} \\ 0, \text{otherwise} \end{cases}, \forall i,j \in V \tag{2}$$

$$\left[\boldsymbol{A}_n^D\right]_{i,j} = \begin{cases} 1, \text{if destinations of OD pair } i \text{ and OD pair } j \text{ are adjacent} \\ 0, \text{otherwise} \end{cases}, \forall i,j \in V \tag{3}$$

where $\left[\boldsymbol{A}_n^O\right]_{i,j}$ ($\left[\boldsymbol{A}_n^D\right]_{i,j}$) refers to the element in the ith row and jth column of adjacent matrix $\boldsymbol{A}_n^O$ ($\boldsymbol{A}_n^D$) of neighborhood relationship.

(2) *Functional similarity graphs*. Different regions in a city may have different functionalities or land-use properties. Some are commercial regions with many shopping malls and restaurants, some are tourist areas with a park, and some are residential zones full of departments and houses. In this paper, we use the land-use properties provided by Smart Location Database (please refer to the details in Section 5.1) to define the functionalities of zones in Manhattan. The properties we select involve households without owning an automobile, house density, population density, employment density, road density as well as the average meters to nearest transit, which are highly related to the land-use type and travel mode choice. Note that the administrative zones have heterogeneous areas, and thus we divide all these measures by the area of the corresponding zone. Let $\boldsymbol{F}_i^O$ and $\boldsymbol{F}_i^D$ denote the vector of the functionalities of the origin zone and destination zone of OD pair i, then two functional similarity adjacent matrices can be constructed as follows:

$$\left[\boldsymbol{A}_f^O\right]_{i,j} = \left[\sqrt{\left(\boldsymbol{F}_i^O - \boldsymbol{F}_j^O\right)\left(\boldsymbol{F}_i^O - \boldsymbol{F}_j^O\right)^T}\right]^{-1}, \forall i,j \in V \tag{4}$$

$$\left[\boldsymbol{A}_f^D\right]_{i,j} = \left[\sqrt{\left(\boldsymbol{F}_i^D - \boldsymbol{F}_j^D\right)\left(\boldsymbol{F}_i^D - \boldsymbol{F}_j^D\right)^T}\right]^{-1}, \forall i,j \in V \tag{5}$$

We can see that, as the vectors of functionalities $\boldsymbol{F}_i^O$,$\boldsymbol{F}_j^O$ of two OD pairs i, j become closer to each other, their functional similarity gets larger.

(3) *Centroid distance graphs*. Due to the irregular zones of different sizes, we further introduce two centroid distance graphs to represent the geographical relationships between OD pairs. The adjacent matrices in the origin-based (distance between the centroids of origins of each two OD pairs) and destination-based centroid distance (distance between the centroids of destinations of each two OD pairs) graphs are defined by the inverse of the straight-line distance between the centroids of the zones (the shorter the distance, the stronger the relationship), shown as follows:

$$\left[\boldsymbol{A}_d^O\right]_{i,j} = \left[haversine\left(lng_i^O, lat_i^O, lng_j^O, lat_j^O\right)\right]^{-1}, \forall i,j \in V \tag{6}$$

$$\left[\boldsymbol{A}_d^D\right]_{i,j} = \left[haversine\left(lng_i^D, lat_i^D, lng_j^D, lat_j^D\right)\right]^{-1}, \forall i,j \in V \tag{7}$$

where $lng_i^O, lat_i^O, lng_j^O, lat_j^O$ are the longitudes and latitudes of the origins of OD pairs i, j, while $lng_i^D, lat_i^D, lng_j^D, lat_j^D$ are the longitudes and latitudes of the destinations of OD pairs i, j, respectively. The function $haversine(\cdot)$ measures the straight-line distance between two locations on earth.

(4) *Mobility pattern correlation graphs*. It is intuitive that OD pairs with analogous mobility patterns (historical demand trends) share

some common characteristics, and thus can guide predictions for each other. Let $\boldsymbol{Q}_i$, be a vector recording the historical demand (over multiple months) of OD pair i. Then the adjacent matrix of the mobility pattern correlation graph is formulated by

$$[\boldsymbol{A}_c]_{i,j} = \frac{Cov(\boldsymbol{Q}_i, \boldsymbol{Q}_j)}{\sqrt{var(\boldsymbol{Q}_i)var(\boldsymbol{Q}_j)}}, \forall i, j \in V \tag{8}$$

where $Cov(\cdot,\cdot)$ calculates the covariance between two vectors, while $var(\cdot)$ calculates the variance of one vector. Fig. 4 illustrates the adjacent matrixes.

4.2.2. Residual Multi-Graph convolutional (RMGC) network

Now we introduce the RMGC network that combines a multi-graph convolutional network and a residual module, to capture the spatial correlations between OD pairs. The foundation of the RMGC network in our model is GCN, which is the primary tool of spatial feature encoding. To deal with the multi-graph problem, we design an MGC network by reshaping the architecture of a GCN layer. The MGC acts as the basic component of this spatial-feature encoder. Afterward, we stack multiple MGC layers in the deep learning network to improve the training performance and introduce a residual network to address the issue of gradient explosion, which constitutes the RMGC block. The output of this RMGC-based encoder is generated through multiple layers of RMGC block and flattened to be a one-dimension feature. In what follows we extend the principle and structure of the aforementioned RMGC network in detail.

