GODDAG: Generating Origin-Destination Flow for New Cities Via Domain Adversarial Training

Can Rong, Jie Feng, and Jingtao Ding

Abstract—**Origin-destination (OD) flow data, which reflects population mobility patterns in the city, is very important in many urban applications, such as urban planning and public resource allocation, etc. However, due to the high cost of money and time during device deployment and social surveys, it is challenging to obtain OD flow data, especially in developing cities and emerging cities where historical OD flow data is scarce. Therefore, it is necessary to investigate a method that can generate OD flow in cities where OD flow data are not available. The research on modeling population mobility in the city has a long history. Traditional gravity models, etc., are too simple to model the complex population mobility; recently proposed machine learning models and deep learning models are not applicable in cities where data are scarce because the parameters must be fitted with abundant data. To solve the problem of difficult access to OD flow data, we propose a method to learn mobility knowledge with ample data in the source city and generate OD flow data in new cities named *GODDAG* (Generating Origin-Destination Flow via Domain Adversarial Training). Our proposed method consists of two parts, one is a GNN (graph neural networks) based mobility model generating OD flow between every two regions based on regional attributes such as census and POI distribution, and the other is a domain adversarial training strategy to make the model have better transfer ability between different cities. Extensive experiments are conducted on two real-world datasets to prove the validity of our methods.**

Index Terms—**Urban computing, origin-destination, neural networks, transfer learning.**

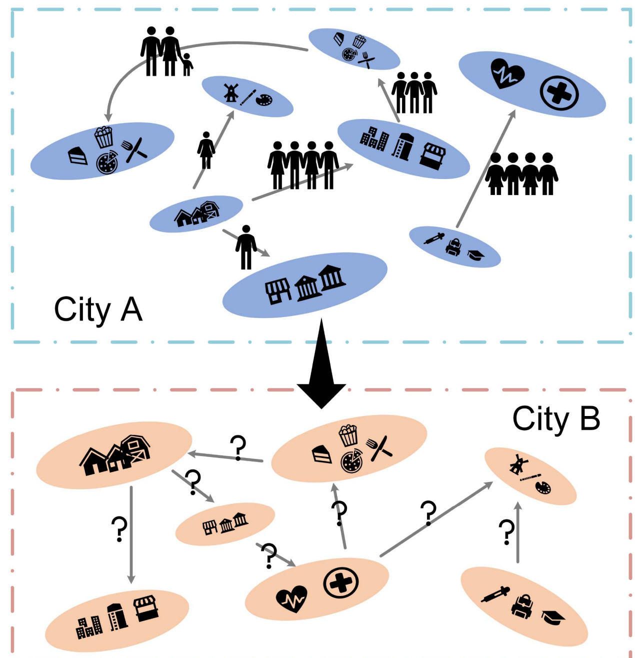

Fig. 1. OD generation schematic overview. OD flow generation for new cities where the OD data is not available.

I. Introduction

UNDERSTANDING the population mobility in the metropolis is a foundation for city managers. It is critical in many urban applications, such as public transport route planning, location-based advertisements and business location selection. In particular, OD flow containing both directions and volumes of every two regions of the city can help governors to have a comprehensive and deep insight into urban population dynamics, so that efficient urban development strategies could be constructed.

However, there are hindrances to the acquisition of OD flow data. Conventionally, OD flow data is collected through visits and surveys by census takers, which involves high time and money costs. With the development of LBS (location-based service) techniques and spreading deployment of sensors device, partial OD flow can be obtained from the aggregation of a large number of individual trajectories recorded by cellphone GPS location [1] and public transport usage records, etc. For reasons such as conflict of interest, it is difficult to obtain the complete travel tracks for aggregation into the complete OD flow. Moreover, in newly built cities where urban construction is not yet complete, the lack of devices makes it impossible to use modern technology to collect OD flow data. And it is more important to estimate the OD flow under specific urban planning, also known as the simulation of urban mobility. Generating OD flow data under certain urban development plans will help in the evaluation of the plans. After all, making changes after the city is completely built would have incalculable financial implications. Therefore, generating OD flow in urban scenarios where OD data is not available is a matter that needs to be urgently studied, as shown in Fig. 1.

Although there have been many studies [2], [3], [4], [5], [6], [7] related to population mobility modeling in cities, they

Manuscript received 12 September 2021; revised 1 February 2023; accepted 21 February 2023. Date of publication 19 April 2023; date of current version 15 September 2023. This work was supported in part by The National Key Research and Development Program of China under Grant 2022YFF0606904, and in part by the National Nature Science Foundation of China under Grants U20B2060, U21B2036, 61971267, 61972223, and 62272260. Recommended for acceptance by J. Lee. *(Corresponding author: Can Rong.)*

The authors are with the Beijing National Research Center for Information Science and Technology (BNRist), Department of Electronic Engineering, Tsinghua University, Beijing 100084, China (e-mail: rc20@mails.tsinghua.edu.cn; feng-j16@mails.tsinghua.edu.cn; dingjt15@tsinghua.org.cn).

Digital Object Identifier 10.1109/TKDE.2023.3268409

have made a limited contribution to OD generation problem. Traditional methods [2], [4] use simple physical laws to model population mobility between regions, which makes them hard to achieve good performance. Gravity model [2] is a widely used classic method inspired by Newton's law of Gravitation. The intervening opportunity model [3] attempts to describe the probability of migration of individuals. Radiation model [4] shapes the flow of population as a process of radiation and absorption. These methods with few parameters don't consider spatial interactions. In recent times, tree-based machine learning models [5], [6] perform well in modeling population flow. Random forest [5] and GBRT [6] are adopted successfully in the trip distribution modeling task. Graph Neural Networks (GNN), which have been very prevailing in the deep learning field recently, are effective in improving the performance of population flow prediction [7]. These methods mentioned above fit the parameters of the model in a single city and do not have the transfer capability to be applicable in other cities. Hence they cannot be adopted for cities with scarce data.

Although some researchers have investigated leveraging transfer learning to obtain population mobility in new cities [8], [9], [10], they spotlight the generation of individual trajectories, focusing on the generation of OD pairs while ignoring the volume of the flows between regions, and our work focuses on generating population flows from a collective perspective combined with regional level geo-context. Therefore, although these efforts have the ability to transfer knowledge, they cannot solve the OD generation problem.

