

A multi-task matrix factorized graph neural network for co-prediction of zone-based and OD-based ride-hailing demand

Siyuan Feng, Jintao Ke *, Hai Yang and Jieping Ye, *Fellow, IEEE*

Abstract—Ride-hailing service has witnessed a dramatic growth over the past decade but meanwhile raised various challenging issues, one of which is how to provide a timely and accurate short-term prediction of supply and demand. While the predictions for zone-based demand have been extensively studied, much less efforts have been paid to the predictions for origin-destination (OD) based demand (namely, demand originating from one zone to another). However, OD-based demand prediction is even more important and worth further explorations, since it provides more elaborate trip information in the near future as reference for fine-grained operations, such as the routing and matching of shared ride-hailing services that pick up and drop off two or more passengers in each ride. Simultaneous prediction of both zone-based and OD-based demand can be an interesting and practical problem for the ride-hailing platforms. To address the issue, we propose a multi-task matrix factorized graph neural network (MT-MF-GCN), which consists of two major components: (1) a GCN (graph convolutional network) basic module that captures the spatial correlations among zones via mixture-model graph convolutional (MGC) network, and (2) a matrix factorization module for multi-task predictions of zone-based and OD-based demand. By evaluations on the real-world on-demand data in Manhattan and Haikou, we show that the proposed model outperforms the state-of-the-art baseline methods in both zone- and OD-based predictions.

Index Terms—ride-hailing, OD-based prediction, mixture-model graph convolutional network, matrix factorization, deep multi-task learning

I. INTRODUCTION

RIDE-HAILING service, serving as a novel option of transportation market, has fast-growing user groups around the world and boosts the emergence of a variety of corresponding operation platforms like Didi, Uber and Lyft. For example, just a single platform like Didi can generate millions of daily ride-hailing demand in Beijing [1], showing the size and popularity of this new market. Utilizing the historical trip records, the ride-hailing platform can predict demand of future trips to assist in dynamic operation strategies, such as surge pricing, vacant vehicle re-positioning, and ride-pooling, etc. Two types of demand can be of interest: zone demand and OD demand. The former one represents

cumulative demand that depart from or arrive at each zone, regarded as outflow zone-based demand (outflow demand) or inflow zone-based demand (inflow demand), respectively. The OD-based demand represents the number of trips from a given zone to another. Zone-based demand depicts a macro-view of spatial distribution of ride-hailing requests, while OD-based demand more precisely shows the quantity of trips with different combinations of origin and destination. The ride-sourcing companies have to make multiple decisions, including vehicle re-positioning, route planning of vehicles and pricing for both non-ride-pooling and ride-pooling passengers in real time. Some tasks need zone-based demand predictions, for example, the platform wants to dispatch an idling vehicle to a region with high predicted outflow demand. Some tasks require OD demand predictions, for example, the platform may proactively raise the trip fare (surge pricing) for an OD with a predicted excess demand (compared to supply level). Moreover, some tasks need both zone-based and OD-based demand predictions, for example, as the platform plans a route for a vehicle who already picks up the first ride-pooling passenger, it needs to determine a route for the vehicle such that (1) there are high outflow demands along this route (zone demand prediction is required); (2) there are demands with similar destinations as the first passenger (OD demand prediction is required). As the platform chooses a suitable route for the vehicle, there is a higher probability that this vehicle can serve one or more additional passengers on the way to deliver the first passenger, which helps to improve vehicle utilization and platform revenue. To summarize, since the platform has to make different decisions, there is a strong need for the platform to obtain accurate zone-based and OD-based demand predictions to support these decisions. Note that zone attraction demand is not simply equal to the summation of OD demand over origin zones in a given time interval. With this difference, the predictions of zone demands and OD demand should be viewed and treated as different tasks. This is because the system is dynamic and non-stationary and thus the trips generated in an origin zone may arrive at a destination zone in a different time interval. Since most of the existing models only predict zone-based ([2], [3], [4]) or OD-based demand ([5], [6]), a multi-task prediction model structure with shared embedding module and separate decoding modules for different tasks can be adopted. Such a multi-task model usually possesses fewer model parameters compared to the sum of parameters of the three single task models for the prediction of each type of demand. For each of the single-

S. Feng, and H. Yang are with the Department of Civil and Environmental Engineering, The Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong, China.

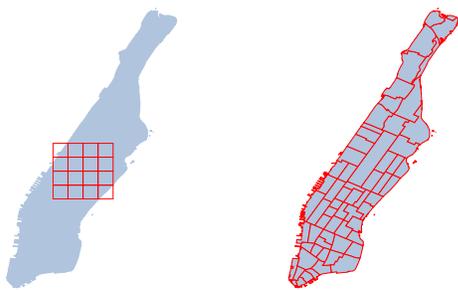
J. Ke is with the Department of Logistics and Maritime Studies, Hong Kong Polytechnic University, Hung Hom, Hong Kong, China.

J. Ye is with the Department of Computational Medicine and Bioinformatics, University of Michigan, Ann Arbor, MI, United States.

* J. Ke is the corresponding author, E-mail: jke@connect.ust.hk

task models for the three tasks, there are some similar model structures, especially in the encoder part. However, these similar structures (for different tasks) can be merged into one structure, where the parameters are co-trained for the three tasks. The required computation resources to store and train several similar modules within the single-task models are then reduced, since there is only one integrated module for the proposed multi-task model.

In addition to the co-prediction problem, a practical challenge is to capture the spatial dependence among the demands in multiple irregular zones. Different types of dependencies should be well considered. The first one is based on geographical closeness. That is, some adjacent zones may share highly similar demand patterns. For example, two adjacent business zones in Lower Manhattan may both have intensive inflow in morning peak and high outflow in evening peak. On the other hand, zones which are distant from each other may also have some latent correlations in demand. For instance, residential areas in both Lower and Upper Manhattan may have high morning outflow and evening inflow, although they are geographically remote from each other. In most previous short-term predictions, the studied area is first partitioned into regular zones (in Euclidean domain) like squares, as shown in Fig. 1a as an instance of Manhattan, based on which the standard CNN (Convolutional Neural Network)-based method (e.g. 2D/3D CNN [3] [7] [6], Conv-LSTM [2] [3]) can be adopted to capture the spatial dependencies. However, this simple zone partition can not well consider the heterogeneity of the zones in terms of administrative and functional properties. A better way is to partition the examined area into multiple regions according to administrative definitions, such as zip codes, as shown in Fig. 1b. The CNN-based methods are no longer applicable to these irregular regions, since they rely on a standard image-like input data structure.



(a) Euclidean domain (b) non-Euclidean domain

Fig. 1: Different partitions of Manhattan. The corresponding zones are represented by the red box.

To tackle the aforementioned critical issues, we propose the multi-task matrix factorized graph neural network (MT-MF-GCN). The model contains two major parts: the GCN (graph convolutional network) basic module for capturing spatial-temporal features in non-Euclidean domains, and the matrix factorization module for multi-task learning. GCN basic

module serves as an encoder for graph embedding. In this module, the aim is to capture demand patterns and represent them by a set of hidden feature vectors for each zone based on historical data. To form the representations, we employ GCN-based structure to capture information from a zone itself and its neighbors. Two correlation matrices are first established, including a distance matrix and a function similarity matrix. The first one measures the geographical proximity between each pair of two zones, and the second one characterizes the functional similarities among zones. We then develop four graphs and corresponding neighborhoods: one graph based on geographical adjacency, and three semantic graphs respectively based on the two proposed correlation matrices and real-time commuting patterns (say OD graph for simplicity). The last semantic graph is interesting since it employs the correlation between OD demand and zone demands for the improvement in prediction accuracy. In general, the larger the OD demand, the larger the inflow/outflow zone demand. In the real-time constructed OD graph, two zones become neighbors when the OD demand for this OD pair is large enough. Since the amount of OD demand and zone demand is correlated with each other, two neighbor zones in a OD graph, although with possibly low similarity in function or geographical location, can also share high correlations in their zone demands. Capturing this correlation helps to employ more complete information to generate hidden representations and make predictions for zone demands. With all the graphs constructed, four groups of mixture-model graph convolutional (MGC) networks are respectively built to capture the non-Euclidean spatial dependencies.

The learned hidden representations in the GCN basic module are then fed into the matrix factorization module. In this part, the representations are decoded into estimated demands for inflow, outflow and OD flow. The prediction of each type of demand is given by the output of a stack of sub-predictors, which are called matrix factorization (MF) layers. Through simple decomposition process, the MF layer can serve as region-specific decoders, which sufficiently considers the uniqueness of the mobility patterns in different zones. Moreover, the parameters required to learn are reduced through the decomposition, providing a smaller-size model structure. In addition, we also design a special decoder placed after the final one of MF layers for OD-based prediction. This decoder structure in our proposed model is in the same spirit of the classic OD-based demand function, in which the OD demand is given by the product of the minimum travel time and its associated origin-based and destination-based characteristics. However, the decoder in our model is different from the classical model in the following aspects: (1) the minimum travel time in the classical model is replaced by a transmission matrix learned by the network; (2) the pre-determined properties of origin and destination in the classical model are substituted by the hidden representations generated by the MF layers. In a word, although our model shares the same formulation with the classical OD-estimation model, all the inputs are hidden parameters learned by the neural network, rather than human investigations. The representation power of the deep neural networks has the potentials to better

capture the demand patterns from historical data, and thus offers a more accurate prediction.