GCN breaks through the restriction of Euclidean structure and thus make it possible for many tasks, including social network analysis (Wu et al., 2018a,b; Ying et al., 2018), abnormal detections on graphs (Monti et al., 2017a,b), graph embedding and traffic forecasting (Geng et al., 2019a; Geng et al., 2019b), etc. Two main types of GCNs, spatial-based and spectral-based, have been developed with different focuses and advantages. Spatial-based networks use local graph convolution units to extract feature information from neighboring vertices, while spectral-based approaches (Defferrard et al., 2016) introduce spectral filters to define graph convolutions. In this paper, we use spectral-based methods to build a basic GCN model, which in essence maps the original graph signals (raw features on a graph) to a parameterized Fourier domain. However, training the parameters is computationally expensive. To address this issue, Defferrard et al. (2016) introduced a Chebyshev polynomial expansion (up to Kth order) to obtain an efficient approximation, and Kipf and Welling (2016) further simplified the spectral filter to a Chebyshev polynomial of order $K = 1$. The later model is also called 1stChebNet and has the following form:

$$\boldsymbol{H}_{l+1} = \sigma\left(\widehat{\boldsymbol{A}}\boldsymbol{H}_l\boldsymbol{W}_l\right) = \sigma\left(\boldsymbol{D}^{-1/2}\boldsymbol{A}\boldsymbol{D}^{-1/2}\boldsymbol{H}_l\boldsymbol{W}_l\right) \tag{9}$$

where $\boldsymbol{H}_{l+1}$ and $\boldsymbol{H}_l$ are the activations in the lth and $l+1$th hidden layer; σ is the activation function, which can be $Relu(\cdot)$ or $Linear(\cdot)$ or $Tahn(\cdot)$; W_l denotes the trainable weight matrix connecting lth and $l+1$th hidden layer. $\boldsymbol{A} = \boldsymbol{A} + \boldsymbol{I}$ is the adjacent matrix added by self-connections for the purpose of maintaining the information of the node itself in convolution (Chai, D. et al., 2018). $\boldsymbol{D}$ is the degree matrix, in which $\boldsymbol{D}_{ii} = \sum_j \boldsymbol{A}_{ij}$ and $\boldsymbol{W}_l$ is a matrix of trainable weights in the lth graph convolutional layer. The transformation function in Eq. (9) describes the mapping from a hidden layer to the next hidden layer through one graph with a single adjacent matrix. The dimensions of the tensors are: $\widehat{\boldsymbol{A}} \in \mathbb{R}^{N\times N}$, $\boldsymbol{H}_l \in \mathbb{R}^{N\times F}$, $\boldsymbol{W}_l \in \mathbb{R}^{F\times O}$ and $\boldsymbol{H}_{l+1} \in \mathbb{R}^{N\times O}$, where N, F, O are the number of nodes, input features, and output features respectively.

Next, to enable the learning from multiple graphs, we design a training architecture shown in Fig. 5. Suppose we have K adjacent matrixes, and let $\widehat{\boldsymbol{A}_k} \in \mathbb{R}^{N\times N}$ denote the kth adjacent matrix, $k \in \{1, \cdots, K\}$. In each training batch with a batch size B, we first duplicate each adjacent matrices by B times and then concatenate them into a tensor $\ddot{\boldsymbol{A}} \in \mathbb{R}^{B\times(N*K)\times N}$. Given the dimensions of the input and output features are F and O respectively, then we have the input tensor $\boldsymbol{H}_l \in \mathbb{R}^{B\times N\times F}$, the learnable weight matrix $\boldsymbol{W} \in \mathbb{R}^{(K*F)\times O}$, and the output tensor $\boldsymbol{H}_{l+1} \in \mathbb{R}^{B\times N\times O}$. To map the input tensor $\boldsymbol{H}_l$ to the output tensor $\boldsymbol{H}_{l+1}$, we first conduct a batch dot production between $\ddot{\boldsymbol{A}}$ and $\boldsymbol{H}_l$, which generates a tensor $\boldsymbol{M} \in \mathbb{R}^{B\times(N*K)\times F}$. Then the generated tensor $\boldsymbol{M}$ is reshaped to a new tensor $\boldsymbol{M} \in \mathbb{R}^{B\times N\times(F*K)}$, and finally the batch dot product of $\boldsymbol{M}$ and $\boldsymbol{W}$ produces the output tensor $\boldsymbol{H}_{l+1} \in \mathbb{R}^{B\times N\times O}$.

Then, to train the networks in a deep neural network structure without suffering from gradient explosion, now we develop two multi-graph convolutional network based (MGC-based) residual blocks: the identity block and convolutional block. Residual learning (He et al., 2016) is a powerful tool that allows the training of super deep networks, and is widely used in many traditional convolutional neural network structures. The basic residual unit (identity block) is formulated by,

$$\boldsymbol{H}_{l+1} = \boldsymbol{H}_l + \sigma\left(\boldsymbol{D}^{-1/2}\boldsymbol{A}\boldsymbol{D}^{-1/2}\boldsymbol{H}_l\boldsymbol{W}_l\right) \tag{10}$$

where the first term at the right-hand-side refers to the shortcut path, and the second term represents the main path. It is noteworthy that a common practice is to stack multiple MGC layers in the main path. Moreover, in the case that requires different feature dimensions between the input tensor $\boldsymbol{H}_l$ and the output tensor $\boldsymbol{H}_{l+1}$, an MGC layer can be added to the shortcut path to maintain the feature dimension consistency (as shown in Fig. 6). This residual block is called the convolutional block.

After constructing RMGC, we stack multiple convolutional and identity RMGC blocks, and flatten the output into a latent vector with a one-dimensional feature as the ultimate output of this RMGC encoder:

Table 1
Prediction error of baseline methods and proposed model with different parameters (mean ± std).

Method	RMSE	MAE	MAPE
HA	8.42	5.40	0.75
XGB	5.20	3.59	0.48
MLP	5.20	3.59	0.49
GBDT	5.21	3.59	0.49
RF	5.61	3.87	0.51
LASSO	5.37	3.70	0.51
LSTM	5.33 ± 0.08	3.68 ± 0.06	0.51 ± 0.03
Spatial LSTM	4.96 ± 0.01	3.60 ± 0.01	0.45 ± 0.01
MGC	4.64 ± 0.02	3.20 ± 0.02	0.42 ± 0.01
ED-MGC	4.58 ± 0.03	3.12 ± 0.02	0.41 ± 0.00
RMGC	4.48 ± 0.05	3.08 ± 0.03	0.40 ± 0.01
ST-ED-RMGC	4.29 ± 0.02	2.96 ± 0.02	0.38 ± 0.01

$$\boldsymbol{L}_1 = \sigma\left(\boldsymbol{W}_1 \cdot Flatten\left(RMGC\left(\boldsymbol{X}, \boldsymbol{W}_{RMGC}\right)\right) + b_1\right) \tag{11}$$

where $RMGC(\cdot)$ refers to the transformation through multiple RMGC blocks with trainable weights $\boldsymbol{W}_{RMGC}$, $Flatten(\cdot)$ is an operator that flattens the outputs of the last RMGC layer, $\boldsymbol{W}_1$ and b_1 are trainable weights and bias in the latent layer. $\boldsymbol{L}_1$ has a dimension of $B \times V_1$, where B is the batch size and V_1 is the latent feature dimension of the RMGC encoder.