It is not an easy task to generate OD flow in a new city mainly due to three challenges. *First*, the flow of people between two regions is influenced by a complex of multiple factors. Each region has its own attributes, and each region's demographics, POI distribution and occupation-resident status etc. will have various impacts. Various factors also play a different role when regions are worked as origin and destination. *Second*, the complex spatial dependency can also make it very difficult to generate OD data. Spatial interaction factors such as distance and mode of transportation can also have a significant impact on population flow and each region's production and attraction for crowd is also influenced by its surrounding neighborhoods. *Finally*, different cities have different urban structures and lifestyles, resulting in different mobility patterns. The model fitted in the source city cannot be used directly on the target city.

To address the above challenges and solve the task of OD flow generation in new target cities, we propose a framework called **G**enerating **O**rigin-**D**estination Flow via **D**omain **A**dversarial Trainin**g** (*GODDAG*). *First*, for dealing with the first and second challenges, we use GNN based spatial feature extractors combined with the cross-product feature construction technique and distance augment design to handle the diversity of influencing factors and the complexity of spatial dependencies. In particular, two GATs are used to extract regional attribute features while two GINs are used to extract spatial interaction features. *Finally*, inspired by GAN [11], we use adversarial training to learn OD flow generators that can be used in the target city using labeled data from the source city and unlabeled data from the target city. As shown in Fig 3, a feature extractor strives to extract common features shared by the source city and the target city, while a domain classifier tries to distinguish the output of the feature extractor.

The contributions of our work can be summarized as follows:

- We propose the *GODDAG*, which uses the knowledge learned from the source city to generate the OD flow of the target city, to solve the problem of lack of OD flow data in developing cities and emerging cities.
- We use GNN-based spatial feature extractors combined with cross-product feature construction technique and distance augment design to generate OD flow data. And the domain adversarial training utilized to drive the feature extractor to extract common features shared by the source and target cities so that the model can be used in the target city.
- We have done extensive experiments to prove that our method performs better than baselines with a margin of 14.5% improvement respect to *RMSE* and 5.62% improvement respect to *CPC* averagely and can solve the problem of lack of OD flow data in new cities.

The following part of this paper is consist of 5 main sections. First, we provide systematic related works reviewing to help to interpret this work in Section II. Then, we introduce details of the problem definition and key conceptions in Section III. After the problem description, the designed model and specific training strategy will be shown in Section IV. In section V, we give the experiments result and analysis the superiority of our proposed method. Finally, we conclude this paper in Section VI.

II. Related Work

In this section, we will review works on population mobility studying, graph learning methods and domain adaptation.

A. Population Mobility

Population mobility-related studies can be divided into three main categories. The first is the traditional method. This part of works focuses on using physical laws to model population movement in cities. Gravity model [12] is a classic model motivated by Newton's law of Gravitation. And there is a class of approaches called intervening opportunity approach [3], of which a very typical one is the radiation model [4]. Radiation model analogy diffusion model in terms of radiation and absorption processes. Each individual who lives in one region could be absorbed by another region. Second, machine learning models have recently achieved good performance in the trip distribution modeling domain, especially for tree-based models. Pourebrahim et al. report that the random forest model performs state-of-art in trip distribution modeling [5]. In [7], Liu et al. also test the performance of Gradient Boosting Regression Tree (GBRT). And GBRT achieves better results than random forest. Finally, Liu et al. [7] bring the graph neural networks into this research area and improve the degree of accuracy of prediction by learning a more representative region embedding. However, the works mentioned above all dive into predicting population flows in a single city. They can't solve the problem of lack of data in emerging cities. He et al. [8] provide a method to generate origin-destination pairs in a new city utilizing Transfer Component Analysis (TCA) [13] but the flow volumes of different origin-destination pairs are ignored. Moreover, there are works [9], [10] trying to generate individual trajectories in new cities in specific scenarios. They can't obtain the population mobility patterns directly from the collective perspective.

B. Graph Neural Networks

Graph neural networks have been developing very rapidly recently. ChebyNet [14] and GCN (graph convolutional networks) [15] are proposed to do graph convolution in the spectral domain. This kind of graph convolution approach can only do transductive learning, and the learned models cannot be utilized on graphs with different topologies. So they're not suitable for transferring population mobility knowledge from the source city to a target one. GraphSAGE [16] and GAT (graph attention networks) [17] are inductive learning neural network architectures that have the ability to generalize to completely unseen graphs. There are published works [7], [18], [19], [20], [21] that show the effectiveness of modeling the spatial dependencies of the cities with graph neural networks. We choose GATs and GINs as our basic graph learning model to learn the region embeddings of the city. In more detail, GATs aggregate information from neighbors using a learnable attention weight based on the relevance of node features. GINs extend the expressive power of GNNs to learn higher-order graph features beyond the 1-WL test.

C. Domain Adaptation

Domain Adaptation is an important branch of transfer learning that addresses the problem of different data distributions. It completes transfer learning by eliminating the domain shifts and finding the similarity between the source domain and the target domain. As we assume that there is no data in the target city, we mainly focus on unsupervised domain adaptation. Since Blitzer et al. [22] gave the theoretical guarantee of learning from different domains, there are a lot of effective unsupervised domain adaptation works [23], [24], [25], [26], [27], [28], [29], [30], [31] emerging in computer vision research area. Among them, domain adversarial training [24], [25], [26], [28], [30] uses an adversarial paradigm that makes the classifier unable to distinguish between the source domain and the target domain, achieving the goal of extracting the commonality between them. We absorb the intuition that learning a spatial feature extractor that could match the feature distributions in the source and the target domains by adversarial training. Our model will be described in detail in the following.

III. Preliminaries

In this section, we will give the introduction of important definitions and the problem description.

Definition 1 (Regions) The city are partitioned into N regions marked as $R_1, R_2, \ldots, R_N$, where $R_i \in \mathcal{R}$. The partition can be arbitrary, such as division referring to the street or longitude and latitude. In this study, the regions are census tracts or street blocks.

Definition 2 (Spatial Features) Spatial features include region attribute features and spatial interaction features. $x_i \in X$ is a vector that describes the regional attributes. It contains demographic information and the quantities of different types of POIs in this work. And $d_{ij} \in D$ mean the interaction features of region i and region j, such as distance, etc.