Finally, the outputs of the predictors for each task are combined in a single loss function for an end-to-end learning. Evaluated on the real-world ride-hailing data in Manhattan, New York, and Haikou, China, the proposed framework demonstrates the superiority against the baseline approaches. In summary, this paper makes the following contributions:

- We propose a multi-task matrix factorized graph neural network (MT-MF-GCN) to achieve the co-prediction of inflow, outflow and OD-based ride-hailing demand within a single model framework.
- We design a GCN-based embedding module with multi-semantic graphs and a mixture-model graph convolutional network to capture non-Euclidean spatial dependencies and generate hidden representations for each zone.
- We design a matrix factorization module for multi-task learning. Region-specific decoders via matrix factorization are utilized to decode the hidden representations, and make predictions by effectively capturing the dependence among zones. A special decoder refined on classic demand estimation is proposed for the prediction of OD-based demand.
- We conduct extensive experiments on the real-world for-hire ride-hailing dataset in Manhattan and Haikou, and demonstrate that the proposed model outperforms the state-of-the-art baselines.
- The proposed model structure can be also utilized for other multi-task applications in future, such as predictions for customer demand for different travel modes, joint predictions for multiple traffic state variables (speed, density, etc.) in a road network.

The remainder of the paper is organized as follows. In section II, we provide a literature review on classic and recent approaches in traffic demand prediction. In section III, some preliminaries of the study are provided, including basic definitions and a formal description of the research problem. The details of the model framework and numerical experiments will be respectively presented in section IV and section V. Finally, in section VI, we provide conclusions and directions for future research.

II. LITERATURE REVIEW

The prediction of zone-based and OD-based ride-hailing demand can be grouped into the family of spatial-temporal forecasting problem. This section conducts an extensive literature review on the classic and state-of-the-art prediction approaches of traffic demand, which provides insights for the current study. The sub-sections of the review include classic and deep learning based prediction methods for traffic demand, matrix factorization technology and their applications in demand forecasting, and multi-task learning for transportation.

A. Prediction approaches

Early studies of both zone-based and OD-based demand prediction depend on the regional properties subjectively decided and surveyed by researchers. For instance, OD trip rates

are influenced by the level of service and properties of origin and destination. When congestion increases on the routes for an OD pair, travelers may change the travel mode or even cancel the trip, which affects the OD-based traffic demand. The properties of origin and destination, including population size, income distribution, vehicle ownership, employment intensity and business properties, are also important factors for the intensity of each OD-based demand. Based on this knowledge, the classic OD-based demand function in [8] is given by

$$m_{i,j} = A_i B_j f(c_{i,j}) \quad (1)$$

where $m_{i,j}$ is trip rate between zone i and j in a studied transportation network, A_i and B_j are parameters associated with the natural properties of origin and destination, $f(\cdot)$ is a function with input $c_{i,j}$, which represents the minimum travel cost in the form of time or distance. For most classic studies in transportation network theory, A_i , B_j and $f(\cdot)$ are usually assumed known and fixed, and the only parameter $c_{i,j}$ is calculated with the equilibrium theory. However, the pre-set region properties can be incomplete and biased, and even in some cases, not available or quite time-consuming for collection. This is a common challenge in other fields involving feature engineering.

Based on the classic methods, a potential improvement comes from the employment of historical data. To better formulate predictions from the past mobility records, a series of methods have been proposed, including the auto-regressive integrated moving average (ARIMA) models ([9], [10], [11], [12]), Bayesian inference approaches ([13], [14]), Kalman filtering model [15], local regression model [16], and neural network models ([17], [18]). Most of these work construct the past traffic information as a time series, without sufficient focus on spatial correlations. To overcome the issue, some studies utilize traffic demand or other states collected from upstream [19] or nearby states [20] as supplement to the estimation for the downstream or center node. However, these approaches usually capture the information from only one or small-size neighbor nodes, which may ignore the knowledge in geographically remote regions.

In recent years, machine learning or deep learning based methods have been widely applied in various areas of transportation, such as pedestrian behavior study ([21], [22]), passenger flow planning [23] and ride-hailing demand prediction ([2], [7], [6]). For prediction task, generally, convolutional neural network (CNN) is utilized to capture spatial correlations, while the recurrent neural networks (RNNs) and their variants long short term memory (LSTM) are employed for mining sequential properties. Most of the studies explore different combinations of these two groups of deep learning paradigms to capture spatial-temporal information in a single model framework. In [2], Ke et al. propose a short-term forecasting model for ride-hailing demand which first combines CNN and LSTM in traffic demand prediction task. In [7], Ye et al. construct multiple auto-encoders to predict demand of pick-up and drop-off service for taxi and bikes. In [6], Liu et al. utilize LSTM to encode the input feature and node-wise correlation matrices to refine the middle representations before final output.

The majority of these studies choose to first partition the studied city or area into regular zones like squares or hexagons. The prediction is then made for demand inside these artificially constructed zones. With such partitions on the Euclidean domains, CNN-based deep learning approaches, including 2D/3D CNN and Conv-LSTM, have been extensively employed to capture spatial correlations and make predictions. However, a more effective idea is to partition geographic zones into graphs. The demand is then estimated via aggregating information from neighborhood on these graphs. The complexity of graph data has generated significant challenges on existing deep learning approaches. The graph data is irregular. Each node in a graph has a different number of neighbors, causing some important operations (e.g., convolutions), which are easy to compute in the Euclidean domain, not directly applicable to the graph domain anymore with standard CNN [24]. To enable this complicated process, a model structure called graph convolutional network (GCN) has been developed recently. In Fig. 2, we show the difference and similarity between a standard 2D CNN and a type of graph convolution. GCN-based methods are also explored in transportation state prediction by past studies. In [25], Wang et al. refine GraphSAGE [26] with fixed and pre-defined weights to aggregate neighborhood information for ride-hailing demand prediction. In [4], contextual gated RNN is combined with GCN to forecast zone-based demand. In [5], Ke et al. capture correlations between each OD pair and add residual mechanism to its GCN module. Compared to [25], all the edge weights of GCN module in the proposed model are obtained via training instead of being artificially pre-defined, providing better flexibility for different problems. Compared to [4] and [5], while they only have a single kind of edge weight for each adjacency matrix, the proposed model employs a mixed weight via mixture-model GCN, in order to capture the correlations between different edge weights. About this point, a more detailed discussion is provided in section IV C. In addition, matrix factorization module is applied as a region-specific decoder, while the other GCN researches either lack this part or simply use linear transformation layers.

B. Matrix factorization applications

Matrix factorization technology has been a classic approach in image compression [27], recommendation systems ([28], [29]), classification ([30], [31]) and prediction task ([32], [33]). In matrix factorization process, the studied matrix is decomposed into multiplication of small-size matrices, which reduce the total number of model parameters and thus effectively save the computation resources. The advantages of matrix factorization enable model with more virtual parameters but fewer actual ones. The complexity of such models can thus be increased for the solution of complicated tasks. In [34], a supervised matrix factorization approach is proposed to learn latent features for link prediction on graphs. In [35], Chen et al. construct a matrix factorization framework to estimate accident risk with heterogeneous data. For deep learning setting, Pan et al. [36] combine matrix factorization layers with neural network models to predict urban flows. In [35], the matrix

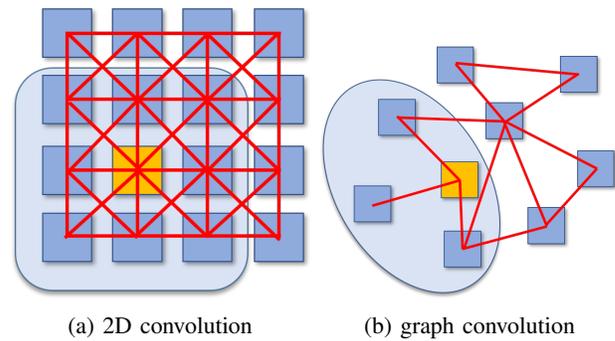


Fig. 2: 2D convolution and graph convolution. In 2D convolution for Euclidean domain, the neighbors of each zone are ordered and fixed in size. The adjacency is represented with red edges. The 2D convolution takes a weighted average over the yellow zone and its neighbors. In graph convolution for non-Euclidean domain, the neighbors of each zone have no orders and vary in size. The graph convolution takes average over the yellow zone and its neighborhood to formulate a hidden representation.

factorization module serves as both embedding and decoding modules, while the proposed model only utilizes it as decoder modules for multi-tasks. The separation of embedding and decoding module can improve the model complexity and thus the prediction accuracy for complicated tasks. Meanwhile, the previous researches either focus on tasks outside transportation demand estimation (bibliographic networks or computational biology for [34], accident risks for [35]), or only single-task prediction (such as [36]). In comparison, the proposed model employs the technology to formulate smaller-size region-specific decoders for multi-task traffic demand predictions.

C. Multi-task learning based approaches

Multi-task learning aims to share information among different tasks that have similar properties. Most of studies in deep learning setting employ multiple shared layers or part of parameters to direct their models towards capturing common knowledge. Separate decoding modules are then employed to restore unique predictions for each task. For transportation, there is also a variety of studies for multi-task learning and predictions. In [7], Ye et al. integrate information from taxi and bike into a single temporal-knowledge mining module. In [37], Kuang et al. utilize a 3D residual network to fuse representations of pick-up and drop-off modes of taxi demand data. In [38], Geng et al. employ grouped GCN and multilinear relationship to realize modality interaction among various hidden features.

However, limited efforts have been made to combine predictions of inflow, outflow, and OD-based demand together in a single model. In our proposed multi-task prediction model, shared embedding module is used to generate hidden representations for ride-hailing demands, and separate modules with region-specific decoders are employed to restore predictions for different tasks.

III. PRELIMINARIES

A. Basic definitions

We first provide several fundamental definitions for the formulation of multi-task ride-hailing demand prediction problem.

