

What is the Human Mobility in a New City: Transfer Mobility Knowledge Across Cities

Tianfu He^{1,2,3}, Jie Bao^{2,3}, Ruiyuan Li^{4,2,3}, Sijie Ruan^{4,2,3}, Yanhua Li⁵, Li Song⁶, Hui He^{1,*}, Yu Zheng^{2,3}

¹Harbin Institute of Technology; ²JD Intelligent City Research; ³JD Intelligent City Research;

⁴Xidian University; ⁵Worcester Polytechnic Institute; ⁶Meituan-Dianping Inc.

Tianfu.D.He@outlook.com;{baojie,ruiyuan.li,ruansijie}@jd.com;yli15@wpi.edu;

songli04@meituan.com;hehui@hit.edu.cn;msyuzheng@outlook.com

ABSTRACT

With the advances of web-of-things, human mobility, e.g., GPS trajectories of vehicles, sharing bikes, and mobile devices, reflects people's travel patterns and preferences, which are especially crucial for urban applications such as urban planning and business location selection. However, collecting a large set of human mobility data is not easy because of the privacy and commercial concerns, as well as the high cost to deploy sensors and a long time to collect the data, especially in newly developed cities. Realizing this, in this paper, based on the intuition that the human mobility is driven by the mobility intentions reflected by the origin and destination (or OD) features, as well as the preference to select the path between them, we investigate the problem to generate mobility data for a new target city, by transferring knowledge from mobility data and multi-source data of the source cities. Our framework contains three main stages: 1) *mobility intention transfer*, which learns a latent unified mobility intention distribution across the source cities, and transfers the model of the distribution to the target city; 2) *OD generation*, which generates the OD pairs in the target city based on the transferred mobility intention model, and 3) *path generation*, which generates the paths for each OD pair, based on a utility model learned from the real trajectory data in the source cities. Also, a demo of our trajectory generator is publicly available online for two city regions. Extensive experiment results over four regions in China validate the effectiveness of the proposed solution. Besides, an on-field case study is presented in a newly developed region, i.e., Xiongan, China. With the generated trajectories in the new city, many trajectory mining techniques can be applied.

CCS CONCEPTS

• Information systems → Spatial-temporal systems.

KEYWORDS

Web of Things, Trajectory Data Mining, Urban Computing

ACM Reference Format:

Tianfu He^{1,2,3}, Jie Bao^{2,3}, Ruiyuan Li^{4,2,3}, Sijie Ruan^{4,2,3}, Yanhua Li⁵, Li Song⁶, Hui He^{1,*}, Yu Zheng^{2,3}. 2020. What is the Human Mobility in a New City: Transfer Mobility Knowledge Across Cities. In *Proceedings of The Web Conference 2020 (WWW '20)*, April 20–24, 2020, Taipei, Taiwan. ACM, New York, NY, USA, 11 pages. <https://doi.org/10.1145/3366423.3380210>

This paper is published under the Creative Commons Attribution 4.0 International (CC-BY 4.0) license. Authors reserve their rights to disseminate the work on their personal and corporate Web sites with the appropriate attribution.

WWW '20, April 20–24, 2020, Taipei, Taiwan

© 2020 IW3C2 (International World Wide Web Conference Committee), published under Creative Commons CC-BY 4.0 License.

ACM ISBN 978-1-4503-7023-3/20/04.

<https://doi.org/10.1145/3366423.3380210>

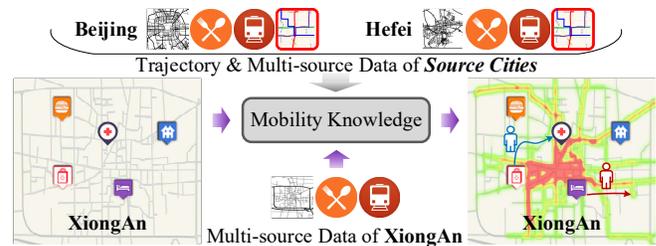


Figure 1: Mobility Knowledge Transfer across Cities.

1 INTRODUCTION

Human mobility in an urban area is represented by the trajectories (e.g., in vehicles, bikes, and mobile devices). In advances in web-of-things services like ride-hailing and sharing bikes in recent years, the massive spatial trajectory data are collected, and many exciting trajectory mining based techniques are emerging, e.g. urban planning [2, 43], business location selection [22], and etc. For a new city, government staffs and entrepreneurs can get constructive suggestions from these techniques to make their decisions, but the first thing is to acquire the human mobility data.

However, the acquisition of mobility data is not an easy task for three reasons: 1) *People's privacy concerns*. Some research works, e.g. [3], show that people prefer not to share their location information. 2) *Commercial concerns*. The high-quality mobility data are collected and owned by only a few companies that provide popular location-based service (LBS) applications. As a result, despite the value of mobility data in business location selection, companies of interest, e.g. the catering chain, can hardly get access to these data. 3) *Deployment expense*. Even for the government, it is time-consuming and costly to develop urban sensors and collect data in large scale. It should also be noted that the urban applications, such as urban planning and business site selection, are more meaningful and beneficial, during the early development phase of an urban region, as it is more costly to change the plan and site locations when the area is already developed. Therefore, it is of good value to infer the human mobility of a city.

There are many research works focus on modeling human mobility or urban transfer learning. [36] proposes a Recurrent Neural Network based model to encode the trajectory and performs a prediction task to get the next step of the trajectory. Besides, [16, 17, 23, 28, 38, 44] investigate generating the trajectories of the city using deep learning methods. However, all the above works focus on the mobility of the same city. Moreover, the existing urban

*Hui He is the corresponding author.

transfer learning frameworks are designed to fill missing values [34] or transfer spatial hotspots [25] to a new city, which are very different from our problem settings. The most similar work to ours is [7], which generates the mobility of a new city when rare events happen. Nevertheless, in this work, the large-scale mobility data in normal days is known in the new city, which is also essentially a quite different problem from ours. To the best of our knowledge, our work is the first to generate spatial trajectories in new cities without mobility data in the new cities.

In this paper, we specifically focus on generating the sharing bikes' trajectory data in a new city. Sharing bike service is a typical use-case of web-of-things. The sharing bikes are equipped with Bluetooth, GPS sensors and the network. The user gets access to the bikes using the smartphone. The detailed trajectory of the whole trip is recorded, which reflects the people's short-range mobility. Compared to the long-range high-speed commutes (e.g. taxi, bus and metro trips), short-range trajectories are more important for both the planning of bike lanes/sidewalks [2, 14] and location selection of like chain stores and charging stations for electric automobiles [22], since these applications are generally designed to serve the pedestrians or bike riders passing by.

In detail, we develop a mobility transfer system, which can generate trajectories for some *target* city by a unified mobility knowledge model learned from the multi-source dataset from the *source* cities that have mobility data. Note that in this work, we focus on the spatial distribution of the trajectories over a long period of time (e.g. 3 months), which is also the key focus in other mobility researches.

Figure 1 gives an overview of the procedure, where the system learns a unified mobility knowledge model based on the spatial features and trajectories of the sources cities (i.e., Beijing and Hefei at the top of the figure). After that, for a target city (i.e., Xiongan in the example), the multi-source data is fed to the unified mobility knowledge model to generate the trajectories in the area (as the heatmap on the right).



Figure 2: The Three Mental Stages of a Trip.

The most important task here is to identify a unified transferable model to reflect the trajectories in different cities. We realize that there are three main steps to generate a trajectory, which is demonstrated in Figure 2: 1) a mobility is driven by an intention, e.g., shopping (Figure 2a); 2) a destination is selected to fulfill the intention, e.g., a plaza (i.e., circle in Figure 2b); and 3) a path is selected to connect the origin and destination (i.e., red line in Figure 2c).

As the conceptual procedure to generate the trajectory is universal for most of the users in any cities, it is possible to build a universal and transferable model based on these intuitions. However, it is still a non-trivial task to generate trajectories in target cities: 1) *diversity of city styles*. Due to the differences in lifestyles,

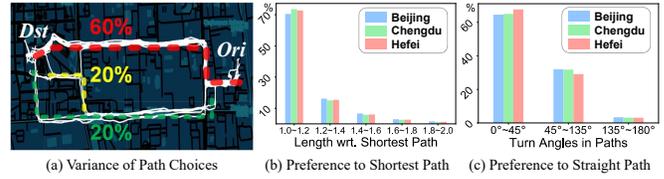


Figure 3: Path Preference Observations.

public transportation services, and more, the explicit travel intention distributions are different across cities; 2) *enormous OD pair space*. It is very hard to build an end-to-end model that directly generates OD pairs in the target city, as the space of OD pair candidates is huge; 3) *the diverse path preference*. Even for the same OD pair, people travel it with different paths, as people have different preferences in choosing the paths. Figure 3a gives an example with real-world trajectories, where three main paths are demonstrated with different probability distributions.