Definition 3 (Geo-Adjacency Network) We utilize undirected weighted graph $G = (V, E, \mathbf{A})$ to represent the spatial dependencies between different regions of the city, where $V = \{v_1, v_2, \ldots, v_n\}$ is the set of nodes on graph G. Each of the nodes v_i represents a specific region R_i in the city. $E = \{e_{ij} | v_i \ and \ v_j \ are \ Geographically \ adjacent.\}$ is the set of edges and $\mathbf{A} \in 0, 1^{n \times n}$ is the adjacency matrix of G.

Definition 4 (Origin-destination Flow) Origin-destination flow is the volume of population movement between two regions. We use $F_{ij} \in \mathcal{F}$ to represent the flow volume from origin R_i to destination R_j. The origin-destination flow studied in this work is the commuting flow, which means the number of people live in the origin R_i and work in the destination R_j.

Definition 5 (Origin-destination Flow Generation) Origin-destination flow prediction means predicting the flow volume set $\mathcal{F}$ of a city given the attrbute features of the regions $\{x_1, x_2, \ldots, x_N\}$, spatial interaction features $\{d_1, d_2, \ldots, d_N\}$ and the geo-adjacency network G of that city.

Definition 6 (Problem Description) The problem of origin-destination flow generation in a new target city could be formulated as a domain adaptation problem to generate $\mathcal{F}_{tar}$ with $(X_{src}, D_{src}, G_{src}, \mathcal{F}_{src})$ of the source city and $(X_{tar}, D_{tar}, G_{tar})$ of the target city given.

IV. Methods

In this section, we give a detailed introduction to the proposed method **G**enerating **O**rigin-**D**estination Flow via **D**omain **A**dversarial Trainin**g** (*GODDAG*). Our method contains an origin-destination flow generation model and a domain adversarial training framework. As shown in Fig. 2. in our method, the origin-destination flow generation model mainly consists of two parts, a GNN-based feature extractor and a specially designed flow predictor. To equip the model with the ability to transfer the knowledge learned in the source city to the target city, we use a domain adversarial learning approach to train the feature extractor and the flow predictor simultaneously. We introduce a feature classifier to complete the domain adversarial training. The feature extractor tries to extract common features shared between the source and target cities, while the feature classifier is trained to distinguish between features from the source city or the target city.

We will then present our approach in two parts, first introducing the origin-destination flow generation model, and then introducing the framework of domain adversarial training. After the process of domain adversarial training with labeled data from the source city and unlabeled data from the target city,

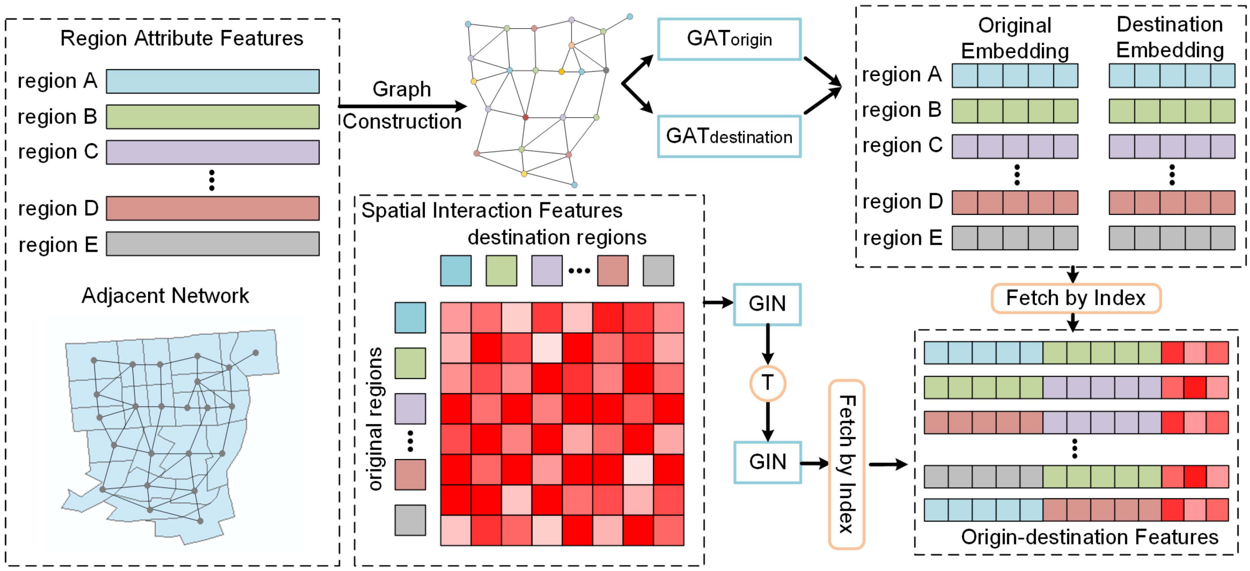

Fig. 2. Graph neural networks feature extractor architecture.

the origin-destination flow generation model could be used in the target city to generate the population flow.

A. Origin-Destination Flow Generation

The origin-destination flow generation model is composed of two parts, the GNN-based feature extractor and the flow predictor.

1) Feature Extractor: The structure of the feature extractor is shown in Fig. 2, where we use GAT (Graph Attention Networks) for region attribute features and GIN (Graph Isomorphism Networks) for region spatial interaction features, respectively.

Since the features play different roles when regions are worked as the origin or the destination, we use two GATs to generate the embeddings for regions as the origin and the embeddings for regions as the destination. The GAT employs the attention mechanism to aggregate the information from the neighbors for each node, e.g. each region in the city, to update the current states in every layer. The process of one GAT layer could be described as follows.

$$h_i' = \bigoplus_{k=1}^{K} \sigma \left(\sum_{j \in \mathcal{N}_i} \alpha_{ij} \mathbf{W}^k h_j \right) \tag{1}$$

where the h_i means the input feature vector of node i while the h_i' means the output, $\bigoplus$ represents concatenation, K is the number of heads in *multi-head attention*, α_{ij}^k is the attention weight between node i and node j of k-th head, $\mathbf{W}^k$ is the learnable weight matrix of linear transformation and $j \in \mathcal{N}_i$ denotes node j is one of node i's neighbors. The computation of attention weight is as follows.

$$\alpha_{ij} = \frac{exp(LeakyReLU(\Theta[\mathbf{W}h_i || \mathbf{W}h_j]))}{\sum_{n \in \mathcal{N}} exp(LeakyReLU(\Theta[\mathbf{W}h_i || \mathbf{W}h_n]))} \tag{2}$$

where Θ and $\mathbf{W}$ are learnable parameters. It should be noticed that the h here means the spatial interaction features of every two regions, i.e. the distance between them.