4.2.3. Spatial LSTM

This subsection presents the other paratactic encoder structure in our model, i.e. the spatial LSTM encoder, which primarily captures the temporal feature. Slightly different from the conventional LSTM, the spatial LSTM in this paper is required to deal with input tensor with both spatial and temporal information simultaneously. To this end, we develop the spatial LSTM that is a part of the encoder, with its technique details presented below.

LSTM, as a recurrent neural network, was born for capturing temporal dependencies, and thus are widely used in many traffic demand forecasting problems. In most previous studies, historical features of one zone or one road segment are taken as inputs, and the demand of this zone or road segment in the next time interval is predicted. In our paper, to incorporate spatial and temporal models in one end-to-end learning framework in a more sensible way, we propose a spatial LSTM that learns features from all OD pairs and outputs the high-level information into a latent vector, instead of using various separate LSTMs for each OD pair. Recall that the input of features $\boldsymbol{X}$ has a shape of $B \times N \times F$, i.e., $\boldsymbol{X} \in \mathbb{R}^{B \times N \times F}$, where B is the batch size, F is the number of sliced windows for extracting historical features ($F = 4$ as defined in Problem 1). In the RMGC encoder, F historical observations, i.e., the demand of all the OD pairs over time, are treated as features; while the number of OD pairs, i.e., N, turns out to be the target dimension of the output tensor. However, the spatial LSTM trains the time series data, and thus the information of spatial nodes should be the feature. Therefore, the proposed spatial LSTM first reshapes $\boldsymbol{X}$ by transposing its second and third dimensions, leading to a new tensor $\underset{LSTM}{\boldsymbol{X}}$ with a shape of $B \times F \times N$, i.e., $\underset{LSTM}{\boldsymbol{X}} \in \mathbb{R}^{B \times F \times N}$. With $\underset{LSTM}{\boldsymbol{X}}$ as input tensor, the LSTM treats the second dimension of F as the time dimension, and the third dimension of N as the feature dimension. In other words, the historical demands in all OD pairs are treated as features that are fed into one single LSTM.

After building the spatial LSTM module, we stack multiple LSTM modules and flatten the output tensor to a latent vector:

$$\boldsymbol{L}_2 = \sigma\left(\boldsymbol{W}_2 \cdot Flatten\left(LSTM\left(\underset{LSTM}{\boldsymbol{X}}, \boldsymbol{W}_{LSTM}\right)\right) + b_2\right) \tag{12}$$

where $LSTM(\cdot)$ represents multiple LSTM layers with trainable weights $\boldsymbol{W}_{LSTM}$, $Flatten(\cdot)$ flattens the outputs of the last LSTM layer, $\boldsymbol{W}_2$ and b_2 are the weights and bias in the latent layer, $\boldsymbol{L}_2$ has a dimension of $B \times V_2$, where V_2 is the latent feature dimension of the spatial LSTM encoder.

4.2.4. Encoder fusion and decoder

We simply fuse the outputs of the abovementioned two encoders by concatenating them together:

$$\boldsymbol{L} = [\boldsymbol{L}_1, \boldsymbol{L}_2] \tag{13}$$

where the latent vector $\boldsymbol{L}$ contains both the spatial and temporal features of the historical demand. To predict the future OD demand, we first introduce an intermediate layer to expand the dimension of vector $\boldsymbol{L}$ to a new vector with a shape of $B \times N$ (N is the number of

OD pairs), and then we reshape the new vector to a tensor with a shape of $B \times N \times 1$, which becomes a valid input format for RMGC. By stacking various RMGC modules, we can finally obtain the estimated demand of all OD pairs on an OD graph, i.e. $\widehat{\boldsymbol{X}}^{(d,t)}$. Formally, the decoder architecture can be formulated by

$$\widehat{\boldsymbol{X}}^{(d,t)} = RMGC^{d}(Reshape(\sigma(\boldsymbol{W}_3 \cdot \boldsymbol{L} + b_3)), \boldsymbol{W}_{RMGC-D}) \tag{14}$$

where $RMGC^{d}(\cdot)$ are multiple RMGC modules with learnable weights $\boldsymbol{W}_{RMGC-D}$, $\boldsymbol{W}_3$ and b_3 are the trainable weights of the intermediate layer for dimension expansion, $Reshape(\cdot)$ is an operator that reshapes a vector into a tensor. Let $\boldsymbol{W}$, $\boldsymbol{b}$ be all the trainable weights and biases in the whole encoder-decoder architecture, we can train the weights and biases by solving the following optimization problem:

$$\min_{\boldsymbol{W},\boldsymbol{b}} \sum_{d} \sum_{t} \| \widehat{\boldsymbol{X}}^{(d,t)} - \boldsymbol{X}^{(d,t)} \|_2^2 + \alpha \|\boldsymbol{W}\|_2^2 \tag{15}$$

where the first term minimizes the squared loss between the predicted OD demand pattern and the actual one, and the second term is an L2-norm regularization term to avoid extremely complex models that may lead to over-fitting. The training algorithm of the model is demonstrated below.