We address the challenges in our system with three main techniques: 1) *mobility intention transfer*. We use a domain generalization technique [27] to learn an adaptation function to project the features of OD pairs of the source cities to a latent space, where the OD feature distributions of all cities (including the source and target cities) are similar to each other. In this way, the differences between cities are minimized and the model is generalized to transfer mobility knowledge to the target city; 2) *OD generation*. This module first generates the OD candidates in the target city by using distance constraints to filter the unlikely OD pairs. Then, mobility intention features are generated and the OD pairs with the most similar latent features in the target city are returned; and 3) *path generation*. As people usually choose paths that are similar to the shortest path with less number of turns (demonstrated in Figure 3b and c), and the preferences across cities are similar. A utility model based method is proposed to learn the path selection preferences and predict their choice probabilities.

In this paper, we use large scale trajectory data from Mobike¹, as its dock-less deployment of Mobikes effectively reflects of the short-term mobility intention of people, and the trajectory generation in new cities helps the company with its expansion strategy and contributes to many urban applications, such as bike path planning [2] and chain-shop location selection [22]. The main contributions of the paper are summarized as follows:

- To our best knowledge, we provide the first attempt to generate spatial trajectories in new cities, without any mobility data in the new city. We focus on short-range mobility and generate trajectory data of sharing bikes, which is valuable for many applications.
- We propose a novel mobility intention model to transfer mobility knowledge. We also propose an origin-destination generation model to a new city.
- We demonstrate that the path preferences are similar among different cities. Based on this insight, we build a utility model to generate the path based on the people's path selection preferences.
- We validate the effectiveness of both OD pair generation and path generation extensively using the massive trajectory data from four regions in China. Moreover, a real-world case study is conducted, which provides insights for urban planning and business.

¹<https://en.wikipedia.org/wiki/Mobike>

2 OVERVIEW

2.1 Preliminaries

DEFINITION 1. (Map-Matched Trajectory) A map-matched trajectory τ is defined as a road segment sequence $\tau = \{r_1 \rightarrow r_2 \rightarrow \dots \rightarrow r_n\}$, where $r_i \in R, 1 \leq i \leq n$.

In this paper, we focus on generating/transferring map-matched trajectories, and IVMM algorithm [42] is used to perform the map matching task over the raw GPS trajectories to the road network.

DEFINITION 2. (OD Pair) An Origin-Destination pair OD_i is a road segment pair $(r_{o,i}, r_{d,i})$, which are the first and last road segment of map-matched trajectory τ_i , respectively.

DEFINITION 3. (Spatial Context Feature) Spatial context feature x_i is a vector associated with an OD pair OD_i . The features are extracted from the multi-source data, including POI, transportation stations and road networks.

DEFINITION 4. (Mobility Intention Feature) Denoted as f_i , it is the hidden representation of spatial context feature x_i , which represent the mobility intentions in a latent space.

DEFINITION 5. (Domain) A domain [30] consists of two components: \mathcal{X} and $P(X)$. \mathcal{X} is the feature space. $P(X)$ is the marginal probability distribution, and $X = \{x_1, \dots, x_n\}, x_i \in \mathcal{X}, 1 \leq i \leq n$.

In this paper, a city is associated with a domain. \mathcal{X} is the spatial context feature space, and $P(X)$ is the spatial context feature distribution of the trajectories in the city X .

2.2 Problem Definition

With the multi-source data from the source and target cities, given only the trajectory data $\mathcal{S}(\tau)$ from the source cities, we want to generate a set of map-matched trajectories in the target city, which have the similar distribution to the ground truth trajectories $\mathcal{T}(\tau)$ in the target city.

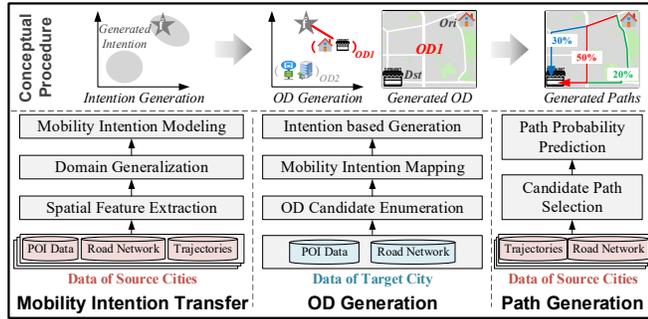


Figure 4: An Overview of the System.

2.3 System Overview

Figure 4, with a conceptual procedure illustrated at the top part, gives an overview of our system:

Stage I: Mobility Intention Transfer. This component generates a unified mobility intention shared across cities. It performs three tasks: 1) *Spatial Context Feature Extraction*, which extracts spatial context features for trajectory OD pairs using the multi-source data; 2) *Domain Generalization*, which projects the spatial context features to the mobility intention space, where the distributions of

different source cities are similar; 3) *Mobility Intention Modeling*, which models the unified mobility intention feature distribution for generating mobility intention in the target city (detailed in Section 3).

Stage II: OD Generation. This component takes the generated mobility motion in adapted space and outputs the OD pair in the target city. This component consists of three tasks: 1) *OD Candidate Enumeration*, which extracts the possible OD pairs in the target city; 2) *Mobility Intention Mapping*, which maps and indexes the enumerated OD candidates to the unified mobility intention space. 3) *Intention based Generation*, which generates OD pairs in the target city, based on the mobility intention model (detailed in Section 4).

Stage III: Path Generation. This component takes a generated OD pair and generates road-granule paths connecting the OD pair with probabilities, which is based on the model learned from the real trajectory choices from the source cities. It consists of two tasks: 1) *Candidate Path Selection*, which employs an algorithm to effectively select candidate paths; 2) *Path Probability Prediction*, which learns a model to predict the choice probability of each candidate path (detailed in Section 5).

3 STAGE I: MOBILITY INTENTION TRANSFER

3.1 Overview

To transfer the mobility intentions to a target city, we need to find an effective adaptation function to summarize the commonalities of mobility intentions across different cities. Based on spatial context features around users' OD locations, we employ the domain generalization to project these features from different cities to a latent space, where the distributions of different cities become similar. In this way, a generative model is built to summarize the mobility intention distribution, which makes it possible to generate mobility intentions in a target city.

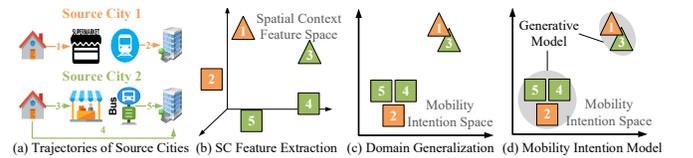


Figure 5: An Illustration of Mobility Intention Transfer.

As a result, three steps are performed here, as demonstrated in Figure 5: 1) spatial context feature extraction, 2) domain generalization, and 3) mobility intention modeling. The OD information in the source cities is illustrated in Figure 5a, where the two cities (marked in green and orange) have five OD pairs extracted from the real trajectories. The first step extracts the spatial context features of the OD pairs, as is shown in Figure 5b. Although there are only two different types of mobility intentions, i.e., shopping (as triangles) and going to work (as squares), the distributions of spatial context features are different between the two cities, as the layouts of the spatial context features are not the same between cities. To address the above domain shift between cities, the domain generalization step learns the adaptation function, which projects the spatial context features to a mobility intention space, where the feature distributions of the two cities are similar (e.g., in Figure 5c).

Then, the unified mobility intention distribution is modeled (as the shade areas in Figure 5d). Based on this model, in the OD generation stage, our system is able to generate ODs with similar mobility intention distribution in a target city, which is detailed in Section 4.

3.2 Spatial Context Feature Extraction

The intention of human mobility is highly related to the spatial context features near the OD locations. Thus, for each origin or destination location, we extract three categories of spatial features from multi-source data in a spatial range θ_δ (empirically set as 300 meters), including POI features, public transportation features, and road network features.

