GTG: Generalizable Trajectory Generation Model for Urban Mobility

Jingyuan Wang [1, 2, 3, *], **Yujing Lin**[1], **Yudong Li**[1]

[1]School of Computer Science and Engineering, Beihang University, Beijing, China
[2]MIIT Key Laboratory of Data Intelligence and Management, Beihang University, Beijing, China
[3]School of Economics and Management, Beihang University, Beijing, China

Abstract

Trajectory data mining is crucial for smart city management. However, collecting large-scale trajectory datasets is challenging due to factors such as commercial conflicts and privacy regulations. Therefore, we urgently need trajectory generation techniques to address this issue. Existing trajectory generation methods rely on the global road network structure of cities. When the road network structure changes, these methods are often not transferable to other cities. In fact, there exist invariant mobility patterns between different cities: 1) People prefer paths with the minimal travel cost; 2) The travel cost of roads has an invariant relationship with the topological features of the road network. Based on the above insight, this paper proposes a **G**eneralizable **T**rajectory **G**eneration model (GTG). The model consists of three parts: 1) Extracting city-invariant road representation based on *Space Syntax* method; 2) Cross-city travel cost prediction through disentangled adversarial training; 3) Travel preference learning by shortest path search and preference update. By learning invariant movement patterns, the model is capable of generating trajectories in new cities. Experiments on three datasets demonstrates that our model significantly outperforms existing models in terms of generalization ability.

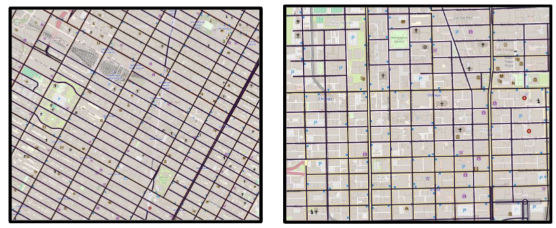
Code — https://github.com/lyd1881310/GTG

1 Introduction

Trajectory data is essential for intelligent transportation (ITS) (Wu et al. 2019; Liu et al. 2024), smart cities (Hettige et al. 2024; Ji et al. 2022), and location-based services (LBS) (Wang, Wang, and Wu 2018; Wang et al. 2021b). Trajectory generation is key to addressing the issue of insufficient trajectory data, especially in scenarios involving privacy protection and conflicts of commercial interests.

Urban trajectory generation is an active research field, and different types of methods have been proposed. These methods can be divided into two categories: knowledge-driven methods and data-driven methods (Kong et al. 2023).

Knowledge-driven methods generate trajectories based on empirical and statistical patterns of human mobility, including the Gravity Model (Zipf 1946), Intervening Opportunities Model (Stouffer 1940), and EPR model (Pappalardo et al. 2015; Pappalardo, Rinzivillo, and Simini 2016). These methods typically analyze human behavior using coarse-grained grids and derive physical priors with practical significance. By empirically summarizing human mobility behavior, these methods have achieved notable results in macroscopic traffic trajectory simulations. However, to capture human mobility behavior with finer granularity, data-driven approaches are increasingly emphasized.



(a) New York, USA (b) Chicago, USA

Figure 1: Similar local topological structures in New York (left) and Chicago (right).

Data-driven methods utilize deep neural networks to capture complex mobility patterns in trajectory data. Based on different architectures, these algorithms can be categorized into Seq2Seq-based (Park et al. 2018), GAN-based (Yu et al. 2017), VAE-based (Long et al. 2023; Huang et al. 2019), and Diffusion-based (Zhu et al. 2023) methods. With the advancements in data collection and management (Chen et al. 2017; Ding et al. 2018), data-driven methods have been widely applied. Although they have achieved significant success by learning from large datasets, they also lead to data dependency.

Knowledge-driven methods require less trajectory data but struggle to capture complex mobility patterns. In contrast, data-driven methods, while offering better performance, face challenges in generalizing to new cities. In data-driven models, changes in a city's topological structure can prevent the effective transfer of previously learned road representations to the new network.