Definition 1. (Graph) The utilized graph is defined as $G(V, E, A)$, where V is the set of zones, E represents the set of edges with different weights, and $A \in \mathbb{R}^{|V| \times |V|}$ denotes the adjacency matrices. Multiple graphs can be constructed via different setting of edge weights and adjacency matrices.

Definition 2. (Inflow and outflow vectors) The total time period is portioned into a discrete series of time slots with a constant interval. We use T to represent the final time slot of the time sequence and t to denote a certain middle time slot. Based on the time slot definition, let $x_{i,in}^t$ and $x_{i,out}^t$, $\forall i \in \{1, \dots, N\}$ and $\forall t \in \{1, \dots, T\}$, denote the inflow and outflow demand of zone i during time slot t , where N is the number of zones. The inflow and outflow demand vectors are denoted by \mathbf{X}_{in}^t and $\mathbf{X}_{out}^t \in \mathbb{N}^N$, $\forall t \in \{1, \dots, T\}$, whose vector entries are $x_{i,in}^t$ and $x_{i,out}^t$ respectively.

Definition 3. (OD matrix) We use $\mathbf{M}^t \in \mathbb{N}^{N \times N}$, $\forall t \in \{1, \dots, T\}$, as the OD demand matrix for each time interval. Each entry of \mathbf{M}^t , $m_{i,j}^t$, represents the number of ride-hailing demands with origin zone i and destination zone j .

B. Research problem

Definition 4. (Zone-based demand prediction) For a time slot t , given a historical inflow or outflow demand sequence $[\mathbf{X}^{t-k}, \dots, \mathbf{X}^t]$, the zone-based demand prediction problem is to predict the demand vector in the next time slot $t+1$, that is, \mathbf{X}^{t+1} .

Definition 5. (OD-based demand prediction) Given a time slot t and the past OD demand matrix sequence $[\mathbf{M}^{t-k}, \dots, \mathbf{M}^t]$, the OD-based demand prediction problem is to forecast the OD matrix in the next time slot $t+1$, that is, \mathbf{M}^{t+1} .

Problem 1. (Multi-task ride-hailing demand prediction) The problem is a combination of zone-based demand prediction problem and OD-based demand prediction problem, where historical demand sequence $[\mathbf{X}_{in}^{t-k}, \dots, \mathbf{X}_{in}^t]$, $[\mathbf{X}_{out}^{t-k}, \dots, \mathbf{X}_{out}^t]$, and $[\mathbf{M}^{t-k}, \dots, \mathbf{M}^t]$ are given, and the task is to predict \mathbf{X}_{in}^{t+1} , \mathbf{X}_{out}^{t+1} and \mathbf{M}^{t+1} simultaneously within a single model.

IV. MODEL FRAMEWORK

A. Overview

In this section, we propose a unified model to solve the multi-task prediction problem of inflow, outflow and OD-based demand. The overview of the model is given in Fig. 3. First, the input features selected based on temporal dependencies is fed into the GCN basic module. The module consists of four sub-modules, which are respectively constructed on different neighborhoods with the stack of mixture-model graph convolutional (MGC) network to capture spatial dependencies. The outputs of the sub-modules are then fused together and fed into the matrix factorization module for multi-task learning. The fused representation is separately decoded by three groups

of matrix factorization (MF) layers. The output of MF layers will be employed to obtain the short-term estimation of inflow, outflow and OD-based ride-hailing demand. In the following part, we will first introduce the construction of correlation matrices and semantic neighborhoods, then specifications of GCN basic module, and finally the details of the matrix factorization module and loss function for multi-task learning.

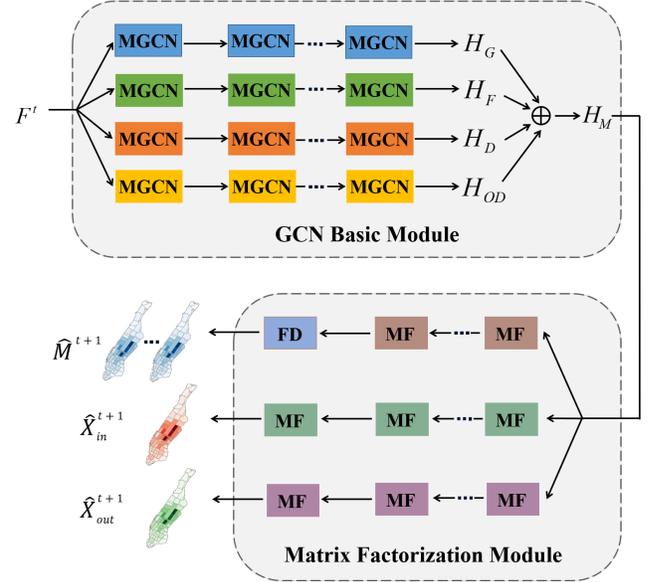


Fig. 3: Model Framework. F^t is the input feature, MGCNs in different colors represent mixture-model GCNs respectively constructed for one geographical and three semantic neighborhoods, with output matrices H_G , H_F , H_D , H_{OD} . After the matrix concatenation operation, MF in different colors represent matrix factorization layers for the three tasks, and FD is a transmission matrix. \hat{M}^{t+1} , \hat{X}_{in}^{t+1} , \hat{X}_{out}^{t+1} are the predicted demand of OD flow, inflow and outflow.

B. Correlation matrices and semantic neighborhoods

The GCN basic module is built on fine identification and definition of multiple types of neighborhoods to efficiently capture spatial dependencies on different levels. The neighborhoods are represented by adjacency matrices, constructed based on geographical closeness, functionality and mobility patterns. Geographical closeness is considered by two kinds of adjacency matrices: geographical adjacency matrix and distance adjacency matrix. In the former one, if two zones are geographically adjacent to each other, they are viewed as neighbors. In the latter one, if the distance between two zones are sufficiently small, they can become neighbors, although they may not be directly adjacent to each other. Both matrices follow the same logic: the closer the two zones to each other, the higher the possibility to share similar historical mobility patterns. However, some distant zones with high similarity in functionalities or frequent mobility interaction can also share similar demand patterns, and thus provide useful and unique information for the message transmission. We use the term "semantic neighbor" to differentiate such closeness within

zones from the regular geographical adjacency (where zones become neighbors only when they are directly geographically adjacent to each other). The neighborhoods formed by semantic neighbors are called semantic neighborhoods, and special adjacency matrices are designed and applied to describe them. For the convenience of presentation, neighborhoods corresponding to distance adjacency matrix are also viewed semantic. Specifically, we first build two correlation matrices, C_D and C_F , to represent the relations in distance and function similarity, which are given by,

$$[C_D]_{i,j} = [dis(i,j)]^{-1} \quad (2)$$

$$[C_F]_{i,j} = \|\mathbf{v}_i - \mathbf{v}_j\|^{-1} \quad (3)$$

where $dis(i,j)$ is the straight-line distance between the centroid of zone i and j , and \mathbf{v}_i and \mathbf{v}_j are the vectors representing features related to function similarity, such as private car ownership, household density, population and employment structure, station distribution and cumulative lengths of road network per square kilometers. In another word, C_D shows the geographical closeness between each pair of two zones, while C_F presents the similarity in intrinsic properties within zones.

Based on the correlation matrices and region-level commuting history, we then construct one geographical adjacency matrix and three semantic adjacency matrices as the representations of different neighborhoods for each zone. The geographical adjacency matrix is given by,

$$[A_G]_{i,j} = \begin{cases} 1, & \text{if zone } i \text{ and } j \text{ are adjacent} \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

which is the most common and natural one for spatial graph in transportation study (as in [4], [5]). However, similarities also exist in the zones which are not geographically adjacent to the studied one. The information stored in these hidden neighbors should not be ignored in the formulation of zone-based representations. To capture these indirect relationships based on functionality, distance and commuting patterns, multiple stacks of MGCN layers are constructed with the corresponding semantic adjacency matrices, as shown in GCN Basic Module in Fig. 3. For this goal, we first establish three auxiliary matrices \overline{A}_D , \overline{A}_F and \overline{A}_{OD} as the following:

$$[\overline{A}_D]_{i,j} = \begin{cases} 1, & \text{if } [C_D]_{i,j} \in S_{D,i} \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

$$[\overline{A}_F]_{i,j} = \begin{cases} 1, & \text{if } [C_F]_{i,j} \in S_{F,i} \\ 0, & \text{otherwise} \end{cases} \quad (6)$$

$$[\overline{A}_{OD}]_{i,j} = \begin{cases} 1, & \text{if } n_{i,j} \geq r_{od} \\ 0, & \text{otherwise} \end{cases} \quad (7)$$

where $[C_D]_{i,j}$ and $[C_F]_{i,j}$ are matrix entries defined in equation (2) and (3), $S_{D,i}$ and $S_{F,i}$ are the sets of the largest r_d^{th} and r_f^{th} entries of row vectors $[C_D]_i$ and $[C_F]_i$ respectively, $n_{i,j}$ is the cumulative number of trip requests from i to j during critical time slots. Three thresholds r_d , r_f and r_{od} are defined to control the size of semantic neighborhoods. In eq.