POI feature. It includes the number of POIs in each category within the spatial range for both OD locations.

Public transportation features. It extracts two types of features: 1) the number of transport stations in different categories (e.g., bus stops or subway stations), and 2) the distances to the nearest different transport stations.

Road network features. This category of features involves the following information: 1) the length of the shortest path between OD; 2) the number of adjacent roads connected to the location; 3) road level at the location; and 4) the graph eccentricity of the location in the road network.

Finally, the spatial context features of both the origin and destination are composed together to derive the spatial context features of an OD pair.

3.3 Domain Generalization

In this step, we train an adaptation function $G_D(\cdot)$ that projects the spatial context features to mobility intention space, where the distribution difference across multiple source cities is minimized. In this way, the trained adaptation function can be applied directly to the unseen target domain. We also validate the effectiveness of domain generalization with real-world data from Mobike.

There has been a large body of algorithms proposed to minimize the distribution difference of two domains e.g., [9–11]. In this work, we employ Transfer Component Analysis (TCA) [29], which eases the implementation and training yet provides good scalability. TCA essentially learns a set of transfer components in Reproducing Kernel Hilbert Space (RKHS) by minimizing the Maximum Mean Discrepancy (MMD) [4] between the datasets from different domains. The algorithm is performed mainly via eigen value decomposition and thus can be parallelized. The learned transfer components C span a subspace, where the distributions of different domains are close to each other. Note that, TCA is applicable for more than two source cities by extending MMD for multiple domains, e.g. [11]. However, in the following examples, we only use two source cities for brevity.

With the transfer components C learned from two source city data X_{S_1}, X_{S_2} with the kernel function K , the adaptation function for spatial intention feature x is straightforward:

$$G_D(x) = K(x, X_S^T) \cdot C, \quad (1)$$

where X_S^T is the transpose of the concatenated matrix from the source cities, i.e., X_{S_1}, X_{S_2} . As a result, given a spatial context feature

sample x_i , the mobility intention feature f_i is:

$$f_i = G_D(x_i). \quad (2)$$

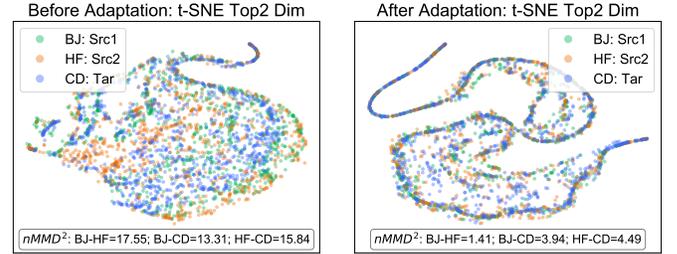


Figure 6: The Effect of Domain Generalization.

Figure 6 shows the result of domain generalization using TCA on our real data set, where Beijing and Hefei are used as source cities, and Chengdu is used as the target city. We train the adaptation function using *only the spatial context features of ODs from source cities*, i.e. Beijing and Hefei, and then we apply the trained function to all of the three cities. Besides, for both spatial context space and mobility intention space, the MMD values (detailed in Equation 8) of all different city pairs are computed. Figure 6(a) is a visualization of the original spatial context feature distributions of the three cities using t-SNE [26]. Figure 6(b) shows the distributions after the adaptation function is applied. It is clear that after applying the adaptation function, not only the domain shift between the source cities is reduced (with $n \cdot MMD^2$ decreasing from 17.55 to 1.41, where n is the total number of OD pairs), but also the mobility intention distribution of the target city becomes similar to the source cities (with $nMMD^2$ to Beijing and Hefei dropping from 13.31 and 15.84 to 3.94 and 4.49 respectively). As a result, the effectiveness of our idea to adopt domain generalization is validated.

Analysis. The ultimate goal is to acquire the mobility intention features of the target city, Chengdu. Since the domain shift is reduced obviously after leveraging the adaption function, it is possible to “simulate” the target city Chengdu in mobility intention space (blue ones in Figure 6b) using the known features of Beijing and Hefei (green and orange ones).

Note that the adaptation function does not guarantee the clustering of features with similar explicit mobility semantics (e.g. shopping or commuting), but the latent semantics instead.

3.4 Mobility Intention Modeling

Since the mobility intention features from the source cities can be generalize to the target city (as is analyzed in Section 3.3), in this step, we build a model that can generate these features and apply it to the target city.

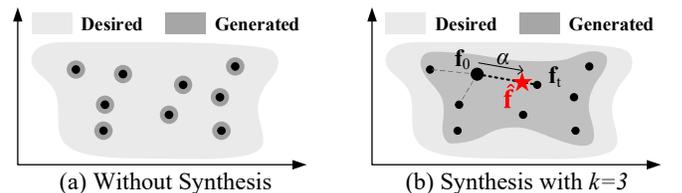


Figure 7: Data Synthesis for Lack of Data Problem.

A naïve generation strategy is to directly sample from the mobility intention features of the source cities. However, given a small dataset, this strategy results in the lack of generalization ability. Figure 7(a) gives a demonstration, where the feature data (depicted in black dots) is under-representative for the mobility intention distribution of the city. To overcome this problem, we propose a data synthesis based generation strategy by k NN query and interpolation, which is similar to SMOTE [5].

Algorithm 1 Mobility Intention Generation Algorithm

Input: $\mathcal{S}(\tau)$, trajectory data from source cities, and the kernel function $K(\cdot, \cdot)$.
Output: Generated mobility intention feature $\hat{\mathbf{f}}$.

// **Spatial Context Feature Extraction**
 1: $\mathcal{S}(\mathbf{x}) \leftarrow$ Spatial Context Features of OD pairs in $\mathcal{S}(\tau)$
 // **Domain Generalization**
 2: Get $G_D(\cdot)$ by Equation 1, with $C \leftarrow TCA(K, \mathcal{S}(\mathbf{x}))$
 3: $\mathcal{S}(\mathbf{f}) \leftarrow \{G_D(x_i) | x_i \in \mathcal{S}(\mathbf{x})\}$
 // **Intention Generation**
 4: $\mathbf{f}_0 \leftarrow \text{RandomChoice}(\mathcal{S}(\mathbf{f}))$
 5: $\{\mathbf{f}_1^n, \dots, \mathbf{f}_k^n\} \leftarrow \mathcal{S}(\mathbf{f}).k\text{NN}(\mathbf{f}_0)$
 6: $\mathbf{f}_t \leftarrow \text{RandomChoice}(\{\mathbf{f}_1^n, \dots, \mathbf{f}_k^n\})$
 7: Generate a random number $\alpha \in [0, 1]$
 8: $\hat{\mathbf{f}} \leftarrow (1 - \alpha) \cdot \mathbf{f}_0 + \alpha \cdot \mathbf{f}_t$
 9: return $\hat{\mathbf{f}}$

Algorithm. Algorithm 1 provides the pseudo-code. With the mobility intention features of the source cities $\mathcal{S}(\mathbf{f})$ calculated (Line 1-3), it first randomly selects an intention feature from the source cities as \mathbf{f}_0 (Line 4). Then a k NN query is employed, and a neighbor point \mathbf{f}_t is selected randomly from the result (Line 5-6). After that, it synthesizes a feature point $\hat{\mathbf{f}}$ by random interpolation between \mathbf{f}_0 and \mathbf{f}_t (Line 7-8), and $\hat{\mathbf{f}}$ is the generated intention.

Figure 7b demonstrates an example with $k = 3$, where the red star is the synthesized data $\hat{\mathbf{f}}$ between \mathbf{f}_0 and its neighbor \mathbf{f}_t with offset ratio α . As a result, we can improve the generalization ability to the desired distribution using a relatively small set of data points by data synthesis. The effectiveness is detailed in Figure 11c of Section 6.3.

4 STAGE II: OD PAIR GENERATION

In this section, we generate OD pairs in the target city based on the mobility intention model transferred from source cities.

Intuition. The idea is simple: given a generated mobility intention feature, we search for the OD pair in the target city that has the most similar mobility intention to it.