The key to achieving cross-city trajectory generation is capturing the invariant human mobility patterns across different urban environments. Based on existing research (Wang et al. 2019b; Wang, Wu, and Zhao 2021), we have the

[*]Corresponding author

following insights about the invariant human mobility patterns. (i) People in different cities have similar travel preferences. Generally, people prefer paths with the minimal travel costs. By learning the combination of travel costs, we can generate trajectories that align with human preferences. (ii) The topological structure of the road network influences the functionality, usage frequency, and congestion levels of roads, thus determining the travel costs. Although the global road network structures differ between cities, there exists similar local topological structures. (see Figure 1), making transfer learning possible.

Based on the above insights, we propose a cross-city trajectory generation model that combines shortest path search and deep learning. This model use *Space Syntax* methods to extract topological features, employ a disentangled adversarial domain adaptation algorithm to learn invariant topological representations and predict travel costs, and finally learn human travel preferences to generate trajectories.

Our main contributions can be summarized in the following three points.

- We combine *Space Syntax* method with the advantages of deep neural networks to capture the invariant human mobility patterns across cities.
- We propose a novel cost prediction and preference learning method based on invariant topological features, and integrate it with the shortest path search algorithm to achieve cross-city trajectory generation. Our approach addresses the issue of cross-city generalization ability in trajectory generation.
- Experiments in multiple scenarios, datasets, and evaluation metrics show that our method has a generalization capability that far exceeds that of the baseline model.

2 Preliminaries

2.1 Definitions

Definition 1 (Road Network) *Road network is represented as a graph* $\mathcal{G} = \langle \mathcal{R}, \mathcal{E} \rangle$, *where* $\mathcal{R} = \{r_i \mid i = 1, 2, \cdots, N\}$ *is the road segment set and* $\mathcal{E} = \{e_{ij}\}$ *is the edge set.* $e_{ij} = 1$ *if* r_i *is adjacent to* r_j, *otherwise* $e_{ij} = 0$.

Definition 2 (Trajectory) *A trajectory is a series of consecutive road segments* $\tau = (r_1, r_2, \cdots, r_l)$, *where* l *denotes the length of the trajectory. A dataset* $\mathcal{T} = \{\tau_i \mid \tau_1, \tau_2, \cdots, \tau_C\}$ *is formed by* C *trajectories.*

2.2 Problem Statement

Given a source city with a road network $\mathcal{G}^{(src)}$ and a trajectory dataset $\mathcal{T}^{(src)}$, the objective is to train a model $\mathcal{F}$ with parameters θ to generate a new trajectory dataset $\hat{\mathcal{T}}^{(tgt)}$ for a target city with a road network $\mathcal{G}^{(tgt)}$ but no trajectory data. The model is trained using the source city data as follows

$$\theta^{(src)} = \arg\min_{\theta} \mathcal{L}\left(\mathcal{F}\left(\mathcal{G}^{(src)}; \theta\right), \mathcal{T}^{(src)}\right), \quad (1)$$

where $\mathcal{L}$ represents the optimization objective. The well-trained model is then applied to the target city to generate a new trajectory dataset

$$\hat{\mathcal{T}}^{(tgt)} = \mathcal{F}\left(\mathcal{G}^{(tgt)}; \theta^{(src)}\right). \quad (2)$$

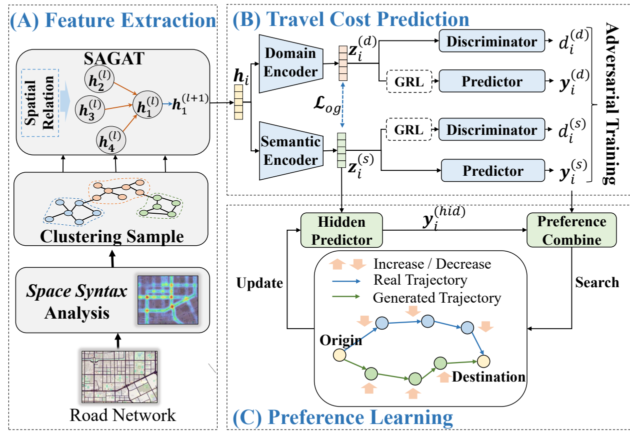


Figure 2: Overview of the framework

The primary objective is to ensure that the generated dataset $\hat{\mathcal{T}}^{(tgt)}$ is similar to the real dataset $\mathcal{T}^{(tgt)}$.