The spatial interaction features of regions can be considered as a two-perspective feature vector, with one perspective representing the origin and another representing the destination, as shown in Fig. 2. So we absorb the idea of MGCNN [32] that defines row- and column- graphs and use GINs, which have the ability of inductive learning (Since we want to transfer our model between different cities.), to deal with the information processing process in origin and destination perspectives. The **T** operation between two GIN modules is the transposition. The operation process of one GIN layer can be expressed as follows.

$$h_i' = MLP \left((1+\varepsilon) h_i + \frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} h_j \right) \tag{3}$$

where ε is a learnable parameter.

And finally, the original region embedding, the destination region embedding and the interaction embedding, i.e. edge embedding, are concatenated together to obtain the final origin-destination features. The final embedding of the origin-destination pair of region i and region j can be described as follows.

$$h_{ij} = h_i^{(origin)} \bigoplus h_j^{(destination)} \bigoplus e_{ij} \tag{4}$$

where $\bigoplus e_{ij}$ means the spatial interaction embedding between region i and region j.

2) Flow Predictor: We adopt Multi-layer Perceptron (MLP) as the basic model and add cross-product transformation technology and distance information augment leveraging. After the origin-destination embedding is obtained from the feature extractor, to enhance the influence of spatial interaction features, we perform a cross-product transformation on region embeddings and spatial interaction features, and concatenate the new

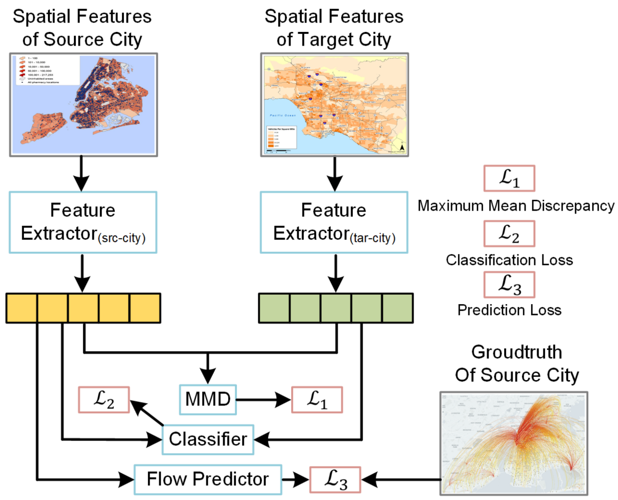

Fig. 3. Domain adversarial training.

second-order features and origin-destination embedding. The cross-product transformation we used here can be denoted as follows.

$$s_{ij} = h_{i\&j} \bigotimes e_{ij} = h_{i\&j} \cdot e_{ij}^T \tag{5}$$

where $h_{i\&j} \in \mathbb{R}^{d_1 \times 1}$ means the region embedding (origin and destination), $e_{ij} \in \mathbb{R}^{d_2 \times 1}$ means the spatial interaction embedding, $s_{ij} \in \mathbb{R}^{d_1 \times d_2}$ means the secord-order features.

Referring to the traditional studies on population mobility, we learn a domain knowledge that the flow of people obeys the power law with respect to distance. So we select part of the origin-destination pairs that have large flow volumes with the same distance to fit the parameters of the power model of flow and distance. And we use this simple power model as a threshold to normalize the flow volume of a specific distance. The operation could be described as follows,

$$f_{ij} = \min(a \cdot {d_{ij}}^b + c, MLP(s_{ij})), \tag{6}$$

where a, b, c are the parameters of the power model fitted by the selected origin-destination flows part, d_{ij} means the distance between region i and region j and f_{ij} means the final generated flow volume. The generation loss is described as follows,

$$\mathcal{L}_3 = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} (f_{ij} - \hat{f}_{ij})^2. \tag{7}$$

B. Adversarial Training

We introduce a classifier to implement the adversarial training, as shown in Fig. 3. The feature extractors try to access common features shared by the source city and target city so that the features extracted in the target city could also be used by the flow predictor trained by data of the source city. In adversarial training, feature extractors try to fool the classifier while the classifier tries to distinguish the feature from the source or the target city. The flow predictor is trained simultaneously.

1) Common Feature Extraction: As shown in Fig. 3, we use two feature extractors to extract features from the source city and the target city, respectively. And we use *MMD* (Maximum Mean Discrepancy) as an indicator ($\mathcal{L}_1 = MMD$) to measure the difference between the features from the source city and the target city. The computation of MMD is as follows.

$$\begin{aligned} MMD = & \frac{1}{n_s^2} \sum_{i=1}^{n_s} \sum_{j=1}^{n_s} k(\phi(x_i^s), \phi(x_j^s)) \\ & + \frac{1}{n_t^2} \sum_{i=1}^{n_t} \sum_{j=1}^{n_t} k(\phi(x_i^t), \phi(x_j^t)) \\ & - \frac{2}{n_s n_t} \sum_{i=1}^{n_s} \sum_{j=1}^{n_t} k(\phi(x_i^s), \phi(x_j^t)), \end{aligned} \tag{8}$$

where k presents the kernel function, which could be the following formula.

$$k(x, x') = e^{-\frac{\|x-x'\|^2}{2\alpha^2}} \tag{9}$$

2) Domain Classification: The classifier shown in Fig. 3 is worked as the discriminator in the adversarial training framework. We adopt MLP (Multi-layer Perceptrons) with the final layer replaced by a $softmax$ layer to be the classifier model and use Cross-Entropy Loss, as shown follow.

$$\mathcal{L}_2 = \sum_{c=1}^{2} \sum_{i=1}^{B} -y_{c,i} log_2(p_{c,i}) \tag{10}$$

where B is the batch size. During the training process, gradient descent is performed on the feature extractor and domain classifier iteratively.