(5), each matrix entry $[\overline{A}_D]_{i,j}$ is 1 if the distance between zone i and zone j is small, and otherwise zero. In eq. (6), $[\overline{A}_F]_{i,j}$ is 1 if zone i and zone j have high similarity in functionality, and otherwise zero. In eq. (7), $[\overline{A}_{OD}]_{i,j}$ is 1 if there is enough OD flow between zone i and zone j in the past time intervals, and otherwise zero. To keep the symmetry of each adjacency matrix, we refine the auxiliary ones as follows:

$$[\mathbf{A}]_{i,j} = \begin{cases} 1, & \text{if } [\overline{\mathbf{A}}]_{i,j} = 1 \text{ or } [\overline{\mathbf{A}}]_{j,i} = 1 \\ 0, & \text{otherwise} \end{cases} \quad (8)$$

where $[\mathbf{A}]_{i,j}$, $[\overline{\mathbf{A}}]_{i,j}$ and $[\overline{\mathbf{A}}]_{j,i}$ here are applicable to the three semantic cases.

With different definitions of adjacency, both the neighborhood structure and the learned feature itself can be quite diversified, and it is important to integrate them into the single model framework for information completeness. In next section, we will show how this process works.

C. GCN basic module

In this module, mixture-model GCN (MGC) is utilized to formulate hidden representations based on different neighborhoods. MGC can be categorized into the family of spatial-based GCN, which has been widely used in applications including node and graph classifications and predictions in the studies of social network, chemistry, e-commerce, etc. Generally, the spatial-based GCN aggregates the feature vector of a center node and its neighbors together as the learned hidden representations of the central one. In this process, the appropriate choice of aggregation method plays the key role. A fundamental approach is proposed in [26], with aggregation in the following form:

$$\mathbf{h}'_i = \sigma(\mathbf{W} \cdot \text{mean}_{j \in Ne(i)}(\mathbf{h}_j)) \quad (9)$$

where \mathbf{h}'_i is the new hidden representation of zone i , \mathbf{W} is a linear transformation matrix to learn, $Ne(i)$ is the index set containing zone i and its neighbors, \mathbf{h}_j is the initial or learned features fed into GCN. This aggregation approach simply takes the average of the local information without differentiation of importance, which may negatively affect the accuracy and stability of learning. Some later studies improve the naive one in equation (9) via imposing pre-defined edge-wise weights before training. To further increase the model complexity, the fixed edge weights can be substituted by weighting functions with hyper-parameters learned by the model itself [39]. Following this idea, the proposed mixture-model GCN is given by:

$$\mathbf{h}'_i = \sigma\left(\frac{1}{|Ne(i)|} \sum_{j \in Ne(i)} \frac{1}{R} \sum_{r=1}^R \Theta_r(\mathbf{u}_{i,j}) \odot \mathbf{W}_r \mathbf{h}_j\right) \quad (10)$$

$$\Theta_r(\mathbf{u}) = \exp\left(-\frac{1}{2}(\mathbf{u} - \boldsymbol{\mu}_r)^T \boldsymbol{\Sigma}_r^{-1}(\mathbf{u} - \boldsymbol{\mu}_r)\right) \quad (11)$$

where $|Ne(i)|$ is the number of neighbors of zone i , R is the number of kernels, $\Theta_r(\cdot)$ is a function calculating edge weights

under each kernel, \odot is the element-wise multiplication operator, $\mathbf{u}_{i,j}$ is the vector of edge weights for the edge linking zone i and zone j , \mathbf{W}_r represents a linear transformation matrix under r^{th} kernel. In equation (11), $\boldsymbol{\mu}_r$ and $\boldsymbol{\Sigma}_r$ are parameters to learn during training. Aggregation method in equation (10) is just similar to the one in equation (9), except the setting of multiple kernels and substitution of pre-set edge weights with the parametric Gaussian function in (11).

An important factor for the construction of edge weights here is the selection of edge weight vector \mathbf{u} . The choice should sufficiently capture different correlations between each pair of two zones, while the vector dimension is limited for saving space and speed of computation. On the one hand, geographical correlation should be considered since intuitively, zones with closer geographic location usually share higher similarities in future mobility demand pattern. On the other hand, distant neighbors can also provide meaningful information, especially for those possessing strong functionality similarity with the studied zone. To consider the geographical closeness and functionality properties simultaneously, we utilize $([\mathbf{C}_D]_{i,j}, [\mathbf{C}_F]_{i,j})$ as the weight vector for edge between zone i and zone j . Although each entry of this vector represents different correlations, complex interactions may exist between them. For example, a neighbor with a certain distance from the studied zone and higher functionality correlation is more likely to share strong similarity in demand representations, compared to one with the same geographical closeness but lower correlation in functionality. This is why we need to integrate different edge weights together into $\Theta_r(\mathbf{u})$ in equation (11), where $\boldsymbol{\mu}_r$ and $\boldsymbol{\Sigma}_r$ are employed and learned by the model itself, in order to depict the intrinsic relationship of the weights based on the shape of the two-dimension Gaussian function.

With the construction of MGCN, we build four learning paths in the GCN basic module based on the adjacency matrices given in the last section. Along each path, multiple MGCN layers are first built with one of the four adjacency matrices, and then stacked together to collect information from different neighborhoods. Usually, two to three layers of MGCN are sufficient for the message passing process. In the final step of this module, the last hidden representations learned from the four paths are fused together, given by:

$$\mathbf{H}_M = [\mathbf{H}_G, \mathbf{H}_F, \mathbf{H}_D, \mathbf{H}_{OD}] \quad (12)$$

where \mathbf{H}_M is the output representation of the GCN basic module, \mathbf{H}_G , \mathbf{H}_F , \mathbf{H}_D , and \mathbf{H}_{OD} are final representations learned from the four paths respectively constructed on \mathbf{A}_G , \mathbf{A}_F , \mathbf{A}_D , \mathbf{A}_{OD} .

D. Matrix factorization module

Patterns of ride-hailing demand for different zones can be quite diversified from each other. For example, residential zones may become major destinations during late afternoon when most people prepare to return home, while the situation is on the contrary for the zones where people work. Since the zone-based patterns of mobility may become quite different from each other, it is difficult to capture the uniqueness of each

zone via a single same decoder. To address the problem, an intuitive solution is to train zone-wise decoders to decode their representations. However, this strategy, on the one hand, will introduce too many extra parameters, which generate heavy burdens in the speed and space of computation, especially with a large number of zones. On the other hand, the intrinsic correlations within different regions should also be considered in the decoders. Multiple methods have been explored to integrate both diversity and similarity of different prediction task inside one forecasting module, including building multi-linear relationship within decoders [38], or the utilization of matrix factorization technology [36]. Following the insights of the last one and on the basis of task definitions of inflow, outflow and OD-based demand prediction, we formulate matrix factorization (MF) layers as shown in Fig. 4, given by:

$$\mathbf{V} = \text{reshape}(\mathbf{V}_R \mathbf{V}_B) \quad (13)$$

$$\mathbf{h}_i^{new} = \mathbf{V}_i \mathbf{h}_i, \forall i \in \{1, \dots, N\} \quad (14)$$

where \mathbf{V} is the decoder tensor whose each layer \mathbf{V}_i represents a linear transformation matrix, $\text{reshape}(\cdot)$ is a function to transform matrix into tensor, \mathbf{h}_i and \mathbf{h}_i^{new} are input and output hidden representations for each zone, and N is the total number of zones. The utilization of tensor \mathbf{V} enables every zone to possess a unique decoder. With matrix factorization in equation (13), we decompose the large tensor \mathbf{V} into two sub-matrices \mathbf{V}_R and \mathbf{V}_B . In this way, the target to learn becomes the sub-matrices instead of the tensor \mathbf{V} , which reduces the total number of decoder parameters. Moreover, the similarity within zones can also be captured by the share of base \mathbf{V}_B in zone-wise decoding matrix \mathbf{V}_i .

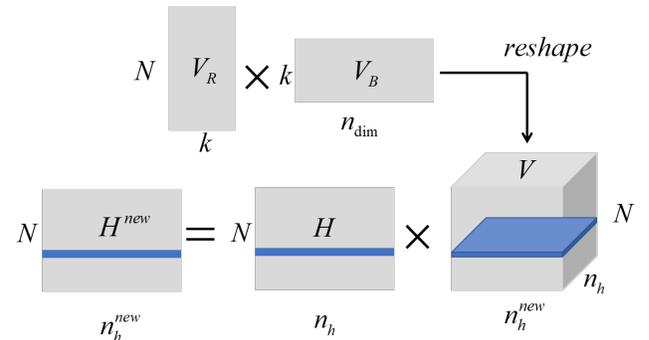


Fig. 4: Matrix factorization process. \mathbf{V}_R and \mathbf{V}_B are two sub-matrices for the proposed model to learn. \mathbf{V} is the tensor actually used as a region-specific decoder. N and k are respectively the number of zones and bases. n_{dim} equals to the product of n_h and n_h^{new} , which are the dimensions of original and new hidden representations for each zone.

With the construction of basic MF layers, we design the whole matrix factorization module for multi-task learning. The module consists of three groups of decoders, each of which contains a sequence of MF layer. The hidden representation generated from the GCN basic module is respectively fed into the three decoder groups for the prediction of inflow, outflow

and OD-based demand. For OD-based demand prediction, we add an extra final decoder given by:

$$m_{i,j} = \mathbf{h}_{f,i}^T \mathbf{V}_{f,i} \mathbf{h}_{f,j} \quad (15)$$

where $\mathbf{h}_{f,i}$ and $\mathbf{h}_{f,j}$ are representations for zone i and j generated by the final MF layer for OD-based prediction, $\mathbf{V}_{f,i}$ is a transmission matrix constructed with matrix factorization. In Fig. 3, we use FD to represent the process in equation (15). In contrast with the classic OD-based demand function in equation (1), $\mathbf{h}_{f,i}$ and $\mathbf{h}_{f,j}$ play the same role as A_i and B_j , representing unique properties of each zone. $\mathbf{V}_{f,i}$ captures correlations between zone i and other zones as $f(c_{i,j})$ in eq. (1). The difference here is that all the parameters in our model are directly learned by the deep learning model, instead of the pre-definition and investigation by researchers as in classic OD demand function. This design of decoder, on the one hand, can uncover and employ hidden information stored in historical data, which may be ignored by human investigators. On the other hand, the time-consuming survey of region properties are also removed, showing the advantage in convenience for data-driven approach.