3 Methodology

3.1 Framework

As is shown in Figure 2, our trajectory generation framework consists of three modules: Topological Feature Extraction, Travel Cost Prediction and Preference Learning. The training is performed using trajectory data in the source city and road network of both source city and target city. Trajectories are generated by inferring the preference values of each road segment and searching the shortest path.

3.2 Topological Feature Extraction

***Space Syntax* Feature Extraction** *Space Syntax* (Hillier et al. 1976) is a theory for analyzing and understanding spatial structure of cities and buildings. We use four types of concepts in *Space Syntax* to describe the topological features of the road network.

Total Depth refers to the sum of the step depth (SD) from a given road segment to all other road segments within a certain range. SD is the minimum number of steps required to reach other target segment from a starting road segment.

$$x_i^{(td)} = \sum_{j \neq i} \mathrm{Len}(\mathrm{ShortestPath}(r_i, r_j)). \quad (3)$$

Integration measures the centrality of a road segment within the entire road network. The integration for road segment r_i can be formulated as

$$x_i^{(in)} = \frac{\mathrm{NC}(r_i)^2}{x_i^{(td)}}, \quad (4)$$

where $\mathrm{NC}(r_i)$ refers to the total number of road segments that need to be traversed to reach all other road segments starting from r_i.

Connectivity indicates the number of directly connected neighbors to a road segment, which can be calculated as

$$x_i^{(co)} = \mathrm{Degree}(r_i) = \sum_j a_{ij}. \quad (5)$$

If r_i is adjacent to r_j, a_{ij} equals 1; otherwise, it equals 0.

Choice, also known as Betweenness, reflects how often a road segment is likely to be encountered when moving through the space. The Choice for road segment r_i is calculated by counting how many times it lies on the shortest paths between all pairs of segments r_j and r_k, as follows

$$x_i^{(ch)} = \sum_{j,k} \delta_{ijk}, \tag{6}$$

where δ_{ijk} indicates whether the shortest path from r_j to r_k passes r_i, formulated as

$$\delta_{ijk} = \begin{cases} 1, & \text{if } r_i \in \text{ShortestPath}(r_j, r_k), \\ 0, & \text{other.} \end{cases} \tag{7}$$

Finally, The *Space Syntax* features and the basic features of road segments, including length $x_i^{(le)}$, type $x_i^{(tp)}$ and direction $x_i^{(dr)}$, are concatenated as

$$\boldsymbol{x}_i = x_i^{(td)} \| x_i^{(in)} \| x_i^{(co)} \| x_i^{(ch)} \| x_i^{(le)} \| x_i^{(tp)} \| x_i^{(dr)} \| t, \tag{8}$$

where time slices index t is incorporated to account for the varying travel costs of roads over time. Discrete variables are encoded through the embedding layer.

Topological Feature Aggregation The raw road segment features are aggregated by GNNs to obtain representations with richer topological information.

Inductive GCN (Hamilton, Ying, and Leskovec 2017) methods address generalization performance issues through subgraph learning. Sampling a variety of sub-graph samples can help further improve performance, but it is not possible to sample the entire graph due to computational efficiency. A compromise is to use the Metis algorithm, which partitions the entire road network into K subgraphs, and k of them are randomly sampled to form a training batch (Chiang et al. 2019), as follows

$$\begin{aligned} \left\{ \widetilde{\mathcal{G}}_1, \widetilde{\mathcal{G}}_2, \cdots, \widetilde{\mathcal{G}}_K \right\} &= \text{METIS}\left(\mathcal{G}, K\right), \\ \mathcal{G}_k = \text{RandomSample} &\left(\left\{ \widetilde{\mathcal{G}}_1, \widetilde{\mathcal{G}}_2, \cdots, \widetilde{\mathcal{G}}_K \right\}, k \right). \end{aligned} \tag{9}$$

Next, Spatial Aware Graph Attention Networks (SAGAT) is designed to process the subgraph samples and obtain the aggregated representations. The SAGAT is a GATv2 (Brody, Alon, and Yahav 2022) network in which we integrate spatial relationships between road segments to enhance the model's spatial awareness. This process could be formulated as

$$\left\{ \boldsymbol{h}_i \middle| r_i \in \mathcal{G}_k \right\} = \text{SAGAT}\left(\mathcal{G}_k\right). \tag{10}$$