3) Training Algorithm: We outline the adversarial training procedure for the origin-destination flow generation model in Algorithm 1. The feature extractor and flow predictor are trained simultaneously during the adversarial training with the help of a classifier.

V. Experiments

In this section, we conduct extensive experiments to answer the following 4 research questions:

- *RQ1:* Can our proposed method be effective in generating OD flow data in new cities?
- *RQ2:* How does the design of each part of our method improve performance?
- *RQ3:* How do the hyper-parameters of the model affect performance?
- *RQ4:* Are the generated population mobility flows spatially distributed in a practical way?

A. Data Description

We collected real-world data and constructed two commuting Origin-destination generation datasets in the United States of America and China. And the experiments are conducted on these two datasets. The datasets are described as follows.

- *America* We form the *regions' spatial features* based on the demographic data, POIs (points of interest) distribution and occupational and residential data. The demographic data

Algorithm 1: Adversarial Training Algorithm.

Input:
Regional features of the source city X^{src}.
Regional features of the target city X^{tar}.
Spatial interaction features of the source city D^{src}.
Spatial interaction features of the target city D^{tar}.
Origin-destination flow of the source city F^{src}.
Count n_G of gradient descents of feature extractor.
Count n_D of gradient descents of the classifier.

Output:
Learned origin-destination flow generation model of the target city.

```
Construct the graphs according to the geo-adjacent network of the source and target city
Initialize the learnable parameters of the model
repeat
  for i^th batch of all batches of data do
    for n_G times do
      Extract origin-destination features h_i^src from X_i^src by feature extractor
      Extract origin-destination features h_j^tar from X_i^tar by feature extractor
      Compute MMD of h_i^src and h_j^tar
      Optimize feature extractor by MMD
    end for
    for n_D times do
      Extract origin-destination features h_i^src from X_i^src by feature extractor
      Extract origin-destination features h_j^tar from X_i^tar by feature extractor
      Compute L_2 of h_i^src and h_j^tar
      Optimize classifier by L_2
    end for
    Generate origin-destination flow F̂^src by feature extractor and flow predictor
    Compute generation loss L_1 based on F^src and F̂^src
    Optimize feature extractor and flow predictor by L_1
  end for
until Generation loss L_1 converge
```

are collected from the yearly American Community Survey (ACS). POI distribution is obtained by crawling the POI data from OSM (OpenStreetMap) [33] and then statistics. Occupational and residential data are from Longitudinal Employer-Household Dynamics Origin-Destination Employment Statistics (LODES) dataset. The *network topologies of the cities*, which is used to build the graph are constructed by the Cartographic Boundary dataset from Census Bureau of the United States. As with Occupational and residential data, the *commuting flow* data is also included in LODES dataset.

- *China.* In China, we use two cities, Beijing and Shanghai, to conduct experiments to validate the validity of our proposed model. The *regions' spatial features* include demographic information and POI distribution. The Chinese demographic information is from National Bureau of Statistics of China. POIs distribution is collected from a Chinese map service provider., one of the biggest LBS (location-based service) providers in China. As for the *network topologies of the cities*, we crawled the web for street block boundary data in Beijing and Shanghai. The *commuting flow* data is obtained by processing GPS track data provided by one of the major mobile application providers in China.

B. Baselines

- *Gravity Model.* Gravity model (GM) [12] is a traditional spatial interaction model which is inspired by Newton's law of Gravitation. The spatial features of one region work as the mass and population flow between two regions follow the power-law distance decay.
- *XGBregressor.* XGBoost (XGB) [34] is a gradient boost decision tree-designed machine learning framework. And it is considered the state-of-art tree-based model.
- *Gradient Boosting Regression Tree.* GBRT [6] combines the ensemble learning and gradient boosting technique to improve the performance of regression trees.
- *Random Forest.* Random Forest (RF) is a traditional machine learning method containing multiple tree models. Pourebrahim et al. [5] recently pointed out that it works well with trip distribution modeling.
- *GAT.* GAT (Graph attention networks) [17] is a neural network architectures presented by Veličković et al. dealing with graph-structured data.
- *GEML.* Liu et al. proposed GMEL [7] which uses graph neural networks and multi-task learning to learn the embeddings of regions and uses the GBRT as the predictor to obtain the commuting flows based on the embeddings.

C. Evaluation Metrics

We use Root Mean Square Error (RMSE), Symmetric Mean Absolute Percentage Error (SMAPE) and Common Part of Commuters (CPC) as the performance evaluation metrics of our experiments.

$$RMSE = \sqrt{\frac{1}{|X|}\sum_{i,j}||\mathbf{X}_{ij} - \hat{\mathbf{X}}_{ij}||_2^2}, \quad (11)$$

$$SMAPE = \frac{100\%}{|X|}\sum_{i,j}\frac{X_{ij} - \hat{X_{ij}}}{(|X_{ij}| + |\hat{X_{ij}}|)/2}, \quad (12)$$

$$CPC = \frac{2\sum_{i,j} min(X_{ij}, \hat{X_{ij}})}{\sum_{i,j}\hat{X_i}j + \sum_{i,j} X_{ij}}, \quad (13)$$

RMSE is a popular metric applied in regression problems. SMAPE is an error indicator that shows the error as a percentage of the true value. CPC is a measure of the common part of the two flows and is widely used in origin-destination flow related studies.