With model output for all the three tasks, we formulate the loss function as below:

$$\mathcal{L} = \beta_{od} \mathcal{L}_{od} + \frac{1}{\alpha} \beta_{in} \mathcal{L}_{in} + \frac{1}{\alpha} \beta_{out} \mathcal{L}_{out} \quad (16)$$

where \mathcal{L}_{od} , \mathcal{L}_{in} , \mathcal{L}_{out} are MSE losses of OD-based, inflow and outflow demand predictions, β_{od} , β_{in} , and β_{out} are weights which sum to one, and α is a coefficient utilized to scale the loss magnitude. The loss functions of each sub-task are formulated as follows:

$$\mathcal{L}_{od} = \frac{1}{S \times |M^{t+1}|} \sum_{s=1}^S \|M_s^{t+1} - \hat{M}_s^{t+1}\|_F^2 \quad (17)$$

$$\mathcal{L}_{in} = \frac{1}{S \times |X_{in}^{t+1}|} \sum_{s=1}^S \|X_{in,s}^{t+1} - \hat{X}_{in,s}^{t+1}\|_2^2 \quad (18)$$

$$\mathcal{L}_{out} = \frac{1}{S \times |X_{out}^{t+1}|} \sum_{s=1}^S \|X_{out,s}^{t+1} - \hat{X}_{out,s}^{t+1}\|_2^2 \quad (19)$$

where S is the number of samples, $|M^{t+1}|$, $|X_{in}^{t+1}|$, $|X_{out}^{t+1}|$ represent the number of entries for OD matrix and zone-based demand vectors respectively. $\|\cdot\|_F$ represents the Frobenius norm for matrix, and $\|\cdot\|_2$ is the L2 norm for vector.

V. EXPERIMENTAL RESULTS

A. Data and models

We implement experiments on two ride-hailing dataset. The first one is a for-hire-vehicle dataset released by New York TLC in September 2018 (<https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page>). The dataset is built on data reported by ride-hailing company like Uber and Lyft, including trip-specific information such as pick-up and drop-off time, origin and destination. The origin and destination of each raw trip record are provided in the form of TLC zone id, which is determined based on zip code. The partition method for

Manhattan by TLC is exactly as shown in Fig. 1b. In this experiment, we mainly focus on trips with more than one customers sharing a single vehicle, called shared service mode. Only orders with non-empty records of timestamp, origin and destination will be counted. After filtering, the dataset contains a total of 18970027 OD transactions.

In addition to ride-hailing dataset with OD record, we also employ the Smart Location Database to collect properties including distance between regions, and zone-specific information like residences number, employment rate and other point of interest. With the combination of the two datasets, we then construct the two correlation matrices specified in Part B of section IV, which capture similarity in distance and functionality respectively. In Fig. 5, we randomly select a TLC zone located at downtown Manhattan and show the correlations between this one and the others with graph representations. The studied zone (id 232) is marked with red color, and the color depth of the other zones relates to the level of correlation. Fig. 5b demonstrates that geographically remote zones can also have high correlation with the studied one. It is necessary to construct multiple semantic adjacency matrices to mine different dependencies sufficiently.

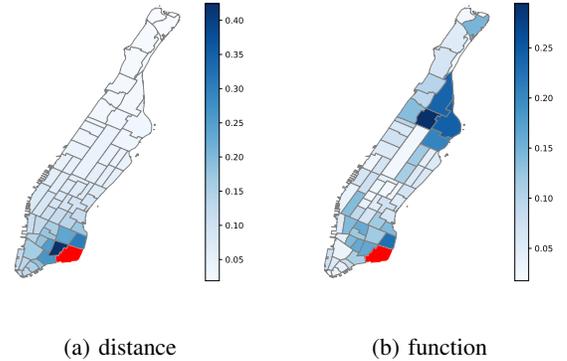


Fig. 5: Representations of correlation matrices for zone 232

Demand patterns may be stable in mega-city like New York. However, in other tourism areas, the pattern may change a lot across seasons. To validate the robustness, we also test our method on a dataset released by DiDi Chuxing for Haikou, a tourism city in southern China. The dataset we used (<https://gaia.didichuxing.com>) contains ride-hailing trips for five month since June 2017. The Haikou city is first partitioned into 10×10 rectangular grids which contains basically all the recorded OD data. Some grids with rare data records are then deleted, leaving 59 grids which include most of the historical OD trips. The remaining grid network is naturally classified into non-Euclidean domain. All the OD transactions considered are then processed to contain zone-based origin and destination information, as well as the corresponding time stamps. Similar process of filtering as done on the Manhattan dataset is then implemented for Haikou data. Correlation matrices and semantic adjacency matrices are also constructed. In addition, our method follows a routine of using both tendency and periodicity features for predictions (introduced later in the following paragraphs), while no specific module is added to

characterize the seasonal patterns. Instead, these patterns are learned by training on a sufficiently long horizon such that the model can experience many scenarios, including the scenarios with suddenly changing demand over weeks.

In the experiment for Manhattan data, we use demand data from January 8, 2018 to November 4, 2018 for training, November 5, 2018 to December 2, 2018 for validation, and December 3, 2018 to December 31, 2018 for testing. For Haikou, demand data from June 5, 2017 to October 2, 2017 is used for training, October 3, 2017 to October 16, 2017 for validation, and October 17, 2017 to October 30, 2017 for testing. To demonstrate the effectiveness of the proposed model, we compare it with the following state-of-the-art machine learning competitors:

- **GraphSAGE [26]**: a graph convolutional model that aggregates adjacent information by taking average of neighbor features.
- **GEML [25]**: a graph convolutional model that refines GraphSAGE with pre-weighted aggregator.
- **ST-GCN [4]**: a graph convolutional model that employs ChebNet [40] with multiple graphs.
- **MLP**: multi-layer perception network, which is a standard deep learning model.
- **XGB**: XGBoost model, a scalable end-to-end gradient boosted decision trees.
- **GBDT**: gradient boosting decision tree model which build multiple regression decision trees.
- **RF**: random forest model which is a classical ensemble approach trained by bootstrapped samples for each constructed decision tree.
- **LASSO**: LASSO model which adds a L1-norm regularization term to the linear regression as constraints on parameter size.
- **HA**: the history average of past inflow, outflow and OD-based demand respectively.

In the experiments, the standard CNN-based method cannot be applied since the demand data is handled in graphs. Instead, we employ GCN-based method including GraphSAGE, GEML and ST-GCN as some of the major baseline approaches to compare the ability of capturing spatial dependencies. For the temporal dependencies, we select historical observations of demands at time intervals $t, t-1$ (the time intervals before one and two hours), $t+1-24$ (the time interval for the same hour on yesterday), $t+1-24 \times 7$ (the time interval for the same hour on the same day of last week) as features. As given by Zhang et al. [41], the former two represent temporal tendency, which shows short-term dependency within time intervals. The long-term temporal dependency, or say periodicity, is considered by the last two, which capture demand patterns repeated over days and over weeks. In this way, the temporal dependency is well-captured directly in input features, instead of employing extra RNN-related modules to do the same task.

In addition to the baseline methods, we also implement eight variants of the proposed one as sensitivity analysis for both the major modules:

- **MT-MF-GCN-V1 to -V4**: the combination of geographical and multiple semantic neighborhoods is replaced

by single one, respectively constructed on geographical adjacency, function similarity, distance and commuting history.

- **MT-MF-GCN-V5**: the matrix factorization layers are replaced by linear transformation (LT) layers. The form of LT layers is the same as in eq. (14), except that we substitute different V_i for each zone with a single same matrix directly learned by the model, rather than obtained from a certain layer of the tensor V .
- **MT-MF-GCN-V6**: a single-task model with the same encoder structure as the proposed one, except that there is only one group of MF layers for the OD demand prediction.
- **MT-MF-GCN-V7 to -V8**: single-task models with the same structure as the proposed one, respectively for inflow and outflow demand predictions.

For model specifications, the three coefficients for inflow, outflow and OD-based prediction error in the loss function are 0.1, 0.1 and 0.8, selected from the set of [0.1, 0.1, 0.8], [0.2, 0.2, 0.6], [0.33, 0.33, 0.33], [0.4, 0.4, 0.2]. The parameter for scaling of inflow and outflow loss items is 1000, selected from the set of 1, 10, 100, 1000. Learning rate is selected from the set of 0.0001, 0.001, 0.01 and 0.1. The layer and size of GCN is determined from the set of [64, 128], [128, 256], [64, 128, 64] and [128, 256, 128]. All the neural network based baselines also share these sets of parameters if applicable. The batch size for training is 16 and the maximum number of epoch is 300. PyTorch is utilized to implement the experiments with ADAM as the optimizer. For LASSO, the related parameter alpha is chosen from the set of 0.1, 1, 10. For XGB and GBDT, the max depth of tree is selected from 3, 5 and 7. For RF, the number of trees is chosen from 50, 100, 200. All the models are fine-tuned on the applicable parameter sets. For details of hardware, a server with 64G RAM and one NVIDIA 1080Ti GPU are employed for the implementation of the experiments.