Algorithm 1. ST-ED-RMGC training algorithm

Input OD pair number $i \in V$,
Historical demand of all OD pairs $\{x_i^1, ..., x_i^T\}, \forall i \in V$,
The graphs $\mathbf{G}$: neighborhood relationship graphs $G_n(V, E, \mathbf{A}_n)$;
functional similarity graphs $G_f(V, E, \mathbf{A}_f)$;
centroid distance graphs $G_d(V, E, \mathbf{A}_d)$;
mobility pattern correlation graph $G_c(V, E, \mathbf{A}_c)$
Output ST-ED-RMGC with well-trained parameters $\mathbf{W}$
// Construct a set of input-output instances $\mathcal{D}$
Initialize a null set: $\mathcal{D} \leftarrow \varnothing$
for time interval $t (1 \le t \le T)$ **do**
Get temporal features of all OD pairs at each time interval: $\widetilde{\mathbf{X}}^{(d,t)} = [\mathbf{X}^{(d-7,t)}, \mathbf{X}^{(d-1,t)}, \mathbf{X}^{(d,t-2)}, \mathbf{X}^{(d,t-1)}]$
// $\mathbf{X}^{(d,t)}$ is the prediction target at time t
Put training sample into the dataset: $\mathcal{D} \leftarrow \mathcal{D} + (\widetilde{\mathbf{X}}^{(d,t)}, \mathbf{X}^{(d,t)})$
end for
Divide $\mathcal{D}$ into training and test datasets $\mathcal{D}_{train}$, $\mathcal{D}_{valid}$, $\mathcal{D}_{test}$

// Training ST-ED-RMGC model
Initialize the hidden status, all weights and bias parameters
Concatenate the graphs: $\mathbf{A} \leftarrow [\mathbf{A}_n, \mathbf{A}_f, \mathbf{A}_d, \mathbf{A}_c]$
Calculate other corresponding matrices according to Eq(8)
for $n = 0 \rightarrow$ number of epoch **do**
Randomly select a batch of sample $\mathcal{D}_b$ from $\mathcal{D}_{train}$, where $b = 1, 2, ..., B$
Reshape the input tensor to fit the LSTM module: $\widetilde{\mathbf{X}}_b^{(t,d)}) \leftarrow Reshape(\widetilde{\mathbf{X}}_b^{(d,t)})$
Obtain the output of RMGC encoder by passing the input tensor through multiple RMGC blocks (Eq(9)): $\widetilde{\mathbf{X}}_b^{RMGC} \leftarrow RMGC(\sigma(\mathbf{D}^{-1/2}(\mathbf{A}+\mathbf{I})\mathbf{D}^{-1/2}\widetilde{\mathbf{X}}_b^{(d,t)}\mathbf{W}_l))$
Flatten the output into a latent vector (Eq(10)): $\mathbf{L}_1 \leftarrow \sigma(\mathbf{W}_1 \cdot Flatten(\widetilde{\mathbf{X}}_b^{RMGC}) + b_1)$
Obtain the output of Spatial LSTM encoder by passing the reshaped input tensor through LSTM module and latent layer(Eq(11): $\mathbf{L}_2 \leftarrow \sigma(\mathbf{W}_2 \cdot Flatten(LSTM(\widetilde{\mathbf{X}}_b^{(t,d)}, \mathbf{W}_{LSTM})) + b_2)$
Estimate the demand by RMGC decoder with joint latent vectors as input (Eq(12),Eq(13)): $\hat{\mathbf{X}}_b^{d,t} \leftarrow RMGC_D(Reshape(\sigma(\mathbf{W}_3 \cdot [\mathbf{L}_1, \mathbf{L}_2] + \mathbf{b}_3)), \mathbf{W}_{RMGC_D})$
Optimize $\mathbf{W}$ by minimizing loss function Eq(14)
end for

5. Experimental results

To evaluate the performance of our ST-ED-RMGC model, we carry out an experiment utilizing the ride-sourcing data from New York City.

Table 2
T-test (P-value) of RMSE of models.

	GBDT	RF	XGB	LASSO	MLP	LSTM	Spatial LSTM	MGC	ED-MGC	RMGC	ST-ED-RMGC
GBDT	–	–	–	–	–	−9.35 (0.00)	116.39 (0.00)	278.63 (0.00)	163.21 (0.00)	95.27 (0.00)	255.58 (0.00)
RF	–	–	–	–	–	20.05 (0.00)	313.17 (0.00)	436.37 (0.00)	267.29 (0.00)	149.27 (0.00)	372.00 (0.00)
XGB	–	–	–	–	–	−9.89 (0.00)	112.78 (0.00)	275.74 (0.00)	161.30 (0.00)	94.28 (0.00)	253.45 (0.00)
LASSO	–	–	–	–	–	2.58 (0.02)	196.24 (0.00)	342.63 (0.00)	205.44 (0.00)	117.18 (0.00)	302.82 (0.00)
MLP	–	–	–	–	–	−9.69 (0.00)	114.13 (0.00)	276.82 (0.00)	162.01 (0.00)	94.65 (0.00)	254.25 (0.00)
LSTM	9.35 (0.00)	−20.05 (0.00)	9.89 (0.00)	−2.58 (0.02)	9.69 (0.00)	–	26.45 (0.00)	60.24 (0.00)	53.36 (0.00)	53.77 (0.00)	71.64 (0.00)
Spatial LSTM	−116.39 (0.00)	−313.17 (0.00)	−112.78 (0.00)	−196.24 (0.00)	−114.13 (0.00)	−26.45 (0.00)	–	144.61 (0.00)	89.86 (0.00)	61.08 (0.00)	160.71 (0.00)
MGC	−278.63 (0.00)	−436.37 (0.00)	−275.74 (0.00)	−342.63 (0.00)	−276.82 (0.00)	−60.24 (0.00)	−144.61 (0.00)	–	−17.22 (0.00)	−0.10 (0.92)	40.18 (0.00)
ED-MGC	−163.21 (0.00)	−267.29 (0.00)	−161.30 (0.00)	−205.44 (0.00)	−162.01 (0.00)	−53.36 (0.00)	−89.86 (0.00)	17.22 (0.00)	–	9.41 (0.00)	48.67 (0.00)
RMGC	−95.27 (0.00)	−149.27 (0.00)	−94.28 (0.00)	−117.18 (0.00)	−94.65 (0.00)	−53.77 (0.00)	−61.08 (0.00)	0.10 (0.92)	−9.41 (0.00)	–	21.11 (0.00)
ST-ED-RMGC	−255.58 (0.00)	−372.00 (0.00)	−253.45 (0.00)	−302.82 (0.00)	−254.25 (0.00)	−71.64 (0.00)	−160.71 (0.00)	−40.18 (0.00)	−48.67 (0.00)	−21.11 (0.00)	–