TABLE I
OVERALL PERFORMANCE ON AMERICA DATASET. THE LOWER THE RMSE, SMAPE AND THE HIGHER THE CPC, THE BETTER THE PERFORMANCE

Data	Metrics	GM	XGB	GBRT	RF	GAT	GMEL	ours
NYC→ LA	RMSE	22.85	21.21	6.92	9.15	7.23	7.21	**5.48**
	SMAPE	0.92	0.90	0.93	0.96	0.95	0.93	**0.73**
	CPC	0.39	0.41	0.54	0.50	0.54	0.54	**0.59**
LA→ NYC	RMSE	12.62	7.73	6.50	8.91	6.39	6.32	**5.14**
	SMAPE	0.72	0.58	0.72	0.56	0.70	0.71	**0.53**
	CPC	0.45	0.59	0.57	0.57	0.62	0.63	**0.65**
NYC→ DC	RMSE	32.51	26.33	30.89	28.00	25.26	25.17	**21.33**
	SMAPE	0.94	0.79	0.73	0.81	0.78	0.78	**0.72**
	CPC	0.41	0.51	0.40	0.52	0.47	0.50	**0.56**
NYC→ San	RMSE	32.80	41.62	29.27	25.57	23.98	23.14	**20.99**
	SMAPE	0.95	1.02	0.92	0.97	0.96	0.96	**0.89**
	CPC	0.44	0.49	0.49	0.57	0.53	0.52	**0.58**
NYC→ Chi	RMSE	18.06	16.49	11.96	11.33	5.29	5.11	**4.88**
	SMAPE	0.92	0.69	1.05	0.65	0.53	0.54	**0.52**
	CPC	0.40	0.57	0.50	0.60	0.67	0.68	**0.70**
NYC→ Sea	RMSE	22.37	21.91	17.26	16.59	21.13	22.42	**13.51**
	SMAPE	0.96	0.91	0.86	0.59	0.80	0.75	**0.54**
	CPC	0.37	0.54	0.55	0.56	0.50	0.52	**0.61**

TABLE II
OVERALL PERFORMANCE ON CHINA DATASET

	Beijing→ $Shanghai$			Shanghai→ $Beijing$		
Metrics	RMSE	SMAPE	CPC	RMSE	SMAPE	CPC
GM	863.15	0.92	0.39	523.13	0.97	0.34
XGB	566.30	0.82	0.52	371.07	0.81	0.46
GBRT	611.84	0.84	0.46	419.69	0.96	0.47
RF	420.65	0.77	0.54	329.55	0.79	0.55
GAT	371.54	0.78	0.54	276.10	0.72	0.55
GMEL	369.71	0.73	0.57	263.94	0.74	0.57
GODDAG	**324.02**	**0.69**	**0.61**	**183.74**	**0.72**	**0.60**

D. Experiment Settings

In experiments, the regional attributes data, regional interaction data from the source and target cities and origin-destination flow data from the source city are used to train the origin-destination flow generation model. The origin-destination flow data from the target city are utilized to test the transfer performance of the model. The tree-based baselines are implemented by *scikit-learn* and the hyper-parameters are searched by built-in grid search tools. In all graph models, the hidden dimension is set to 64 and the number of layers is set to 3. And we use the ADAM optimizer to do gradient descent.

E. Performance Comparison (RQ1)

In this section, we will show the comparison between our proposed method and several baselines on two real-world datasets introduced above. From Tables I and II, we can see that it is feasible to generate origin-destination flow data in the target city using the knowledge learned from the source city. Table I gives the experimental results on the America dataset under RMSE, SMAPE and CPC. We have done extensive experiments with six cities in the U.S. as transfer learning scenarios, and the experimental results prove that our method achieves the best and most stable performance on all metrics. The traditional gravity mode performs poorly because it is based on simple laws of physics and cannot model the complex pattern of population mobility in cities. The tree model-based machine learning approach performs better than the gravity model, thanks to its ability to model nonlinearities. The three tree-based models perform differently under different transfer scenarios, with GBDT and random forest performing more robustly than XGBregressor. The GNN-based approaches are superior to the tree-based approach in most scenarios because they utilize not only information about the origin and destination, but also information about the neighbors of the origin and destination. GAT is a GNN that uses the attention mechanism to aggregate neighbor information to get representative embeddings. GMEL adds multi-task learning of in/out flow prediction to the GAT and combines it with GBRT to make final predictions, so it performs better than the GAT alone. Our method, which incorporates domain adversarial training, shows a significant improvement in transfer learning capability and performs very consistently in *all scenarios and metrics*. From the experimental results of the Chinese dataset, as shown in Table II, we can get the same conclusion as Table I. Since the granularity of street block division in China is relatively coarse, the RMSE will be relatively larger, but it can be seen from SMAPE and CPC that the performance of baselines and our proposed method is consistent with the results on the America dataset.

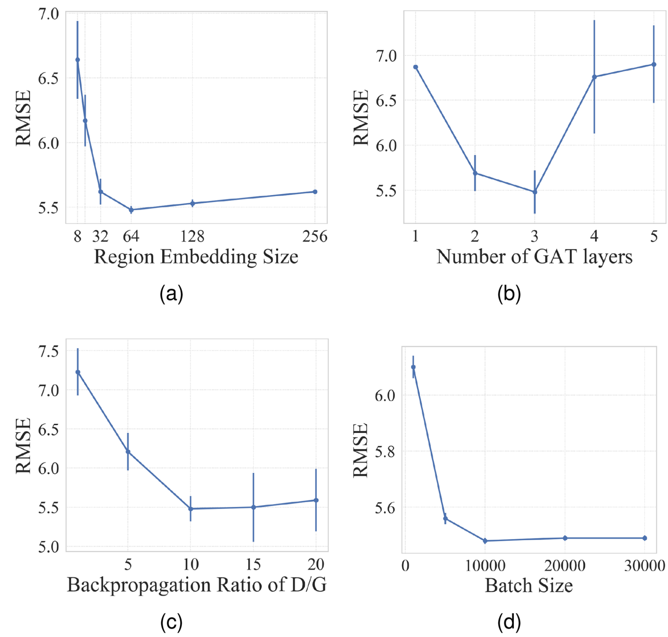

Fig. 4. Effects of hyper-parameters.