B. Comparison with baselines

To evaluate the prediction performance, we utilize three measurements including Root Mean Square Error (RMSE), Mean Absolute Error (MAE) and Mean Absolute Percentage Error (MAPE). The performance of baselines and the proposed model, MT-MF-GCN, are summarized in TABLE I for the Manhattan dataset. For all the three tasks of inflow, outflow and OD-based prediction, the MT-MF-GCN model outperforms both the traditional group of methods and classic GCN models. For Manhattan dataset, compared to the best performance of traditional ones, the proposed model reduces RMSE/MAE/MAPE by 10.256%/12.079%/5.965% for OD-based demand, 17.992%/16.179%/20.413% for inflow demand, and 21.566%/19.256%/22.843% for outflow demand. In comparison with the best performance of GCN-based baselines (GraphSAGE, GEML and ST-GCN), the improvement mainly comes from zone-demand prediction, where the reductions of errors are 8.934%/8.355%/12.994% for inflow demand, and 5.964%/6.688%/7.317% for outflow demand.

For tests on Haikou data, the results are shown in TABLE II. In comparison with the best performance of

traditional prediction models, the proposed model reduces RMSE/MAE/MAPE by 6.634%/15.144%/10.811% for OD-based prediction task. For inflow-based demand prediction, the reductions are 4.129% for RMSE and 6.902% for MAE, while MAPE basically keeps the same. For outflow-based demand prediction, RMSE/MAE/MAPE are reduced by 3.100%/9.762%/6.593%. Compared to the best results from GCN-based baselines, the proposed model reduces RMSE/MAE/MAPE by 7.463%/3.846%/5.714% for OD-based prediction, 1.751%/10.751%/12.671% for outflow prediction and more than 5% for the three measurements for inflow prediction. The results demonstrate the superiority of the proposed multi-task matrix factorized graph neural network model (MT-MF-GCN) over other traditional or GCN-based prediction models on the tested datasets. The improvement mainly comes from the following aspects: 1) The proposed model can capture the spatial dependence in terms of distance, neighborhood, functionality and mobility interaction simultaneously, while the other GCN-based models only consider one or part of them. Diversified neighborhood structures provide more complete information for the proposed model to formulate its hidden representations. 2) The baseline models either utilize pre-defined edge weights (such as GEML), or only single edge weight (such as GraphSAGE and ST-GCN) to construct GCN layer for each neighborhood. In comparison, the proposed model adopts mixture-model GCN layers for graph embedding, where the combination of different edge weights are considered and self-adjusted by model training. The mixture-model GCN structure helps to more accurately capture general hidden representations in the shared embedding module for multi-tasks. 3) The proposed model utilizes matrix factorization technology to build its region-specific decoders. Compared to linear transformation decoders used in other baselines, the proposed one can provide higher complexity to sufficiently restore the unique information for each sub-tasks, and still with a relatively low price. 4) The multi-task model structure leads to good solutions for sub-tasks. The shared encoder can efficiently capture general information, while the decoders built on matrix factorization layers keep the parameter uniqueness for each sub-task. The effectiveness of the multi-task structure is verified by the testing results, where the proposed model shows better performance in each sub-task over single-task models such as MLP, GraphSAGE and ST-GCN.

To further verify the real-time applicability of the method, we first compare the training times within MLP, GEML and the proposed MT-MF-GCN model. On Manhattan dataset, the training time for MT-MF-GCN over 100 epoch with 16 samples in a batch is 135 minutes, compared to 42 minutes for GEML and 7 minutes for MLP. Regarding the speed of model prediction on the test set, 1.4 seconds (testing time) is required by the proposed model for generating the predictions on the three tasks for the whole test set of Manhattan, containing all the samples within one month. The testing time is 0.87 seconds for GEML, and 0.5 seconds for MLP. Although the proposed model has longer training time than the baselines due to a more complex architecture, the test time is acceptable. In actual operations, it is the test time that matters since the

company can train the model offline and conduct predictions with the trained model online.

TABLE I: Comparison with baselines for Manhattan dataset

OD-based demand			
Model	RMSE	MAE	MAPE
HA	1.312	0.537	0.987
LASSO	1.213	0.772	0.591
RF	0.874	0.518	0.573
GBDT	0.858	0.515	0.570
XGB	0.859	0.515	0.570
MLP	0.858	0.505	0.584
GraphSAGE	0.826	0.472	0.547
GEML	0.789	0.451	0.546
ST-GCN	0.776	0.446	0.541
MT-MF-GCN	0.770	0.444	0.536
Inflow demand			
Model	RMSE	MAE	MAPE
HA	14.59	8.852	0.526
LASSO	12.954	8.021	0.484
RF	11.278	7.340	0.410
GBDT	10.778	6.944	0.393
XGB	10.674	6.925	0.393
MLP	10.416	6.805	0.387
GraphSAGE	9.700	6.482	0.375
GEML	11.813	8.020	0.496
ST-GCN	9.380	6.224	0.354
MT-MF-GCN	8.542	5.704	0.308
Outflow demand			
Model	RMSE	MAE	MAPE
HA	14.464	9.026	0.520
LASSO	12.494	7.980	0.476
RF	11.196	7.406	0.420
GBDT	10.670	7.017	0.401
XGB	10.614	6.996	0.401
MLP	10.463	6.912	0.394
GraphSAGE	9.939	6.677	0.389
GEML	10.478	7.101	0.432
ST-GCN	8.853	5.981	0.328
MT-MF-GCN	8.325	5.581	0.304

C. Comparison with variants

As a sensitivity analysis for both the major modules, we test the performances of variants V1 to V8, which are shown in TABLE III and IV. By comparing V1 to V4 with the proposed model, we verify the necessity to combine geographical adjacency graph (V1) with multiple semantic graphs together. By comparing V5 with the proposed model, we aim to emphasize on the importance of MF layers and demonstrate to what degree they outperform regular linear decoders. By comparing V6 to V8, the goal is to show the effectiveness in integration of the graph embedding modules.

Usually, the most natural neighborhood is constructed by geographical adjacency. With this straightforward definition (V1), the prediction errors of the three tasks, are instead the worst, especially for the inflow and outflow demand prediction on both dataset. On the contrary, all the variants with semantic neighborhoods (V2 to V4) show improvement in all the prediction tasks compared to V1 for the Manhattan dataset, and also improvement in zone-based demand prediction tasks for the Haikou dataset. The combination of all the graphs, as suggested in our model, further strengthens the performance in the multi-task prediction. This demonstrates the significance of employing the extra correlations and information from multiple semantic neighborhoods.

TABLE II: Comparison with baselines for Haikou dataset

OD-based demand			
Model	RMSE	MAE	MAPE
HA	8.152	1.038	0.999
LASSO	8.094	1.751	0.915
RF	2.385	0.391	0.361
GBDT	2.349	0.383	0.337
XGB	2.333	0.383	0.338
MLP	2.231	0.388	0.333
GraphSAGE	2.355	0.395	0.392
GEML	2.202	0.349	0.328
ST-GCN	2.251	0.338	0.315
MT-MF-GCN	2.083	0.325	0.297
Inflow demand			
Model	RMSE	MAE	MAPE
HA	39.169	11.289	0.305
LASSO	32.526	9.959	0.284
RF	30.918	9.683	0.301
GBDT	31.236	9.514	0.282
XGB	30.644	9.600	0.282
MLP	30.078	9.475	0.284
GraphSAGE	34.634	12.988	0.524
GEML	31.622	11.679	0.503
ST-GCN	31.251	10.853	0.391
MT-MF-GCN	28.836	8.821	0.287
Outflow demand			
Model	RMSE	MAE	MAPE
HA	43.194	11.090	0.293
LASSO	35.780	9.973	0.278
RF	33.366	9.432	0.295
GBDT	32.439	9.385	0.276
XGB	31.901	9.301	0.273
MLP	32.989	9.631	0.273
GraphSAGE	31.463	9.664	0.334
GEML	33.290	10.709	0.447
ST-GCN	32.518	9.404	0.292
MT-MF-GCN	30.912	8.393	0.255

For V5, the RMSE/MAE/MAPE increase by 10.9%/11.2%/21.1% and 6.1%/10.9%/12.2% for inflow prediction respectively on Manhattan and Haikou dataset, compared to the proposed one. The difference in performance arises from the complexity of layer structure. In the linear transformation layer of V5, a single matrix is utilized to decode representations of all the zones. As a result, the parameters in the matrix may be directed to represent common characteristics of all the zones, with the sacrifice of part of uniqueness of each zone. This problem is addressed in the proposed model with matrix factorization decoders. In eq. (13) and eq. (14), each zone is provided with its own decoder matrix to sufficiently represent their uniqueness. The correlation within decoders is also considered via co-training of base matrix V_B .

As a multi-task model, the proposed one performs at least similarly well as the single-task models (MT-MF-GCN-V6, -V7 and -V8) in each corresponding sub-task. However, it takes fewer encoder parameters to achieve such performance compared to the summation of single-task models, since the multi-task model share and only need to train one graph embedding module (GCN Basic Module), while there are three such modules required to be separately stored and trained for the single-task ones. This indicates the effectiveness of the proposed model in discovering good solutions for zone demand and OD demand predictions simultaneously.