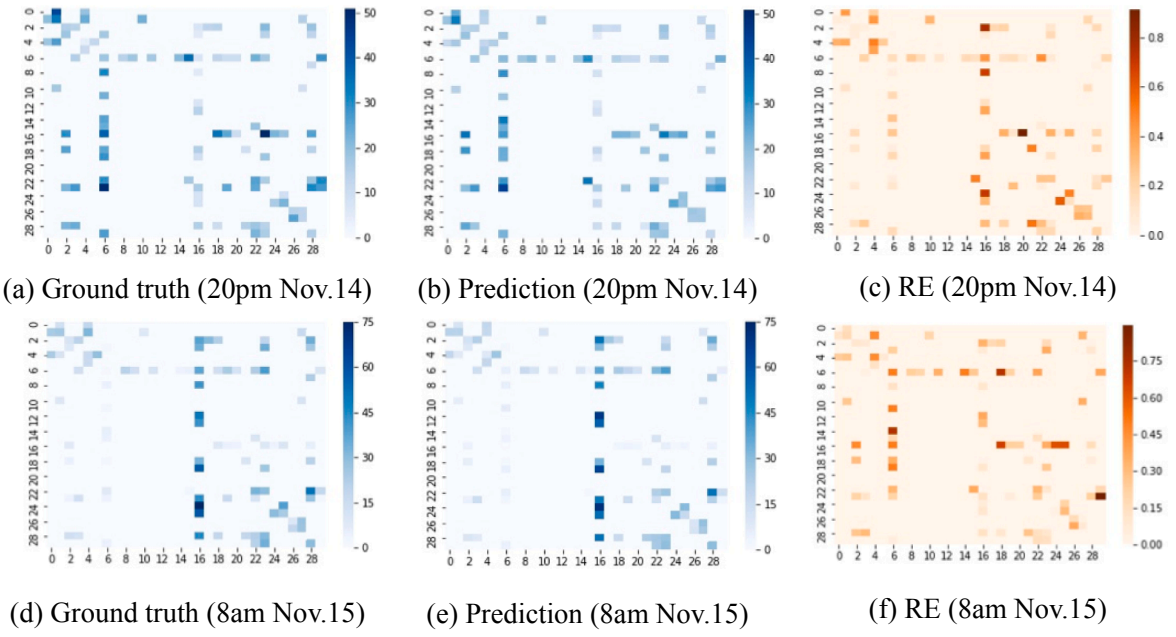


Fig. 8. The prediction result of OD distribution.

5.1. Data description

The location map we used is from the Smart Location Database (SLD), a free data product and service provided by the U.S. EPA Smart Growth Program[1]. The Manhattan area is divided into districts according to administrative zip code, and each of them records the information including area, demographics, employment, and transit. The database of ride-sourcing demand is for-hire vehicle records from January 2018 to April 2019 collected by New York City Taxi & Limousine Commission. Each trip record is associated with attributes including date, time, and taxi zone location ID (same with the ID in the aforementioned location map) of pick-up and drop-off events[2].

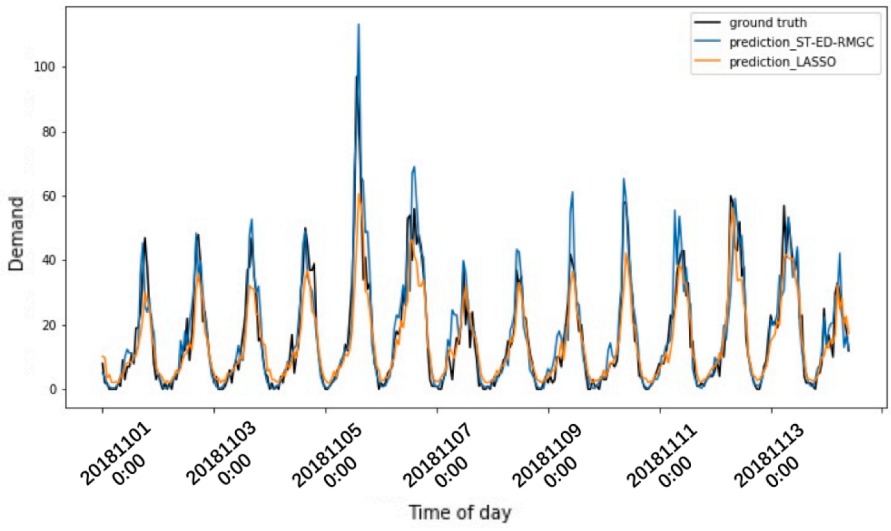
Next, we use an example to demonstrate why OD demand prediction is challenging. Fig. 7 illustrates the OD demand per hour of two representative OD pairs over one month. Firstly, we can see that, although the OD demand has some daily patterns or periodicities, the standard deviations of the OD demand at the same hour across different days are very high. This indicates that daily patterns have strong fluctuations and uncertainties. Second, the daily patterns of different OD pairs are different from each other. As shown in the figure below, the demand patterns of two OD pairs with the same origins (zone No. 43) are quite different from each other. This implies that the underlying spatial relationships between two OD pairs are hard to observe, even for OD pairs with the same origins or destinations. It makes the prediction of OD demand much more difficult than the zone-based demand prediction. Moreover, the OD matrix is very sparse and the demand in most OD pairs is almost zero, which indeed increases the difficulty of precise prediction.

For data split, we use the data from January 8th, 2018 to November 9th, 2018 for training, and the data from December 1st, 2018 to December 21st, 2018 for testing. The data in between, i.e., from November 9th, 2018 to December 1st, 2018 is used for validation.