As a result, generation of OD pairs can be decomposed into three steps, as demonstrated in Figure 8: 1) *OD candidate enumeration*, which enumerates all possible OD pairs in the target city, i.e., the arrows denoted as {1, 2} in Figure 8a. 2) *Mobility intention mapping*, which maps the enumerated OD candidates to the mobility intention space (as in Figure 8b). And 3) *Intention based generation*, which finds the most similar OD candidate in mobility intention space, which is shown in Figure 8c, where the mobility intention model is transferred from the source cities (i.e., shade areas). Given the generated mobility intention $\hat{\mathbf{f}}$, we find {1} the most similar OD pair as the generation result.

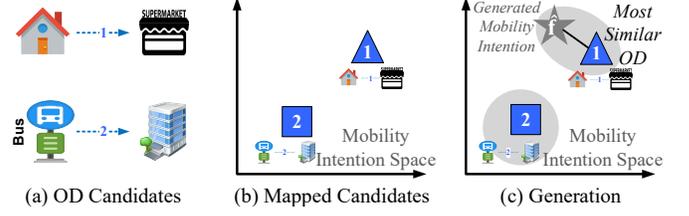


Figure 8: An Illustration of OD Generation.

4.1 OD Candidate Enumeration

We enumerate all candidate OD pairs in the target city. Instead of the brute-force enumeration, which is $O(n_r^2)$, n_r is the size of road segments, we empirically select the OD pairs with shortest path length within 6.0km, as the most of the bike trips (91.7%) are within 6.0km [2]. This empirical trick can help decrease the number of OD candidates to $n_r \cdot n_\sigma$, where n_σ is the average number of roads within 6.0km of a road, valuing around 2000 depending on the city regions.

4.2 OD Candidate Mapping

To keep the consistency of the target city OD pairs and the transferred mobility intention model, we map all enumerated OD candidates to the same mobility intention space. As a result, we apply the same spatial context feature extraction scheme described in Section 3.2, along with the adaptation function $G_D(\cdot)$ in Equation 1 trained by the source cities to convert the spatial context features to mobility intention space, denoted as $\mathcal{T}_c(\mathbf{f})$.

4.3 Intention based Generation

With the intention features of OD candidates, we draw $\hat{\mathbf{f}}$ from the transferred mobility intention model, and find the OD candidate with the most similar features. Since the mobility intention space is a low dimensional latent space, we simply use the inverse of the Euclidean distance as the similarity metric, i.e. for $\hat{\mathbf{f}}$ and a candidate \mathbf{f}_c , the similarity is $\text{Sim}(\hat{\mathbf{f}}, \mathbf{f}_c) = 1/\|\hat{\mathbf{f}} - \mathbf{f}_c\|_2$. Then, the searching procedure is equivalent to finding the nearest neighbor, i.e. for the given intention $\hat{\mathbf{f}}$, the generated OD pair \widehat{OD} is

$$\widehat{OD} = OD_c$$

s.t.

$$c = \arg \min_{0 \leq i < |\mathcal{T}_c(\mathbf{f})|} \|\hat{\mathbf{f}} - \mathbf{f}_i\|_2. \quad (3)$$

To speed up the search, we build a KD-Tree index in mobility intention space for the mapped candidates.

5 STAGE III: PATH GENERATION

5.1 Overview

In this section, we describe the procedure that generates the paths between the OD pairs in the target city. Generating paths based on given OD pairs is a non-trivial task for the following two reasons: 1) there are many possible paths between the OD in a road network, but only a limited number of paths are traveled frequently; and

2) the distributions between the candidate paths are very different, which is affected by the features of the paths.

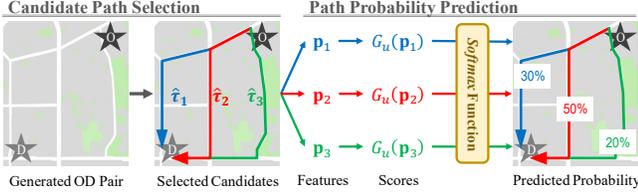


Figure 9: An Illustration of Path Generation.

To this end, we propose a two-step path generation model, demonstrated in Figure 9: 1) *Candidate Path Selection*, which calculates m candidate paths based on a given OD pair; and 2) *Path Probability Prediction*, which predicts the choice probability distribution among all candidate paths. Finally, the path is generated from the candidates with respect to their choice probabilities.

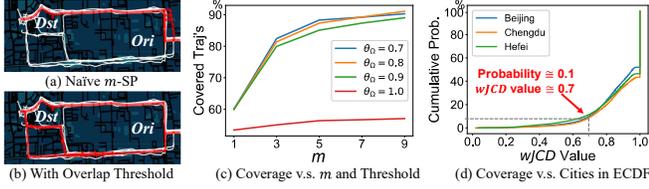


Figure 10: Intuition and Results Our Path Selection.

5.2 Candidate Path Selection

Main idea. In this step, m candidate paths are selected between the given OD pairs. A straightforward way is to compute the top m shortest paths, e.g., using Yen’s algorithm [41]. As demonstrated in Figure 3b, over 70% of the bike trip lengths are very close to the length of the shortest path. However, in the road network, the top- m shortest paths are very similar to each other, as shown in Figure 10a, where the top 5 shortest paths (i.e., in red) only have some minor differences at the beginning and can only cover a very limited portion of all real trajectories (i.e., illustrated as white lines).

To this end, we introduce an overlap constraint to the path generation algorithm to filter the overlapped candidates. In our implementation, weighted Jaccard ($wJCD$) value is employed, as it is a common metric to evaluate the degree of overlapping between two sequences [6, 12]. The formula is as follows:

$$wJCD(\tau_1, \tau_2) = \frac{(\tau_1 \cap \tau_2).len}{(\tau_1 \cup \tau_2).len} = \frac{(\tau_1 \cap \tau_2).len}{\tau_1.len + \tau_2.len - (\tau_1 \cap \tau_2).len}, \quad (4)$$

where $(\tau_1 \cap \tau_2).len$ is the total length of the overlapped road segments between the two paths. For example, if two paths are with length 10km and the length of their overlapping road segments is 8km, the $wJCD$ value is $8km / (10 + 10 - 8)km = 0.667$. In this paper, paths with $wJCD$ larger than 0.7 are considered as well-overlapped.

Algorithm. Algorithm 2 shows the pseudo-code of the path selection. First, we employ Yen’s algorithm [41], which iteratively

Algorithm 2 Overlap Threshold based Candidate Path Selection

Input: Origin and destination roads r_O and r_D , road network G_{rn} , overlap threshold θ_Ω , and the number of candidate paths m .

Output: The list of candidate paths T_{cand} .

```

1:  $T_{cand} \leftarrow \emptyset$ 
   // Yen’s shortest paths enumerator.
2:  $yen \leftarrow Yen(G_{rn}, r_O, r_D)$ 
3: while  $|T_{cand}| < m$  and  $yen.hasNext()$  do
4:    $\hat{\tau} \leftarrow yen.next()$ 
5:    $maxJCD \leftarrow \max_{\tau_i \in T_{cand}} wJCD(\hat{\tau}, \tau_i)$ 
6:   if  $maxJCD < \theta_\Omega$  then
7:     Append  $\hat{\tau}$  to  $T_{cand}$ 
8: return  $T_{cand}$ 
    
```

generates paths between OD ordered by the path lengths (Line 2). For each newly generated path $\hat{\tau}$, we compute its $wJCD$ to each candidate path in T_{cand} (Line 4-5). If all $wJCD$ values are less than θ_Ω , the newly generated path $\hat{\tau}$ is inserted in the candidate path set T_{cand} (Line 6-7). The algorithm terminates when: 1) up to m paths are selected; or 2) there are no more loopless paths between OD .

Example. Figure 10b gives an example of the path generation results using the overlap threshold based candidate path selection algorithm, where the red lines are the generated top 3 candidate paths. The generated paths cover much more real bike trips comparing to the naive top- m Shortest Path approach in Figure 10a.

Analysis. Figure 10c shows the path coverage performance (i.e., recall ratio of the real top m traveled paths) comparison between Overlap Threshold based Candidate Path Selection (with different θ_Ω settings) and the naive shortest m path algorithm, using the real bike trips in Beijing. From the figure, we can notice that our algorithm outperforms the naive top- m SP approach.