TABLE III: Comparison with variants for Manhattan dataset

OD-based demand			
Model	RMSE	MAE	MAPE
MT-MF-GCN	0.770	0.444	0.536
MT-MF-GCN-V1	0.774	0.445	0.537
MT-MF-GCN-V2	0.773	0.444	0.536
MT-MF-GCN-V3	0.771	0.446	0.533
MT-MF-GCN-V4	0.771	0.446	0.534
MT-MF-GCN-V5	0.772	0.447	0.538
MT-MF-GCN-V6	0.771	0.446	0.534
Inflow demand			
Model	RMSE	MAE	MAPE
MT-MF-GCN	8.542	5.704	0.308
MT-MF-GCN-V1	9.714	6.384	0.344
MT-MF-GCN-V2	9.163	6.060	0.322
MT-MF-GCN-V3	8.933	5.953	0.324
MT-MF-GCN-V4	8.872	5.893	0.325
MT-MF-GCN-V5	9.473	6.343	0.373
MT-MF-GCN-V7	8.646	5.734	0.308
Outflow demand			
Model	RMSE	MAE	MAPE
MT-MF-GCN	8.325	5.581	0.304
MT-MF-GCN-V1	8.785	5.874	0.322
MT-MF-GCN-V2	8.775	5.895	0.320
MT-MF-GCN-V3	8.561	5.760	0.316
MT-MF-GCN-V4	8.595	5.779	0.324
MT-MF-GCN-V5	8.574	5.837	0.329
MT-MF-GCN-V8	8.410	5.617	0.304

TABLE IV: Comparison with variants for Haikou dataset

OD-based demand			
Model	RMSE	MAE	MAPE
MT-MF-GCN	2.083	0.325	0.297
MT-MF-GCN-V1	2.116	0.330	0.298
MT-MF-GCN-V3	2.134	0.327	0.299
MT-MF-GCN-V4	2.129	0.326	0.301
MT-MF-GCN-V5	2.113	0.330	0.304
MT-MF-GCN-V6	2.090	0.326	0.300
Inflow demand			
Model	RMSE	MAE	MAPE
MT-MF-GCN	28.836	8.821	0.287
MT-MF-GCN-V1	30.181	9.383	0.315
MT-MF-GCN-V3	28.833	9.121	0.308
MT-MF-GCN-V4	29.299	9.061	0.294
MT-MF-GCN-V5	30.602	9.780	0.322
MT-MF-GCN-V7	28.883	9.091	0.302
Outflow demand			
Model	RMSE	MAE	MAPE
MT-MF-GCN	30.912	8.393	0.255
MT-MF-GCN-V1	32.299	8.697	0.266
MT-MF-GCN-V3	31.472	8.588	0.272
MT-MF-GCN-V4	32.242	8.682	0.261
MT-MF-GCN-V5	32.147	8.670	0.264
MT-MF-GCN-V8	30.908	8.986	0.281

D. Visualizations of predictions

In order to provide a direct presentation, we select a TLC zone of Manhattan with id 79 and illustrate prediction results for it from Fig. 6 to Fig. 8. The chosen zone is an entertaining area located at downtown Manhattan, which possesses the largest historical daily demand of ride-sharing service. Fig. 6 presents the mean observed and predicted inflow and outflow patterns within a week for December. A phenomenon is that there is only one peak period for the inflow demand on weekday, which is larger and prior to the evening peak of outflow. A potential explanation is that most people come to zone 79 for dinner, which usually happens after 5 pm. This notable difference and other mobility patterns can be

effectively observed and captured by the proposed model.

In Fig. 7 and Fig. 8, we show the origin and destination distribution respectively for requests departing to and from a selected zone during evening peak. The selected zone, zone 79, is marked with a green box and the color depth relates to the level of demand. For example, a zone marked with 0-4 in Fig. 7 means that the number of OD trips departing from this zone to zone 79 is within the range of 0 to 4, while a zone marked with 4-7 in Fig. 8 means that the number of OD trips departing from zone 79 to this zone is within the range of 4 to 7. Thus, similar spatial distributions of color-depth indicate high similarity in OD flow patterns with a fixed destination (Fig. 7) or origin (Fig. 8). As shown in Fig. 7 and Fig. 8, the predicted OD flow distribution is similar to the real distribution. The ride-sharing demand is predicted to concentrate on downtown Manhattan at evening peak, while distant orders rarely emerge, which accords with the observed pattern.

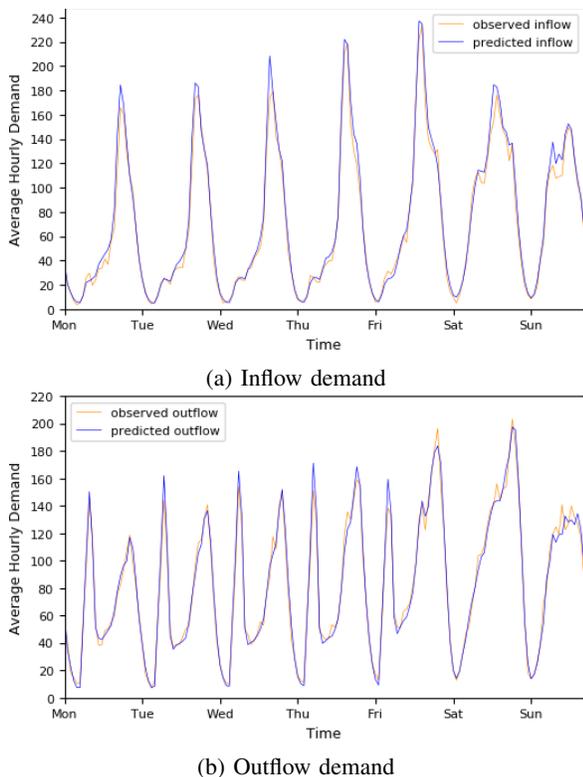


Fig. 6: Hourly zone-based predictions of MT-MF-GCN model for zone 79. The results are generated by taking average of all the predictions for each week during December 2018.

VI. CONCLUSION

This paper studies the co-prediction problem of inflow, outflow and OD-based ride-hailing demand, which provides valuable information for the operation of ride-hailing platform. To address the challenges of multi-task learning and predictions with irregular zone partition, we first develop a GCN basic module based on multi-graph mixture-model GCN. The correlation matrices and multiple neighborhoods are constructed to capture both geographical and semantic

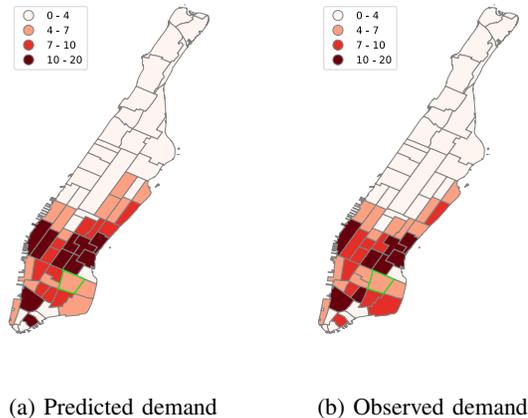


Fig. 7: Heat maps of demand departing to zone 79 with different origins during evening peak for MT-MF-GCN predictions and observed demand.

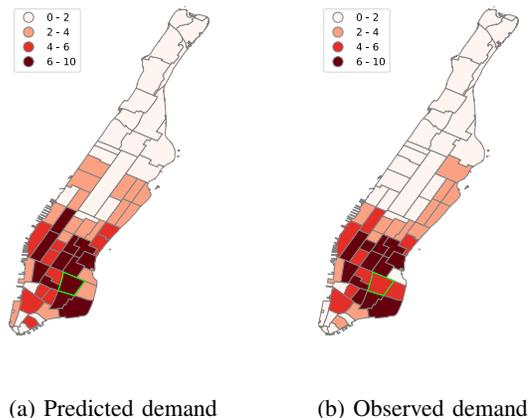


Fig. 8: Heat maps of demand departing from zone 79 with different destinations during evening peak for MT-MF-GCN predictions and observed demand.

correlations. Based on different neighborhoods, mixture-model GCNs are utilized to aggregate information with Gaussian weighting functions learned by the model itself. Then we employ a multi-task learning module with matrix factorization (MF) layers as region-specific predictors, to separately decode representations for each zone and each task. The proposed end-to-end model is called multi-task matrix factorized graph neural network. With evaluation on two real-world ride-hailing datasets, the proposed model is shown able to outperform baseline methods significantly. The effectiveness of each module and the significance of co-prediction structure are then validated by a sensitivity analysis. In the future, the proposed framework can be further explored from the following perspectives: 1) Constant time intervals for prediction can be replaced by time-variant intervals to improve the flexibility and accuracy of the prediction model. 2) Current objective function is formalized as a linear combination of loss function of each sub-task. The coefficients of sub-tasks

in the objective are determined via empirical tests. In the future, other forms of objective function can be explored, and some multi-objective optimization technology can be applied to adjust the coefficients in loss function with more adaptivity. 3) Different transportation modes and services can be tested with the proposed model framework, like bike-sharing, take-out services and even shared autonomous on-demand mobility, etc. 4) The current model framework is designed for prediction in a single city. Transfer Learning technology can be applied in future to co-train models with shared modules for different cities, in order to increase the data efficiency. 5) The proposed model mainly considers tendency and periodicity features for its predictions. A mechanism/module to incorporate seasonal features, however, is also important and merits future explorations.

ACKNOWLEDGMENT

The work described in this paper was supported by a grant from Hong Kong Research Grants Council under project HKUST16208619 and a NSFC/RGC Joint Research grant N_HKUST627/18 (NSFC-RGC 71861167001). This work was also supported by the Hong Kong University of Science and Technology - DiDi Chuxing (HKUST-DiDi) Joint Laboratory. Also, the second data source for Haikou is provided by Didi Chuxing GAIA Initiative.