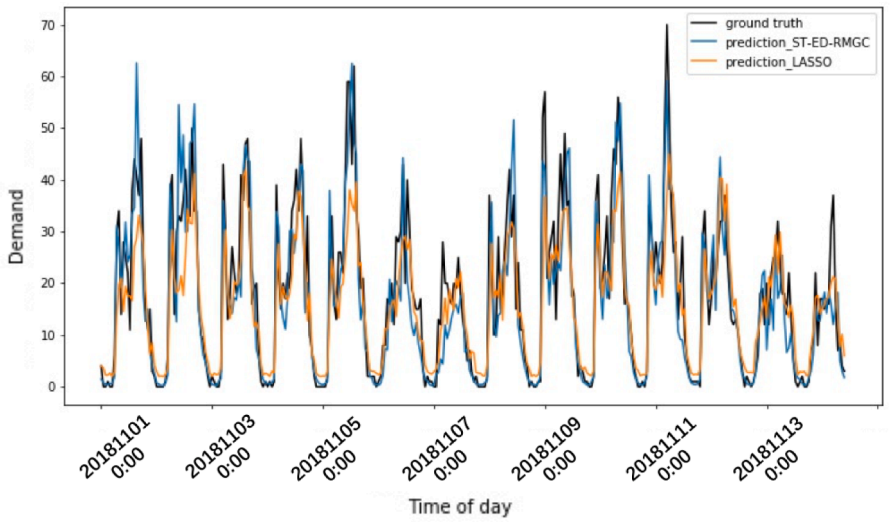
5.2. Model setting

In the encoder, we stack one RMGC convolutional block and one RMGC identity block. The main path of the RMGC convolutional block contains three MGC layers (with 32, 32, 128 hidden units), while the shortcut path contains one MGC layer (with 128 hidden units). The RMGC identity block has a main path containing three MGC layers (with 32, 32, 128 hidden units). On the other hand, the temporal encoder includes two LSTM layers with 128 and 64 hidden units respectively. The outputs of RMGC and LSTM encoders are flattened and then transformed into two latent vectors with dimensions of 900 and 100 respectively. The decoder uses one RMGC convolutional block and then one RMGC identity block, with the same settings as that in the encoder, which is then followed by an MGC layer that generates the estimated OD demand. All the activations in the hidden layers are Relu, while the activation in the output later is a linear function. The optimizer used in the model is Adam with a learning rate of 5e-5 and a decay of 1e-6. During the training phase, we set the batch size to be 32.

[1] https://www.epa.gov/smartgrowth/smart-location-mapping
[2] https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page



(a) OD pair A



(b) OD pair B

Fig. 9. The prediction result of different OD pairs over time.

5.3. Models comparison

In this section, we compare the proposed ST-ED-RMGC with several traditional machine learning models and two graph convolutional networks: an MGC and an RMGC. The models are described below:

- HA: Historical average is one of the most fundamental statistical methods of prediction. We use the average historical OD demand over the past four weeks.
- XGB: XGBoost is an implementation of gradient boosted decision trees designed for speed and performance (Chen and Guestrin, 2016). It makes outstanding performances on a variety of classification and regression predictive modeling problems. The subsampling ratio of the model is 0.8, while the maximum depth of the tree is tuned within 3, 5, and 7, among which 7 is the best parameter.
- MLP: Multi-layer Perception is the basic neural network that contains at least three layers, i.e., input layer, hidden layer, and output layer. It is trained with back-propagation. There are two hidden layers in the tuned model with size 128 and 64. We tune three learning rates: 0.001 0.01 to 0.1, among which the optimal learning rate is 0.001.
- GBDT: Gradient Boosting Decision Tree is an iterative decision tree algorithm with multiple regression decision trees. Three groups of maximum depth (3, 5, and 7) are tuned, and it turns out that a maximum depth of 7 makes the best performance.

- RF: Random forest is the model that is trained by bootstrapped samples of each decision tree. The number of trees is tuned from the values 10, 100, and 200, among which the optimal value is 200.
- LASSO: LASSO also takes historical data as input like normal regression, but with loss function considering least absolute shrinkage and selection operator. The parameter determining the trade-off between empirical errors and mode complexity is tuned from 0.1, 1, and 10. The best parameter is 0.1.
- LSTM: Long short-term memory model is an artificial recurrent neural network with feedback connections. The gates and cells in the model are utilized to regulate and memorize the information of a sequence. We construct a structure with two LSTM layers, with 128 and 64 LSTM cells respectively. We tune the learning rates, among which the optimal one is 0.001.
- Spatial LSTM: a special type of LSTM described in Section 4.2.3. The model has three hidden layers with 256,128 and 64 hidden units respectively and a learning rate of 0.0001.
- MGC network: a multi-graph convolutional network containing three MGC layers (with 256, 128, 64 hidden units). The learning rate is 0.01.
- ED-MGC network: A encoder-decoder structure with multiple MGC layers as encoder and decoder respectively. The number of hidden units of both encoder and decoder is 128.
- RMGC network: a residual multi-graph convolutional network with one RMGC convolutional block and one RMGC identity block. This network has the same structure as the MGC network.
- ST-ED-RMGC network: the parameters are presented in Section 5.2.

The parameters of all the abovementioned models are fine-tuned. We assess the prediction error of the models by three metrics, RMSE (Root Mean Square Error), MAE (Mean Absolute Error), and MAPE (Mean Absolute Percentage Error). Due to the sparseness of OD demand, we only focus on the MAPE result of OD pairs whose demand is more than 1 unit (otherwise, the MAPE becomes infinity if there is a zero demand). The performances of the models are shown in Table 1.

From the table, we can see that GCNs (MGC, RMGC, and ST-ED-RMGC) significantly outperform the traditional machine learning and deep learning methods such as MLP, XGBoost, LSTM. This implies that there exist strong spatial correlations among the OD pairs, which can be well captured by the proposed GCNs through the well-defined OD graphs and adjacent matrices. It can also be found that, although the residual units bring marginal gains to prediction accuracies (comparing RMGC with MGC), the encoder-decoder structure in ST-ED-RMGC further improves prediction performance based on MGC, as seen from the comparison between ED-MGC and MGC. This indicates that the designed encoder-decoder framework can well combine the features learned from the spatial LSTM (temporal correlations) and RMGC (spatial correlations). Table 2. further verifies the significant improvement of the proposed model in prediction performance, as the P-value of the T-statistic comparing with other models are all less than 0.05.

5.4. Prediction results

To present the prediction results, we first examine the OD demand distribution of particular hours. We select the demands of the examined OD pairs that are distributed over 30 districts and build a 30 by 30 matrix to illustrate the prediction results, as shown in Fig. 8. The left figures in each line show the ground truth values, the middle figures represent the predicted values, while the right figures demonstrate the relative prediction error (RE), of each OD pair. Two time slots (20 pm and 8 am) are selected for illustration. It can be shown that the OD demand has a strong imbalance across space and time. For example, people travel to district No. 2 and 6 in the morning and go back to district No. 16 in the evening. Our model well captures these human mobility patterns and thus makes precise predictions.