We also validate the effectiveness of this algorithm in different cities, e.g., Chengdu and Hefei. Under the same parameter settings that $m = 5$ and $\theta_\Omega = 0.7$, as demonstrated in Figure 10d, we get similar performance in Beijing, Chengdu and Hefei, i.e., over 90 percents of trajectories can be matched to one of the m paths.

Moreover, it is obvious that a larger m leads to better coverage. However, the computational cost of a larger m increases. The trade-off between the coverage and efficiency will be demonstrated in experiments.

5.3 Path Probability Prediction

Estimating the choice probability distribution of the candidate paths is another important task in path generation. As we demonstrated in the introduction, Figure 3b and c, the users’ preference in choosing the path between an OD pair depends on the features of the paths, e.g., total length, number of turns, and directions.

To this end, we employ a utility model [31] to predict the choice probability distribution of the candidate paths. The utility model gives each path a utility score and uses *Softmax* function to calculate the probabilities. The path probability prediction component of Figure 9 illustrates the procedure, where there are three candidate paths selected, i.e. $\{\hat{\tau}_1, \hat{\tau}_2, \hat{\tau}_3\}$. We first extract features for each path represented as three vectors $\{p_1, p_2, p_3\}$, and then through a utility

model, the probabilities for each candidate path is predicted. In this section, we detail the two steps.

Path Feature Extraction. We extract topological features to represent each candidate path $\hat{\tau}$ involving: the number of road segments, the length of the path, the number of left and right turns, the number of U-turns, the traversing frequency of each road level, and the number of wrong road transits ($r_i \rightarrow r_{i+1}$ is considered wrong if r_{i+1} is not the way to achieve the shortest path from r_i).

Model Training. The main intuition of the utility model is to find a utility function G_u for paths that fitting the training data. Figure 9 details the supervised utility model training process. Each training example is the features $\{\mathbf{p}_1, \dots, \mathbf{p}_m\}$ and ground truth probabilities $\{y_1, \dots, y_m\}$ of the m candidate paths associated with an OD.

During the training phase, the model first computes the utility score by function G_u , and then applies the *Softmax* function to convert scores to probabilities. Finally, the training loss is computed between the predicted and the ground truth probabilities. We use Cross-Entropy in the loss function. Therefore, the formal loss function for a training example is formulated as

$$\begin{aligned} \text{Loss}(y, \hat{y}) &= \text{CrossEntropy}(y, \hat{y}); \\ \hat{y}_i &= \frac{\exp(G_u(\mathbf{p}_i))}{\sum_m \exp(G_u(\mathbf{p}_m))}. \end{aligned} \quad (5)$$

As based on our observations, the path selection preferences of people in different cities are similar, in this paper, we directly apply the model learned from the trajectories of the source cities, to the target city.

Implementation. We set two Fully-Connected neural network layers as utility function G_u . As a result, this training process can be easily converted to the popular Stochastic Gradient Descent optimization form. In this work, we adopt *Adam* as the optimizer, with a learning rate of 0.01.

Ground Truth. In this supervised utility model training process, for each OD training example, we first need to compute the ground truth probability of the candidate paths associated with the OD pair. However, since the trajectories are less likely to exactly match the candidate paths, the probability cannot be computed simply by counting the frequency of trips on each path. Realizing this, in this work, we decide to simulate the ground truth by matching each trajectory to the most similar candidate path, and then the frequency of each path is alternated as the number of matches, i.e. the ground truth visiting frequency D_i of candidate path $\hat{\tau}_i$ is estimated as

$$D_i = |\{\tau_j | \hat{\tau}_i = \arg \max_{\hat{\tau}_i} wJCD(\tau_j, \hat{\tau}_i)\}|. \quad (6)$$

Here τ_j 's are trajectories between the given OD. As a result, the estimated choice probability ground truth for path $\hat{\tau}_i$ is derived:

$$y_i = \frac{D_i}{\sum_j D_j}. \quad (7)$$

In addition, to guarantee the quality of the training data, only the OD pairs with more than 30 trajectories are used as training data.

6 EXPERIMENT

In this section, we first describe the experimental datasets, evaluation metrics and the baselines approaches. Then, we present the

Table 1: Details of the Datasets

	Chaoyang	Haidian	Chengdu	Hefei
Region Size	18.8km ²	16.1km ²	14.5km ²	15.3km ²
# of Trajectories	128,546	123,188	127,577	128,733
# of Roads	3,180	3,252	1,668	1,755
# of POIs	26,030	23,751	23,236	8,044
# of Stations	1,905	1,323	1,003	321

evaluation results with different settings. Finally, an on-field case study is conducted.

6.1 Data Descriptions

Mobike Trajectories. We collected a portion of bike trajectories² in three months (from 01/04/2018 to 01/07/2018) from four cities/regions in China: (i) Chaoyang district of Beijing, (ii) Haidian district of Beijing, (iii) Chengdu city and (iv) Hefei city. All these regions have a large number of Mobike usage.

Multi-source Data. The multi-source datasets include POI data, transport station data and Road network data. Details of these datasets are summarized in Table 1.

6.2 Evaluation Metrics

Due to the enormous space of the map-matched trajectory, directly evaluating the distribution difference of two trajectory set is computationally unfeasible, which is also discussed in [28]. To this end, many works use spatial distribution of GPS points to evaluate the effectiveness of generation results [23, 28]. We argue that it is not appropriate to simply compare the spatial distribution of separate trajectory points of the generated and ground truth data, since different trajectory sets can result in completely the same trajectory point distribution. Realizing this, we divide the evaluation by two the stages separately, i.e. the evaluation of *OD generation* and the evaluation of *path generation with fixed OD*:

OD generation. The ground truth and the generated OD pairs are represented as two sets \mathcal{A} and $\hat{\mathcal{A}}$, and each element contains 4 entries [$lat_O, lng_O, lat_D, lng_D$], which are the latitudes and longitudes of the OD pair. We use $n \cdot MMD^2$ to evaluate the distribution difference between \mathcal{A} and $\hat{\mathcal{A}}$, where n is the number of OD pairs in \mathcal{A} and $\hat{\mathcal{A}}$. MMD [4] essentially computes the distance between the centroids of two distributions in Reproducing Kernel Hilbert Space (RKHS). Formally, for an RKHS \mathcal{H} with kernel function ϕ , MMD is calculated as follows:

$$MMD(\mathcal{A}, \hat{\mathcal{A}}) = \left\| \frac{1}{|\mathcal{A}|} \sum_{a_i \in \mathcal{A}} \phi(a_i) - \frac{1}{|\hat{\mathcal{A}}|} \sum_{\hat{a}_i \in \hat{\mathcal{A}}} \phi(\hat{a}_i) \right\|_{\mathcal{H}}. \quad (8)$$

Path generation with Fixed OD. We use KL Divergence to evaluate the performance of path generation:

$$KLD(P_{gt}, P_{gen}) = - \sum_{r_i \in R_{\tau}} P_{gt}(r_i) \cdot \log \frac{P_{gen}(r_i)}{P_{gt}(r_i)}, \quad (9)$$

²The whole dataset is not used due to our data confidential agreement.

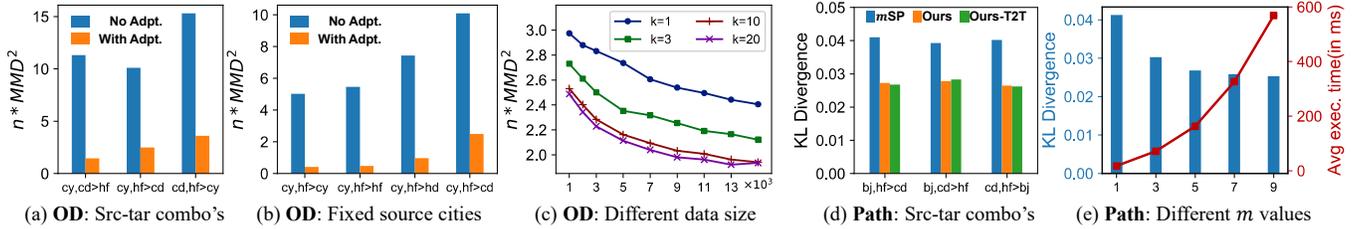


Figure 11: Experiments on OD generation and path generation.

where $P_{gt}(r_i)$ and $P_{gen}(r_i)$ are the probabilities to travel the road r_i with the ground truth and the generated trajectories, respectively. Laplace smoothing is used to avoid zero probabilities in the computation of KL Divergence. R_τ is the set of all road segments.