F. Ablation Study (RQ2)

We also take ablation study experiments between New York City and Los Angeles to check the validity of the design of each part of the proposed method. The basis of the approach is the GAT, to which designs are added

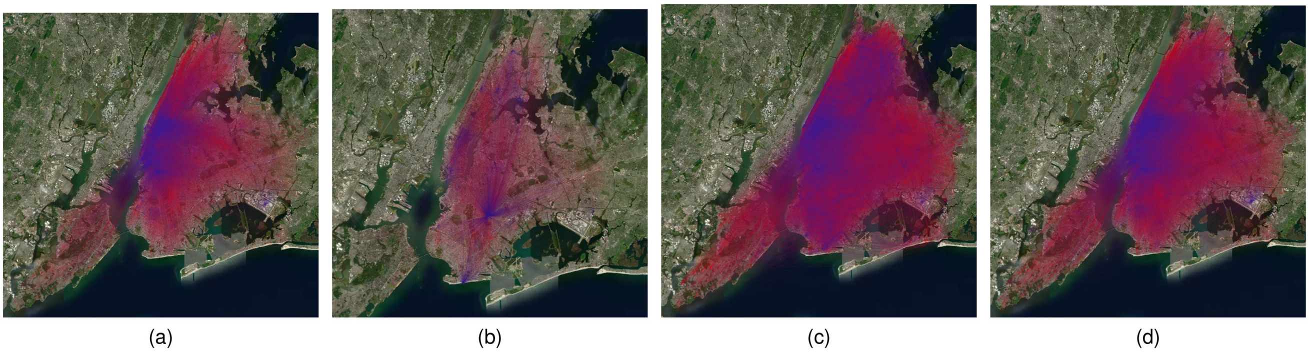

Fig. 5. Visualization of the generated OD flow of baselines and our model and ground truth. The diagrams from left to right are from (a) ground truth, (b) gravity model, (c) graph attention networks and (d) our proposed method.

TABLE III
ABLATION STUDY

	NYC→ LA			LA→ NYC		
Metrics	RMSE	SMAPE	CPC	RMSE	SMAPE	CPC
GAT	7.23	0.95	0.54	6.79	0.70	0.61
ours-v1	6.81	0.83	0.54	6.31	0.64	0.62
ours-v2	6.60	0.79	0.57	5.74	0.64	0.62
ours-v3	6.18	0.77	0.57	5.49	0.58	0.63
ours-v4	5.52	0.74	0.59	5.22	0.54	0.64
ours	5.48	0.73	0.59	5.14	0.53	0.65

incrementally to improve performance. The ablation studies adopt the same experiment settings introduced before. The results are shown in Table III. In Table III, ours-v1 means GAT+2DGNN, ours-v2 means GAT+2DGNN+CP, ours-v3 means GAT+2DGNN+CP+dis_aug, ours-v4 means GAT+2DGNN+CP+dis_aug+GRL, ours means our complete model. The details are as follows.

ours-v1: This version can show the improvement caused by 2DGNN. 2DGNN means that two GINs are used one after another to process the spatial interaction features. The purpose of this addition is to further explore the semantics of spatial interaction features. The previous studies [5], [7] only extract regional attribute features and ignore edge-wise spatial interaction features. Table III gives a report about the quantitative performance gain in all metrics mentioned above.

ours-v2: This design of experiments checks the validity of cross-product (CP) operation. The operation of CP is used widely in the recommender system to utilize the second-order features to get better achievement of the model. We do cross-product on regional attribute features with spatial interaction features to obtain the second-order interaction features of regional attributes and interactions. This operation further reduces the generation error of origin-destination flow data according to Table III.

ours-v3: This results show the function of distance augment (dis_aug). We use the power-law model obtained in traditional studies to further enhance the stability of the generated data by using distance to limit the size of the generated flow volume. This can also bring some performance gains.

ours-v4: We replace the adversarial training with gradient reverse layers (GRL) to check performance gain from adversarial paradigm. We remove loss1 in Fig. 3 and use gradient reverse layer proposed in [23] to examine the effect of MMD loss. And from the last two rows in Table III we can see that domain adversarial training could improve the transfer capability of the model and explicitly using MMD to drive the feature extractor to extract similar features is effective.

G. Hyper-Parameters Analysis (RQ3)

We perform a detailed analysis of the sensitivity of the hyper-parameters of the proposed method. Fig. 4(a) shows the effect of embedding size of each region after processing of GAT in the feature extractor. We can see that there exists an optimal value. Too small embedding size will reduce the effectiveness of region embeddings while too large embedding size leads to hard training. As shown in Fig. 4(b), GAT suffers from the problem of oversmoothing, so more than 3 layers of GAT will bring performance degradation. To maintain the balance between the classifier and the feature extractor in adversarial training, in this method, the classifier needs to be assigned 10 times the number of gradient descents as the feature extractor according to 4(c). As we can see from 4(d), the impact of batch size on adversarial training is also relatively large, mainly affecting the stability of MMD. The larger the MMD loss obtained, the more it can express the true source domain and target domain distinction, while too large does not bring more performance improvement.

H. Visualization (RQ4)

To further prove the validity of our proposed OD generation framework and give an intuitive presentation, we show a bit of visualization of the comparison of generated OD flow from our model and baselines with the ground truth. We choose New York City as our target city and Los Angeles as the source city. As shown in Fig. 5, the visualization of OD flows is presented as an arc diagram, where an arc line means that there exists flow between the locations at two ends of the line. The red color means the origin region of the flow while the blue means the destination. The brightness of the lines gives information on OD flow volumes, which is the larger the population flow between the origin and destination the brighter the line. The images are, left to right in order, the ground truth of the OD flow in New York City and the Generated OD flow from the gravity model, graph attention networks and *GODDAG*.

From Fig. 5(a) we can see that people usually work in the city center and live on the outskirts of the city. But people generally prefer to live closer to the city center to avoid a long commute, so the farther away from the city center, the fewer people live there. From Fig. 5(b), the gravity model produced relatively large shifts in the distribution of the generated OD flow and even got the downtown location wrong away from Manhattan district because of its poor modeling capabilities. The graph networks-based models all give the right position of the centralized workplace in the city center. However, without transfer learning techniques, the destinations of many flows are spread in Bronx, the north of New York City, and Brooklyn, the south side, which is conflict with the ground truth. Surprisingly, the OD flow generated by our method has almost the same spatial distribution with the ground truth. The visualization gives further confirmation of the validity of our method.

On the other hand, from the visualized results we can see that all these methods give larger predictions for OD of small volume, which indicates that the long-tail distribution of OD is a difficult problem to solve, and this is also a direction of our future research.

VI. Conclusion

In this article, we investigate solving the problem of the lack of OD flow data in emerging cities by transfer learning. By proposing a method called *GODDAG* based on graph learning and domain adversarial training, we transfer the knowledge learned from the source city to the target city to generate OD flow data. In the training process, a classifier trying to distinguish the feature extracted from the source city or the target city drives the feature extractors to extract common features shared in two cities so that the flow predictor trained by features of the source city could be used by the target city. We construct two real-world datasets by collecting public data from multi-source. Extensive experiments have been done on these two datasets to prove the validity of our method.