To have a close look at the result, we select two OD pairs with high variance and different maximum demand volume. Fig. 9 plots their two-week trends of the ground truth values, predicted values of one of the baselines (i.e., LASSO), and predicted values of the ST-ED-RMGC. It can be seen that the temporal patterns of OD demands are different across different days. On some days, there are two peaks; on other days, there is only one peak or a strong peak along with a weak peak. Clearly, the ST-ED-RMGC can better capture the temporal fluctuations across different days, while the baseline is susceptible to over-reacting or under-reacting to the unstable oscillations.

6. Conclusions and future work

In this paper, we study the OD-based ride-sourcing demand prediction problem. Compared with existing OD-based demand prediction approaches, we consider geographical and semantic correlations among OD pairs. That is, several OD graphs are constructed to measure the complex non-Euclidean spatio-temporal pair-wise relationships between OD pairs, from various aspects, including the geographical distances, neighboring relationships, mobility pattern correlations, and functional similarities. We then propose the ST-ED-RMGC model with an encoder-decoder framework, which first encodes the spatial and temporal characteristics with RMGC networks and spatial LSTM networks into a latent vector space, and then utilizes RMGC networks again to decode the latent information for predicting the future OD demand. The proposed model is evaluated with real-world ride-sourcing mobility data in Manhattan, New York City, and is found to outperform the baselines by significant margins. In terms of real-world application, the ride-sourcing platform can predict the OD demand in short term by using the proposed model. Regular training and updating are necessary to better capture the possible change of demand pattern. In this study, we use one-year data for training and testing, which is representative to reflect the periodic demand pattern. Nevertheless, our model can be trained and tested in a larger dataset with a longer period (such as ten years), when more computing resources are available. For future work, more external features like weather,

temperature, and emergencies can be incorporated to improve prediction accuracy. Moreover, our model can be extended to predict abnormal passenger demand due to accidents or public events.

CRediT authorship contribution statement

Jintao Ke: Methodology, Software, Writing - original draft, Writing - review & editing. **Xiaoran Qin:** Formal analysis, Investigation, Writing - original draft. **Hai Yang:** Resources, Writing - review & editing. **Zhengfei Zheng:** Software, Data curation. **Zheng Zhu:** Data curation, Validation. **Jieping Ye:** Conceptualization, Project administration.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The work described in this paper was supported by a NSFC/RGC Joint Research Grant N_HKUST627/18. This work was also supported by the Hong Kong University of Science and Technology - DiDi Chuxing (HKUST-DiDi) Joint Laboratory.

References

Bruna, J., Zaremba, W., Szlam, A., LeCun, Y., 2014. Spectral net-works and locally connected networks on graphs. Proceedings of International Conference on Learning Representations.

Chai, D., Wang, L. and Yang, Q., 2018, November. Bike flow prediction with multi-graph convolutional networks. In Proceedings of the 26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (pp. 397-400). ACM.

Chen, T., Guestrin, C., 2016, August. Xgboost: A scalable tree boosting system. In: Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining (pp. 785-794). ACM.

Defferrard, M., Bresson, X., Vandergheynst, P., 2016. Convolutional neural networks on graphs with fast localized spectral filtering. Adv. Neural Inf. Process. Syst. 3844–3852.

Do, L.N., Vu, H.L., Vo, B.Q., Liu, Z., Phung, D., 2019. An effective spatial-temporal attention based neural network for traffic flow prediction. Transportation Res. Part C: Emerging Technol. 108, 12–28.

Dong, Y., Wang, S., Li, L., Zhang, Z., 2018. An empirical study on travel patterns of internet based ride-sharing. Transportation Res. Part C: Emerging Technol. 86, 1–22.

Gao, H., Wang, Z., Ji, S., 2018. Large-scale learnable graph convolutional networks. In: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. ACM, 2018, pp.1416–1424.

Geng, X., Li, Y., Wang, L., Zhang, L., Yang, Q., Ye, J., Liu, Y., 2019. Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting. 2019 AAAI Conference on Artificial Intelligence (AAAI'19).

Geng, X., Wu, X., Zhang, L., Yang, Q., Liu, Y., Ye, J., 2019b. Multi-Modal Graph Interaction for Multi-Graph Convolution Network in Urban Spatiotemporal Forecasting. arXiv preprint arXiv:1905.11395.

Guo, S., Lin, Y., Feng, N., Song, C., Wan, H., 2019, July. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In: Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 33, pp. 922-929).

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770–778.

Ke, J., Yang, H., Li, X., Wang, H., Ye, J., 2020a. Pricing and equilibrium in on-demand ride-pooling markets. Transport. Res. Part B: Methodol. 139, 411–431.

Ke, J., Yang, H., Zheng, Z., 2020b. On ride-pooling and traffic congestion. Transport. Res. Part B: Methodol. 142, 213–231.

Ke, J., Yang, H., Zheng, H., Chen, X., Jia, Y., Gong, P., Ye, J., 2018. Hexagon-based convolutional neural network for supply-demand forecasting of ride-sourcing services. IEEE Trans. Intell. Transp. Syst. in press.

Ke, J., Zheng, H., Yang, H., Chen, X.M., 2017. Short-term forecasting of passenger demand under on-demand ride services: A spatio-temporal deep learning approach. Transportation Res. Part C: Emerging Technol. 85, 591–608.

Kipf, T.N., Welling, M., 2016. Semi-supervised classification with graph convolutional networks. In: International Conference on Learning Representations (ICLR) 2017.

Levie, R., Monti, F., Bresson, X., Bronstein, M.M., 2017. Cayleynets: Graph convolutional neural networks with complex rational spectral filters. IEEE Trans. Signal Process. 67 (1), 97–109.

Li, R., Wang, S., Zhu, F., Huang, J., 2018. Adaptive graph convolutional neural networks. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 3546–3553.

Li, W., Pu, Z., Li, Y., Ban, X.J., 2019. Characterization of ridesplitting based on observed data: A case study of Chengdu, China. Transportation Res. Part C: Emerging Technol. 100, 330–353.