6.3 OD Generation

In this section, we present the effectiveness of OD generation under different settings. We set the source cities as Chaoyang and Hefei, the target city as Chengdu, the synthesis parameter $k = 5$ and the number of training data as 5000. The entire trajectory data is used for evaluation, which is divided equally into four sets, and the average MMD value is calculated to avoid memory overload. The evaluation repeats three times to overcome the generation variance.

Source-target combinations. We enumerate the combination of source and target cities to see the effectiveness of our adaption function. Figure 11a shows the performance of our solution with and without adaption function. It is clear from the figure that: 1) the effectiveness of OD generation improves when the adaption function is used, which validates our idea to mapping the features in a unified mobility intention space. 2) The OD transfer performance from Chengdu-Hefei to Chaoyang is relatively low. This is because Chaoyang district, as a part of the capital city, Beijing, is different from the two other small cities. As a result, it is relatively more challenging to transfer mobility knowledge from small cities to big cities. Our congestion here is that the big cities contain more diverse regions that can cover more spatial context features in small cities.

Different target cities. We keep the source cities Chaoyang-Hefei invariant, and evaluate the performances on different cities, i.e., Chaoyang, Hefei, Haidian, and Chengdu. Results are presented in Figure 11b, where we have the following observations: 1) self-validations to Chaoyang and Hefei show good performances, which implies the representation ability of mobility intention. 2) the transfer to Haidian is relatively better than Hefei because Haidian and Chaoyang are similar, as both of them are in Beijing.

Different data sizes. We also study the OD generation performance under different numbers of trajectories for the training phase. Moreover, we evaluate the effectiveness of the proposed data synthesis technique described in Section 3.4 by setting different k values. Figure 11c shows two main observations: 1) the $nMMD^2$ decreases with the increase of data size, which implies the necessity of learning mobility knowledge from the large-scale trajectory data; and 2) by introducing data synthesis, the number of trajectories required in generation is reduced.

6.4 Path Generation with Fixed OD

In this section, we describe the experimental results of path generation. We compare our solution with two baselines.

- **mSP.** It chooses the top- m paths between OD as the path candidates and learns the utility model from the source cities.
- **Ours-T2T.** This method uses our Algorithm 2, but the model directly applies the path probability prediction model learned from the target city and used to the same city (target city) directly.

Source-target combinations. We enumerate source-target combinations and set candidate paths number as $m = 5$. Figure 11d shows the results, where we have two important observations: 1) our solution outperforms the naive top- m shortest path solution (i.e., mSP), which implies the necessity of introducing the overlap threshold to select candidate paths; and 2) the model trained with the trajectories in the same city (i.e., Ours-T2T) does not make a note-worthy improvement, which validates our intuition that people in different cities have similar preference to choose paths. Therefore, we do not need to build different path preference models based on different source/target cities.

Different m values. We study the trade-off between effectiveness and execution time by settings different candidate path numbers m from 1 to 9. The bars in Figure 11e shows the effectiveness (i.e., KL divergence), and the line shows the average time cost for generating one trajectory in milliseconds. From the figure, we can see that: 1) with the increase of m , the path generation performance improves, as more candidate paths guarantee more comprehensive coverage. However, the effectiveness with $m = 7$ and $m = 9$ improve slightly over $m = 5$, which implies the most of users only select up to 5 different paths for an OD pair; and 2) the execution time increases when m gets larger. When $m = 9$, it takes nearly 600 milliseconds to generate one trajectory in average. As a result, generating 1,000 trajectories takes around 10 minutes, which becomes the efficiency bottleneck in our system (generating 1,000 ODs takes only 300 milliseconds). As a result, we set $m = 5$ as the default value to achieve the best trade-off between accuracy and efficiency.

6.5 A Demo of the Generated Trajectory Heatmap

In addition to the quantified evaluations described in Section 6.3 and Section 6.4, we also visualize the heatmap distributions of the trajectories between our generation results and the ground truth. Figure 13 demonstrates the results for the city of Chaoyang and Hefei. From the figure, we can observe that, for both of the cases, the generated spatial distribution of the trajectories are very similar to the ground truth distribution.

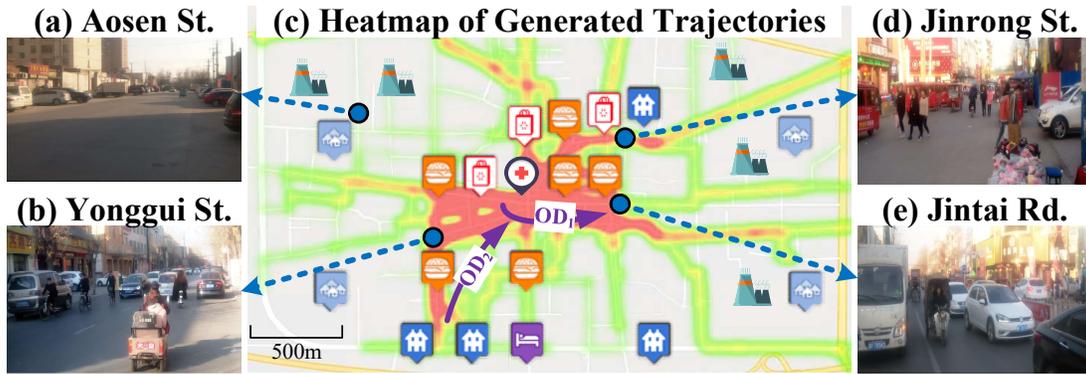


Figure 12: On-field Case Studies in XiongAn.

6.6 On-field Case Studies in XiongAn

We conduct a real-world case study in XiongAn, a promising city attracting strong attention from both the entrepreneurs and government staffs in China. We generate the trajectories using our system. We first analyze the popular OD pairs, and then visualize the generated trajectories using heatmap. Figure 12c gives the heatmap and POI distribution of this city. According to our generated ODs, we have two observations: 1) *finding 1: many inter-trips in the downtown area*. A set of popular OD pairs like OD_1 in the figure appears in the downtown area with many POIs around; 2) *finding 2: many trips between the south area and downtown*. There also exists a popular travel OD from the southwest to the downtown, i.e. OD_2 , as a new residential area is built in the southwest.

When we arrived there, we found only a few people in the rural area, e.g. Aosen St. in Figure 12a, which is consistent with our generation results. On the other side, there is more crowd flow in Jinrong St. and Jintai Rd (i.e., illustrated Figure 12d and e), which are the main roads for inter-trips connecting the downtown area (i.e., consistent with *finding 1*). Moreover, we find that many people are traveling in Yonggui St (Figure 12b). The road connects the southwest area and the downtown, which validates our *finding 2*.

From the generated heatmap, we can also get the following insights for urban applications: 1) *business site selection*. As Yonggui St. is a street with plenty of traveling people, but with a limited number of POIs. Thus, it is a good place to set up new businesses, like restaurants and stores; 2) *government planning 1*. As shown in Figure 12e, non-motor vehicles have to ride on the center of the Jintai Rd., as there are no bike lanes. Thus, our suggestion here is to build bike lanes along the road; and 3) *government planning 2*. As shown in Figure 12d, there are many vendor booths along the road, which attracts pedestrians. Our suggestion is to construct more

shop buildings nearby to not only improve the shopping experience, but also ease the congestion.

7 DEMO SYSTEM

A demo of our bike trajectory generation system is publicly available online [1], with mobility intention model trained using trajectory data of Chengdu and Hefei. Figure 14 presents the system interface. The system allows users to generate trajectories, preview the heatmap, and show the details of generated trajectories. It contains the following components:

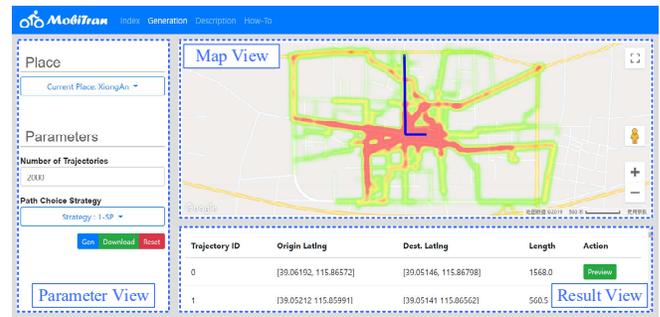


Figure 14: System Interface.