REFERENCES

- [1] Y. Tong, Y. Chen, Z. Zhou, L. Chen, J. Wang, Q. Yang, J. Ye, and W. Lv, "The simpler the better: a unified approach to predicting original taxi demands based on large-scale online platforms," in *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, 2017, pp. 1653–1662.
- [2] J. Ke, H. Zheng, H. Yang, and X. M. Chen, "Short-term forecasting of passenger demand under on-demand ride services: A spatio-temporal deep learning approach," *Transportation Research Part C: Emerging Technologies*, vol. 85, pp. 591–608, 2017.
- [3] 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.
- [4] X. Geng, Y. Li, L. Wang, L. Zhang, Q. Yang, J. Ye, and Y. Liu, "Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 3656–3663.
- [5] J. Ke, X. Qin, H. Yang, Z. Zheng, Z. Zhu, and J. Ye, "Predicting origin-destination ride-sourcing demand with a spatio-temporal encoder-decoder residual multi-graph convolutional network," *arXiv preprint arXiv:1910.09103*, 2019.
- [6] 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.
- [7] J. Ye, L. Sun, B. Du, Y. Fu, X. Tong, and H. Xiong, "Co-prediction of multiple transportation demands based on deep spatio-temporal neural network," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 305–313.
- [8] Y. Sheffi, *Urban Transportation Networks: Equilibrium Analysis With Mathematical Programming Methods*, 01 1984.
- [9] M. S. Ahmed and A. R. Cook, *Analysis of freeway traffic time-series data by using Box-Jenkins techniques*, 1979, no. 722.
- [10] M. Levin and Y.-D. Tsao, "On forecasting freeway occupancies and volumes (abridgment)," *Transportation Research Record*, no. 773, 1980.
- [11] N. Zhang, Y. Zhang, and H. Lu, "Seasonal autoregressive integrated moving average and support vector machine models: Prediction of short-term traffic flow on freeways," *Transportation Research Record*, vol. 2215, no. 1, pp. 85–92, 2011.
- [12] M. Khashei, M. Bijari, and G. A. R. Ardali, "Hybridization of autoregressive integrated moving average (arima) with probabilistic neural networks (pnns)," *Computers & Industrial Engineering*, vol. 63, no. 1, pp. 37–45, 2012.
- [13] X. Fei, C.-C. Lu, and K. Liu, "A bayesian dynamic linear model approach for real-time short-term freeway travel time prediction," *Transportation Research Part C: Emerging Technologies*, vol. 19, no. 6, pp. 1306–1318, 2011.
- [14] Z. Zhu, B. Peng, C. Xiong, and L. Zhang, "Short-term traffic flow prediction with linear conditional gaussian bayesian network," *Journal of Advanced Transportation*, vol. 50, no. 6, pp. 1111–1123, 2016.
- [15] C.-C. Lu and X. Zhou, "Short-term highway traffic state prediction using structural state space models," *Journal of Intelligent Transportation Systems*, vol. 18, no. 3, pp. 309–322, 2014.
- [16] C. Antoniou, H. N. Koutsopoulos, and G. Yannis, "Dynamic data-driven local traffic state estimation and prediction," *Transportation Research Part C: Emerging Technologies*, vol. 34, pp. 89–107, 2013.
- [17] D. Park and L. R. Rilett, "Forecasting multiple-period freeway link travel times using modular neural networks," *Transportation research record*, vol. 1617, no. 1, pp. 163–170, 1998.
- [18] K. Y. Chan, T. S. Dillon, J. Singh, and E. Chang, "Neural-network-based models for short-term traffic flow forecasting using a hybrid exponential smoothing and levenberg-marquardt algorithm," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 2, pp. 644–654, 2011.
- [19] S. Sun, C. Zhang, and G. Yu, "A bayesian network approach to traffic flow forecasting," *IEEE Transactions on intelligent transportation systems*, vol. 7, no. 1, pp. 124–132, 2006.
- [20] Z. Zhu, L. Tang, C. Xiong, X. Chen, and L. Zhang, "The conditional probability of travel speed and its application to short-term prediction," *Transportmetrica B: Transport Dynamics*, vol. 7, no. 1, pp. 684–706, 2019.
- [21] Y. Ma, E. W. M. Lee, and R. K. K. Yuen, "An artificial intelligence-based approach for simulating pedestrian movement," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 11, pp. 3159–3170, 2016.
- [22] Y. Ma, E. W. Lee, Z. Hu, M. Shi, and R. K. Yuen, "An intelligence-based approach for prediction of microscopic pedestrian walking behavior," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 10, pp. 3964–3980, 2019.
- [23] J. Yuen, E. W. M. Lee, S. Lo, and R. K. K. Yuen, "An intelligence-based optimization model of passenger flow in a transportation station," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 3, pp. 1290–1300, 2013.
- [24] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, "A comprehensive survey on graph neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, 2020.
- [25] 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.
- [26] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Advances in neural information processing systems*, 2017, pp. 1024–1034.
- [27] Z. Yuan and E. Oja, "Projective nonnegative matrix factorization for image compression and feature extraction," in *Scandinavian Conference on Image Analysis*. Springer, 2005, pp. 333–342.
- [28] D. Lian, C. Zhao, X. Xie, G. Sun, E. Chen, and Y. Rui, "Geomf: joint geographical modeling and matrix factorization for point-of-interest recommendation," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2014, pp. 831–840.
- [29] H. Ma, H. Yang, M. R. Lyu, and I. King, "Sorec: social recommendation using probabilistic matrix factorization," in *Proceedings of the 17th ACM conference on Information and knowledge management*, 2008, pp. 931–940.
- [30] D. Guillamet, B. Schiele, and J. Vitria, "Analyzing non-negative matrix factorization for image classification," in *Object recognition supported by user interaction for service robots*, vol. 2. IEEE, 2002, pp. 116–119.
- [31] Y. Gao and G. Church, "Improving molecular cancer class discovery through sparse non-negative matrix factorization," *Bioinformatics*, vol. 21, no. 21, pp. 3970–3975, 2005.
- [32] J. D. Rennie and N. Srebro, "Fast maximum margin matrix factorization for collaborative prediction," in *Proceedings of the 22nd international conference on Machine learning*, 2005, pp. 713–719.

- [33] Y. Liao, W. Du, P. Geurts, and G. Leduc, "Dmfsgd: A decentralized matrix factorization algorithm for network distance prediction," *IEEE/ACM Transactions on Networking*, vol. 21, no. 5, pp. 1511–1524, 2012.
- [34] A. K. Menon and C. Elkan, "Link prediction via matrix factorization," in *Joint european conference on machine learning and knowledge discovery in databases*. Springer, 2011, pp. 437–452.
- [35] Q. Chen, X. Song, Z. Fan, T. Xia, H. Yamada, and R. Shibasaki, "A context-aware nonnegative matrix factorization framework for traffic accident risk estimation via heterogeneous data," in *2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*. IEEE, 2018, pp. 346–351.
- [36] Z. Pan, Z. Wang, W. Wang, Y. Yu, J. Zhang, and Y. Zheng, "Matrix factorization for spatio-temporal neural networks with applications to urban flow prediction," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2019, pp. 2683–2691.
- [37] L. Kuang, X. Yan, X. Tan, S. Li, and X. Yang, "Predicting taxi demand based on 3d convolutional neural network and multi-task learning," *Remote Sensing*, vol. 11, no. 11, p. 1265, 2019.
- [38] X. Geng, X. Wu, L. Zhang, Q. Yang, Y. Liu, and J. Ye, "Multimodal graph interaction for multi-graph convolution network in urban spatiotemporal forecasting," *arXiv preprint arXiv:1905.11395*, 2019.
- [39] F. Monti, D. Boscaini, J. Masci, E. Rodola, J. Svoboda, and M. M. Bronstein, "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.
- [40] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Advances in neural information processing systems*, 2016, pp. 3844–3852.
- [41] J. Zhang, Y. Zheng, and D. Qi, "Deep spatio-temporal residual networks for citywide crowd flows prediction," in *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.



Jieping Ye received his PhD degree in computer science from the University of Minnesota, Twin Cities, in 2005. He was head of Didi AI Labs, a VP of Didi Chuxing. He is also a professor of University of Michigan, Ann Arbor. His research interests include big data, machine learning, and data mining with applications in transportation and biomedicine. He has served as a Senior Program Committee/Area Chair/Program Committee Vice Chair of many conferences including NIPS, ICML, KDD, IJCAI, ICDM, and SDM. He serves as an Associate Editor of Data Mining and Knowledge Discovery, IEEE Transactions on Knowledge and Data Engineering, and IEEE Transactions on Pattern Analysis and Machine Intelligence. He won the NSF CAREER Award in 2010. His papers have been selected for the outstanding student paper at ICML in 2004, the KDD best research paper runner up in 2013, and the KDD best student paper award in 2014.



Siyuan Feng received a B.S. degree in civil engineering from Tongji University, a M.S. degree in Civil Engineering from University of California, Berkeley, and is now a Ph.D. candidate with the Department of Civil and Environmental Engineering, Hong Kong University of Science and Technology. His research interests include economical modelling of shared mobility, and deep learning in spatial-temporal traffic forecasting.



Jintao Ke received a B.S. degree in civil engineering from Zhejiang University and Ph.D. degree in Civil and Environmental Engineering from Hong Kong University of Science and Technology. He is now a Research Assistant Professor (RAP) in the Department of Logistics and Maritime Studies, Hong Kong Polytechnic University. His research interests include spatial-temporal traffic forecasting, shared mobility (ride-sourcing and ride-sharing services), transportation economy, machine learning in transportation, urban computing.



Hai Yang is currently a Chair Professor at The Hong Kong University of Science and Technology. He is internationally known as an active scholar in the field of transportation, with more than 260 papers published in SCI/SSCI indexed journals and an H-index citation rate of 56. Most of his publications appeared in leading international journals, such as Transportation Research and Transportation Science. Prof. Yang is in the Distinguished Journal Editorial Board of Transportation Research Part B: Methodological, a top journal in the field of transportation.