References

[1] F. Calabrese, G. Di Lorenzo, L. Liu, and C. Ratti, "Estimating Origin-Destination Flows Using Mobile Phone Location Data," in *Proc. IEEE Pervasive Comput.*, vol. 10, no. 4, pp. 36–44, April 2011, doi: 10.1109/MPRV.2011.41.

[2] H. C. Carey, *Principles of Social Science*, vol. 3. Philadelphia, PA, USA: JB Lippincott & Company, 1871.

[3] S. A. Stouffer, "Intervening opportunities: A theory relating to mobility and distance', 1merican sociological review," *New York*, vol. 5, pp. 6–845, 1940.

[4] F. Simini, M. C. González, A. Maritan, and A.-L. Barabási, "A universal model for mobility and migration patterns," *Nature*, vol. 484, no. 7392, pp. 96–100, 2012.

[5] N. Pourebrahim, S. Sultana, A. Niakanlahiji, and J.-C. Thill, "Trip distribution modeling with twitter data," *Comput., Environ. Urban Syst.*, vol. 77, 2019, Art. no. 101354.

[6] C. Robinson and B. Dilkina, "A machine learning approach to modeling human migration," in *Proc. 1st ACM SIGCAS Conf. Comput. Sustain. Societies*, 2018, pp. 1–8.

[7] Z. Liu, F. Miranda, W. Xiong, J. Yang, Q. Wang, and C. Silva, "Learning geo-contextual embeddings for commuting flow prediction," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 808–816.

[8] T. He et al., "What is the human mobility in a new city: Transfer mobility knowledge across cities," in *Proc. Web Conf.*, 2020, pp. 1355–1365.

[9] Y. Pang, K. Tsubouchi, T. Yabe, and Y. Sekimoto, "Intercity simulation of human mobility at rare events via reinforcement learning," in *Proc. 28th Int. Conf. Adv. Geographic Inf. Syst.*, 2020, pp. 293–302.

[10] R. Jiang et al., "Transfer urban human mobility via POI embedding over multiple cities," *ACM Trans. Data Sci.*, vol. 2, no. 1, pp. 1–26, 2021.

[11] I. J. Goodfellow et al., "Generative adversarial networks," 2014, *arXiv:1406.2661*.

[12] M. Lenormand, A. Bassolas, and J. J. Ramasco, "Systematic comparison of trip distribution laws and models," *J. Transport Geogr.*, vol. 51, pp. 158–169, 2016.

[13] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, "Domain adaptation via transfer component analysis," *IEEE Trans. Neural Netw.*, vol. 22, no. 2, pp. 199–210, Feb. 2011.

[14] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," 2016, *arXiv:1606.09375*.

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

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

[17] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," 2017, *arXiv:1710.10903*.

[18] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," 2017, *arXiv: 1707.01926*.

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

[20] H. Yao, X. Tang, H. Wei, G. Zheng, and Z. Li, "Revisiting spatial-temporal similarity: A deep learning framework for traffic prediction," in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 5668–5675.

[21] H. Shi et al., "Predicting origin-destination flow via multi-perspective graph convolutional network," in *Proc. IEEE 36th Int. Conf. Data Eng.*, 2020, pp. 1818–1821.

[22] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan, "A theory of learning from different domains," *Mach. Learn.*, vol. 79, no. 1, pp. 151–175, 2010.

[23] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2015, pp. 1180–1189.

[24] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 7167–7176.

[25] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb, "Learning from simulated and unsupervised images through adversarial training," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2107–2116.

[26] J. Hoffman et al., "CyCADA: Cycle-consistent adversarial domain adaptation," in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2018, pp. 1989–1998.

[27] M. Long, H. Zhu, J. Wang, and M. I. Jordan, "Deep transfer learning with joint adaptation networks," in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2017, pp. 2208–2217.

[28] M. Long, Z. Cao, J. Wang, and M. I. Jordan, "Conditional adversarial domain adaptation," 2017, *arXiv:1705.10667*.

[29] K. Saito, K. Watanabe, Y. Ushiku, and T. Harada, "Maximum classifier discrepancy for unsupervised domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 3723–3732.

[30] S. Sankaranarayanan, Y. Balaji, C. D. Castillo, and R. Chellappa, "Generate to adapt: Aligning domains using generative adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 8503–8512.

[31] Y. Zhang, T. Liu, M. Long, and M. Jordan, "Bridging theory and algorithm for domain adaptation," in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2019, pp. 7404–7413.

[32] F. Monti, M. M. Bronstein, and X. Bresson, "Geometric matrix completion with recurrent multi-graph neural networks," 2017, *arXiv: 1704.06803*.

[33] OpenStreetMap contributors, "Planet dump retrieved from https://planet.osm.org," 2017. [Online]. Available: https://www.openstreetmap.org

[34] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2016, pp. 785–794.

Can Rong received the MEng degree in software engineering from Peking University in 2017. He is currently working toward the PhD degree with the Department of Electronic Engineering of Tsinghua University, advised by Prof. Yong Li. His research areas include urban computing and spatio-temporal data mining. He has recently been working on how to apply graph neural networks to model the interactive characteristics of regions in cities.

Jie Feng received the BE degree in electrical engineering from Tsinghua University in 2016. He is currently working toward the PhD degree with the Department of Electronic Engineering of Tsinghua University, advised by Prof. Yong Li. His research interest falls in the area of spatial-temporal data mining. He currently works on applying deep learning methods into the spatial-temporal data mining filed to improve the performance of the practical model in many challenging practical tasks like mobility prediction and flow forecasting.

Jingtao Ding received the BS degrees in electronic engineering and the PhD degree in electronic engineering from Tsinghua University, Beijing, China, in 2015 and 2020, respectively. He is currently a post-doctoral research fellow with the Department of Electronic Engineering, Tsinghua University. His research interests include mobile computing, spatiotemporal data mining and user behavior modeling. He has more than 30 publications in journals and conferences such as *IEEE Transactions on Knowledge and Data Engineering*, *ACM Transactions on Information Systems*, KDD, NeurIPS, WWW, ICLR, SIGIR, IJCAI, etc.