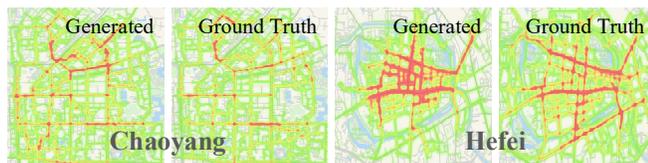


Figure 13: The Generated Heatmap in Chaoyang and Hefei.

Parameter View. The user selects an area for generation (Chaoyang District or XiongAn in this demo), the number of trajectories, and the path generation algorithm to get generated trajectories. The user can also download the generated trajectories.

Result View. In this view, a table shows the details of each trajectory: the trajectory ID, the OD locations and its total length. The user can have also a view of each trajectory.

Map View. The upper right section is the map view, which displays the trajectory heatmap and the detailed trajectories (polylines in blue).

8 RELATED WORK

In this section, we summarize the four most related areas: 1) urban computing, 2) transfer learning, 3) human mobility and 4) Trajectory Recovering.

Urban Computing. Urban computing [45] aims to address problems in the city. On one hand, many works study the mobility in the city, e.g. [40] predicts the taxi demand to enable smart scheduling. [35, 46] try to understand human mobility patterns from check-ins. [13] studies the effect of illegal parking events on sharing bike trajectory distribution passing and proposes to detect illegal vehicle parkings, and [24] proposes an online trajectory anomaly detection algorithm. On the other hand, trajectory data management systems [20, 21, 32] are built to improve the efficiency of urban trajectory mining applications. In our problem, we transfer the human mobility patterns across cities.

Transfer Learning. Many works solve urban problems by applying traditional transfer learning assuming the target domain is available. E.g. [34] tackles the data insufficiency problem. [19] predicts the Home-to-Work time for families in a new city based on survey data of families in both source and target cities. On the other hand, the traditional unseen target domain problems, e.g. [9, 10, 39], uses domain generalization technique [27] to address the problem of label data unavailability. The closest work to us is the transfer of sharing bike stacking hot spots, which uses multi-source data to predict the density of sharing bikes for each road [25]. However, it only transfers the spatial hotspots of bike stacking, while we transfer not only the OD pairs, but also the paths between them.

Human Mobility. Human mobility modeling tries to learn the mobility of people and reproduce the real movements. [17] summarizes the records of cellular networks in distributions, and [44] proposes a framework to learn mobility knowledge from multi-source mobility data. Some works focus on the mobility prediction task, e.g. [38]. Generative models are also used for trajectory generation, e.g. [8, 16, 23, 28]. All the above works only study the mobility modeling technique of a single city, instead of the mobility commonalities across the cities. Note that [7] also generates trajectories in new cities. However, it uses large scale trajectories in the new city to generate *abnormal* trajectories.

Trajectory Recovering. Trajectory recovering studies the problem of recovering the entire route between two locations in a trajectory. E.g. [18, 42] use empirical ways for recovering, and [37] proposes a data-driven and probabilistic approach that overcomes the data sparsity problem. Recently, computer vision based techniques are applied to recover trajectories [33], which recovers the maps from GPS points to aid trajectory recovery in areas without road map. These works mainly focus on recovering only a single route that matches the sparse trajectory. The modeling of route choice behavior, e.g. [31], focuses on the effects of choice set composition in route choice modeling, and [15] builds a choice model for San Francisco using GPS trajectories. However, none of the existing works here address our entire problem, and they did not provide studies on path preferences across different cities.

9 FUTURE RESEARCH OPPORTUNITIES: AN OUTLOOK

In this work, we focus on transferring the spatial (i.e. temporal-irrelevant) distribution of short-range trajectories. The results help analyze not only the hot spots of pedestrians, but also the popular routes over the target city region. However, the temporal distribution and the long-range mobility remains unsolved.

Temporally dynamic distribution. For a new city, inferring the dynamic mobility distribution, i.e., the mobility distribution of a given time window, is the key to precise location selection and urban management. However, this task is very challenging: 1) The cities vary in life styles. E.g., big cities have obvious rush-hour patterns compared to small towns. 2) data sparsity. The mobility data is separated into the time windows, making it difficult to accumulate sufficient data, especially in late-night.

Long-range Mobility. Long-range mobility over the city, usually more than 6km, plays an important role in public transportation services like road network planning and bus/subway route planning. It should be noted that, for long-range mobility, OD distribution is much more important than the path preference: the routes can be optimized by the government under the analysis of OD distribution. With both long-range and short-range mobility transfer, the mobility of the whole city can be recovered hierarchically. I.e. the inter-region long-term transits and inner-region short-term trajectories.

10 CONCLUSION

In this paper, we make the first attempt to study human mobility generation for a new city, by transferring the “mobility knowledge”. The solution framework first transfers mobility intention from the source cities to the target city, to generate the OD pairs. Then, by analyzing people’s path preference in different cities, we extract shared preference patterns, to generate the paths to connect each OD pair. Extensive experiment results in four regions in China are provided. In the evaluation of OD generation, our proposed method outperforms the baseline obviously, with $nMMD^2$ decreasing at least 75%. When in the path generation, we observe that the path preferences across cities are very similar. Finally, a real-world case study is conducted in XiongAn, which provides us with many insights for urban applications, e.g., store location selection and urban planning.

11 ACKNOWLEDGEMENT

We sincerely thank Hao Teng from JD Intelligent Cities Business Unit for conducting the case-studies in cold winter, and Chao Tian from Tencent and Xian-En Li from Bytedance for the discussions in the early stage of this work. Hui He was supported by the National Key R&D Program of China under grant No. 2017YFB0803304. The National Natural Science Foundation of China (NSFC) under grant No. 61672186, 61872110. Yanhua Li was supported in part by NSF grants CNS-1657350 and CMMI-1831140, and a research grant from DiDi Chuxing Inc.