Li, Y., Yu, R., Shahabi, C., Liu, Y., 2017. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. In: International Conference on Learning Representations (ICLR) 2018.

Lin, L., He, Z., Peeta, S., 2018. Predicting station-level hourly demand in a large-scale bike-sharing network: A graph convolutional neural network approach. Transportation Res. Part C: Emerging Technol. 97, 258–276.

Liu, L., Qiu, Z., Li, G., Wang, Q., Ouyang, W., Lin, L., 2019. Contextualized spatial-temporal network for taxi origin-destination demand prediction. IEEE Trans. Intell. Transp. Syst. in press.

Liu, Y., Li, Y., 2017. Pricing scheme design of ridesharing program in morning commute problem. Transportation Res. Part C: Emerging Technol. 79, 156–177.

Monti, F., Boscaini, D., Masci, J., Rodola, E., Svoboda, J., Bronstein, M.M., 2017. Geometric deep learning on graphs and manifolds using mixture model cnns. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 5115–5124.

Monti, F., Bronstein, M., Bresson, X., 2017b. Geometric matrix completion with recurrent multi-graph neural networks. In: Advances in Neural Information Processing Systems, pp. 3697–3707.

Nourinejad, M., Roorda, M.J., 2016. Agent based model for dynamic ridesharing. Transportation Res. Part C: Emerging Technol. 64, 117–132.

Su, Q., Wang, D.Z., 2019. Morning commute problem with supply management considering parking and ride-sourcing. Transportation Research Part C: Emerging Technologies, in press.

Sun, J., Zhang, J., Li, Q., Yi, X., Zheng, Y., 2019. Predicting Citywide Crowd Flows in Irregular Regions Using Multi-View Graph Convolutional Networks. arXiv preprint arXiv:1903.07789.

Wang, B., Luo, X., Zhang, F., Yuan, B., Bertozzi, A.L., Brantingham, P.J., 2018. Graph-based deep modeling and real time forecasting of sparse spatio-temporal data. arXiv preprint arXiv:1804.00684.

Wang, Y., Yin, H., Chen, H., Wo, T., Xu, J., Zheng, K., 2019, July. 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 (pp. 1227-1235). ACM.

Wu, L., Sun, P., Hong, R., Fu, Y., Wang, X., Wang, M., 2018. Socialgcn: An efficient graph convolutional network based model for social recommendation. arXiv preprint arXiv:1811.02815.

Schaller, B., 2018. The new automobility: Lyft, Uber and the future of American cities. Transport. Res. Board.

Wu, Y., Tan, H., Qin, L., Ran, B., Jiang, Z., 2018b. A hybrid deep learning based traffic flow prediction method and its understanding. Transportation Res. Part C: Emerging Technol. 90, 166–180.

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., Yu, P.S., 2019. A comprehensive survey on graph neural networks. arXiv preprint arXiv:1901.00596.

Xiong, X., Ozbay, K., Jin, L. and Feng, C., 2019. Dynamic Origin-Destination Matrix Prediction with Line Graph Neural Networks and Kalman Filter. arXiv preprint arXiv:1905.00406.

Yang, S., Ma, W., Pi, X., Qian, S., 2019. A deep learning approach to real-time parking occupancy prediction in transportation networks incorporating multiple spatio-temporal data sources. Transportation Res. Part C: Emerging Technol. 107, 248–265.

Yao, H., Tang, X., Wei, H., Zheng, G., Yu, Y., Li, Z., 2018a. Modeling spatial-temporal dynamics for traffic prediction. arXiv preprint arXiv:1803.01254.

Yao, H., Wu, F., Ke, J., Tang, X., Jia, Y., Lu, S., Gong, P., Ye, J., Li, Z., 2018. Deep multi-view spatial-temporalnetwork for taxi demand prediction. 2018 AAAI Conference on Artificial Intelligence (AAAI'18).

Ying, R., He, R., Chen, K., Eksombatchai, P., Hamilton, W.L., Leskovec, J., 2018. July. Graph convolutional neural networks for web-scale recommender systems. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 974–983.

Yu, B., Li, M., Zhang, J., Zhu, Z., 2019. 3D Graph Convolutional Networks with Temporal Graphs: A Spatial Information Free Framework for Traffic Forecasting. arXiv preprint arXiv:1903.00919.

Zhang, J., Zheng, Y., Qi, D., 2017. February. Deep spatio-temporal residual networks for citywide crowd flows prediction. Thirty-First AAAI Conference on Artificial Intelligence.

Zhang, J., Zheng, Y., Sun, J., Qi, D., 2019a. Flow prediction in spatio-temporal networks based on multitask deep learning. IEEE Trans. Knowl. Data Eng. in press.

Zhang, K., Liu, Z., Zheng, L., 2019. Short-Term Prediction of Passenger Demand in Multi-Zone Level: Temporal Convolutional Neural Network With Multi-Task Learning. IEEE Transactions on Intelligent Transportation Systems, in press.

Zha, L., Yin, Y., Xu, Z., 2018. Geometric matching and spatial pricing in ride-sourcing markets. Transport. Res. Part C: Emerg. Technol. 92, 58–75.

Zhang, Y., Cheng, T., Ren, Y., 2019c. A graph deep learning method for short-term traffic forecasting on large road networks. Comput.-Aided Civ. Infrastruct. Eng. in press.

Zhang, Z., Li, M., Lin, X., Wang, Y., He, F., 2019d. Multistep speed prediction on traffic networks: A deep learning approach considering spatio-temporal dependencies. Transportation Res. Part C: Emerging Technol. 105, 297–322.

Zhu, Z., Chen, X., Zhang, X., Zhang, L., 2018. Probabilistic data fusion for short-term traffic prediction with semiparametric density ratio model. IEEE Trans. Intell. Transp. Syst. 20 (7), 2459–2469.

Zhu, Z., Peng, B., Xiong, C., Zhang, L., 2016. Short-term traffic flow prediction with linear conditional Gaussian Bayesian network. J. Adv. Transportation 50 (6), 1111–1123.

Zhu, Z., Tang, L., Xiong, C., Chen, X., Zhang, L., 2019. The conditional probability of travel speed and its application to short-term prediction. Transportmetrica B: Transport Dyn. 7 (1), 684–706.