REFERENCES

- [1] 2018. Mobility Transfer System. <http://t.cn/EKKzTre>.
- [2] Jie Bao, Tianfu He, Sijie Ruan, Yanhua Li, and Yu Zheng. 2017. Planning bike lanes based on sharing-bikes' trajectories. In *KDD*. ACM, 1377–1386.
- [3] Louise Barkhuus and Anind K Dey. 2003. Location-Based Services for Mobile Telephony: a Study of Users' Privacy Concerns.. In *Interact*, Vol. 3. Citeseer, 702–712.
- [4] Karsten M Borgwardt, Arthur Gretton, Malte J Rasch, Hans-Peter Kriegel, Bernhard Schölkopf, and Alex J Smola. 2006. Integrating structured biological data by kernel maximum mean discrepancy. *Bioinformatics* 22, 14 (2006), e49–e57.
- [5] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. 2002. SMOTE: synthetic minority over-sampling technique. *JAIR* 16 (2002).
- [6] Erhan Erkut and Vedat Verter. 1998. Modeling of transport risk for hazardous materials. *Operations research* 46, 5 (1998), 625–642.
- [7] Zipei Fan, Xuan Song, Ryosuke Shibasaki, Tao Li, and Hodaka Kaneda. 2016. CityCoupling: Bridging Intercity Human Mobility. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing* (Heidelberg, Germany) (*UbiComp '16*). ACM, New York, NY, USA, 718–728. <https://doi.org/10.1145/2971648.2971737>
- [8] Jie Feng, Yong Li, Chao Zhang, Funing Sun, Fanchao Meng, Ang Guo, and Depeng Jin. 2018. Deepmove: Predicting human mobility with attentional recurrent networks. In *Proceedings of the 2018 World Wide Web Conference*. International World Wide Web Conferences Steering Committee, 1459–1468.
- [9] Chuang Gan, Tianbao Yang, and Boqing Gong. 2016. Learning attributes equals multi-source domain generalization. In *CVPR*. 87–97.
- [10] Muhammad Ghifary, W Bastiaan Kleijn, Mengjie Zhang, and David Balduzzi. 2015. Domain generalization for object recognition with multi-task autoencoders. In *ICCV*. 2551–2559.
- [11] Thomas Grubinger, Adriana Birlutiu, Holger Schöner, Thomas Natschläger, and Tom Heskes. 2017. Multi-domain transfer component analysis for domain generalization. *Neural processing letters* 46, 3 (2017), 845–855.
- [12] Chenjuan Guo, Bin Yang, Jilin Hu, and Christian Jensen. 2018. Learning to route with sparse trajectory sets. In *ICDE*. IEEE, 1073–1084.
- [13] Tianfu He, Jie Bao, Ruiyuan Li, Sijie Ruan, Yanhua Li, Chao Tian, and Yu Zheng. 2018. Detecting Vehicle Illegal Parking Events using Sharing Bikes' Trajectories.. In *KDD*. 340–349.
- [14] Tianfu He, Jie Bao, Sijie Ruan, Ruiyuan Li, Yanhua Li, Hui He, and Yu Zheng. 2019. Interactive Bike Lane Planning using Sharing Bikes' Trajectories. *IEEE Transactions on Knowledge and Data Engineering* (2019).
- [15] Jeffrey Hood, Elizabeth Sall, and Billy Charlton. 2011. A GPS-based bicycle route choice model for San Francisco, California. *Transportation letters* 3, 1 (2011).
- [16] D. Huang, X. Song, Z. Fan, R. Jiang, R. Shibasaki, Y. Zhang, H. Wang, and Y. Kato. 2019. A Variational Autoencoder Based Generative Model of Urban Human Mobility. In *2019 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*. 425–430. <https://doi.org/10.1109/MIPR.2019.00086>
- [17] Sibren Isaacman, Richard Becker, Ramón Cáceres, Margaret Martonosi, James Rowland, Alexander Varshavsky, and Walter Willinger. 2012. Human mobility modeling at metropolitan scales. In *Proceedings of the 10th international conference on Mobile systems, applications, and services*. ACM, 239–252.
- [18] George Rosario Jagadeesh and Thambipillai Srikanthan. 2014. Robust real-time route inference from sparse vehicle position data. In *ITSC*. IEEE, 296–301.
- [19] Mehdi Katranji, Etienne Thuillier, Sami Kraiem, Laurent Moalic, and Fouad Hadj Selem. 2016. Mobility data disaggregation: A transfer learning approach. In *ITSC*. IEEE, 1672–1677.
- [20] Ruiyuan Li, Huajun He, Ruben Wang, Yuchuan Huang, Junwen Liu, Sijie Ruan, Tianfu He, Jie Bao, and Yu Zheng. 2020. JUST: JD Urban Spatio-Temporal Data Engine. In *ICDE*. IEEE.
- [21] Ruiyuan Li, Huajun He, Ruben Wang, Sijie Ruan, Yuan Sui, Jie Bao, and Yu Zheng. 2020. TrajMesa: A Distributed NoSQL Storage Engine for Big Trajectory Data. In *ICDE*. IEEE.
- [22] Yuhong Li, Jie Bao, Yanhua Li, Yingcai Wu, Zhiguo Gong, and Yu Zheng. 2018. Mining the Most Influential k -Location Set from Massive Trajectories. *IEEE Transactions on Big Data* 4, 4 (2018), 556–570.
- [23] Ziheng Lin, Mogeng Yin, Sidney Feygin, Madeleine Sheehan, Jean-Francois Paiement, and Alexei Pozdnoukhov. 2017. Deep generative models of urban mobility. *IEEE Transactions on Intelligent Transportation Systems* (2017).
- [24] Yiding Liu, Kaiqi Zhao, Gao Cong, and Zhifeng Bao. 2020. Online Anomalous Trajectory Detection with Deep Generative Sequence Modeling. In *ICDE*. IEEE.
- [25] Zhaoyang Liu, Yanyan Shen, and Yanmin Zhu. 2018. Where Will Dockless Shared Bikes be Stacked?:—Parking Hotspots Detection in a New City. In *KDD*. ACM.
- [26] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *JMLR* 9, Nov (2008), 2579–2605.
- [27] Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. 2013. Domain generalization via invariant feature representation. In *ICML*. 10–18.
- [28] Kun Ouyang, Reza Shokri, David S Rosenblum, and Wenzhuo Yang. 2018. A Non-Parametric Generative Model for Human Trajectories.. In *IJCAI*. 3812–3817.
- [29] Sinno Jialin Pan, Ivor W Tsang, James T Kwok, and Qiang Yang. 2011. Domain adaptation via transfer component analysis. *IEEE Transactions on Neural Networks* 22, 2 (2011), 199–210.
- [30] Sinno Jialin Pan, Qiang Yang, et al. 2010. A survey on transfer learning. *TKDE* 22, 10 (2010), 1345–1359.
- [31] Carlo Giacomo Prato and Shlomo Bekhor. 2007. Modeling route choice behavior: how relevant is the choice set composition? *TRB* 2003, 2003 (2007), 64–73.
- [32] Sijie Ruan, Ruiyuan Li, Jie Bao, Tianfu He, and Yu Zheng. 2018. CloudTP: A Cloud-based Flexible Trajectory Preprocessing Framework. In *2018 IEEE 34th International Conference on Data Engineering (ICDE)*. IEEE, 1601–1604.
- [33] Sijie Ruan, Cheng Long, Jie Bao, Chunyang Li, Zisheng Yu, Ruiyuan Li, Yuxuan Liang, Tianfu He, and Yu Zheng. 2020. Learning to generate maps from trajectories. AAAI.
- [34] Ying Wei, Yu Zheng, and Qiang Yang. 2016. Transfer knowledge between cities. In *KDD*. ACM, 1905–1914.
- [35] Fei Wu and Zhenhui Li. 2016. Where did you go: Personalized annotation of mobility records. In *CIKM*. ACM, 589–598.
- [36] Hao Wu, Ziyang Chen, Weiwei Sun, Baihua Zheng, and Wei Wang. 2017. Modeling trajectories with recurrent neural networks. *IJCAI*.
- [37] Hao Wu, Jiangyun Mao, Weiwei Sun, Baihua Zheng, Hanyuan Zhang, Ziyang Chen, and Wei Wang. 2016. Probabilistic robust route recovery with spatio-temporal dynamics. In *KDD*. ACM, 1915–1924.
- [38] Zidong Yang, Ji Hu, Yuanchao Shu, Peng Cheng, Jiming Chen, and Thomas Moscibroda. 2016. Mobility modeling and prediction in bike-sharing systems. In *Proceedings of the 14th annual international conference on mobile systems, applications, and services*. ACM, 165–178.
- [39] Huaxiu Yao, Yiding Liu, Ying Wei, Xianfeng Tang, and Zhenhui Li. 2019. Learning from Multiple Cities: A Meta-Learning Approach for Spatial-Temporal Prediction. In *The World Wide Web Conference*. ACM, 2181–2191.
- [40] Huaxiu Yao, Fei Wu, Jintao Ke, Xianfeng Tang, Yitian Jia, Siyu Lu, Pinghua Gong, Jieping Ye, and Zhenhui Li. 2018. Deep Multi-View Spatial-Temporal Network for Taxi Demand Prediction. In *AAAI*.
- [41] Jin Y Yen. 1971. Finding the k shortest loopless paths in a network. *management Science* 17, 11 (1971), 712–716.
- [42] Jing Yuan, Yu Zheng, Chengyang Zhang, Xing Xie, and Guang-Zhong Sun. 2010. An interactive-voting based map matching algorithm. In *MDM*. IEEE Computer Society, 43–52.
- [43] Nicholas Jing Yuan, Yu Zheng, Xing Xie, Yingzi Wang, Kai Zheng, and Hui Xiong. 2015. Discovering urban functional zones using latent activity trajectories. *TKDE* 27, 3 (2015), 712–725.
- [44] Desheng Zhang, Jun Huang, Ye Li, Fan Zhang, Chengzhong Xu, and Tian He. 2014. Exploring human mobility with multi-source data at extremely large metropolitan scales. In *Proceedings of the 20th annual international conference on Mobile computing and networking*. ACM, 201–212.
- [45] Yu Zheng, Licia Capra, Ouri Wolfson, and Hai Yang. 2014. Urban computing: concepts, methodologies, and applications. *TIST* 5, 3 (2014), 38.
- [46] Yuan Zhong, Nicholas Jing Yuan, Wen Zhong, Fuzheng Zhang, and Xing Xie. 2015. You are where you go: Inferring demographic attributes from location check-ins. In *WSDM*. ACM, 295–304.