SAGAT is composed of multiple GAT layers stacked together, where the input of the first layer is a linear transformation of the features.

$$\boldsymbol{h}_i^{(0)} = \text{MLP}(\boldsymbol{x}_i). \tag{11}$$

For the layer $l+1$, the attention weights are calculated with the output of last layer $\boldsymbol{h}_i^{(l)}$, as follows

$$\begin{aligned} \alpha_{ij}^{(l+1)} &= \frac{\exp\left(e_{ij}^{(l)}\right)}{\sum_{j' \in \mathcal{N}_i} \exp\left(e_{ij'}^{(l)}\right)}, \\ e_{ij}^{(l)} &= \boldsymbol{a}^\top \sigma \left(\boldsymbol{W}_s \boldsymbol{h}_i^{(l)} + \boldsymbol{W}_t \boldsymbol{h}_j^{(l)} + \boldsymbol{W}_e \boldsymbol{s}_{ij} \right), \end{aligned} \tag{12}$$

where $\boldsymbol{a}$, $\boldsymbol{W}_s$, $\boldsymbol{W}_t$, $\boldsymbol{W}_e$ are learnable parameters, $\mathcal{N}_i$ is the neighbors of the road segment r_i and σ is LeaklyReLU function. $\boldsymbol{s}_{ij}$ represents the spatial relationships between road segment r_i and r_j, formulated as

$$\boldsymbol{s}_{ij} = \text{Bet}(r_i, r_j) \| \text{Angle}(r_i, r_j) \| \text{Dist}(r_i, r_j), \tag{13}$$

where $\text{Bet}(r_i, r_j)$ is the Betweenness of road segment pair r_i and r_j, which is defined as the ratio of the number of shortest paths that traverse r_i and r_j to the total number of shortest paths in the entire network. $\text{Angle}(r_i, r_j)$ represents the turning angle, and $\text{Dist}(r_i, r_j)$ represents the travel distance from the center of road segment.

The output of layer $l+1$ is the weighted aggregation of the representations of neighboring nodes, as follows

$$\boldsymbol{h}_i^{(l+1)} = \sigma \left(\sum_{j \in \mathcal{N}_i} \alpha_{ij}^{(l+1)} \boldsymbol{h}_j^{(l)} \right) + \boldsymbol{h}_i^{(l)}, \tag{14}$$

where σ is the ReLU function.

3.3 Travel Cost Prediction with Disentangled Representation

After aggregating the topological features, we aim to predict the travel cost based on the representation of the road segments while ensuring good generalization to the target city.

The difference in representation distribution between source and target cities reducing the model's generalization ability. To address this, disentangled learning and adversarial domain adaptation are used to create city-invariant representations. Building on (Cai et al. 2019), the method assumes road segment information is determined by two independent latent variables: a semantic latent variable $\boldsymbol{z}^{(s)}$ and a domain latent variable $\boldsymbol{z}^{(d)}$. These are extracted using a semantic and a domain encoder, respectively. $\boldsymbol{z}^{(s)}$ captures semantic information for predicting trajectory costs, while $\boldsymbol{z}^{(d)}$ contains city-specific domain information, as follows

$$\begin{aligned} \boldsymbol{z}_i^{(s)} &= \text{SemEncoder}\left(\boldsymbol{h}_i\right), \\ \boldsymbol{z}_i^{(d)} &= \text{DomEncoder}\left(\boldsymbol{h}_i\right). \end{aligned} \tag{15}$$

The adversarial domain adaptation technique is used to train these representations, incorporating a predictor for travel cost estimation and a discriminator for city identification.

Travel Cost Prediction The travel cost, represented by the average travel time and speed of road segments during specific periods, is denoted as

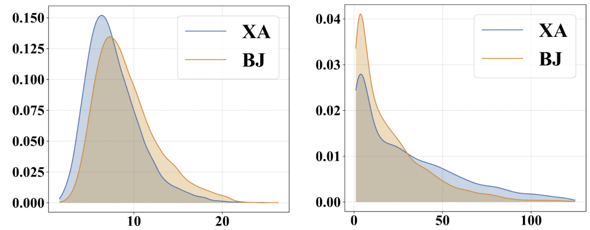
$$\boldsymbol{y}_i = \left\{ y_i^{(m)} \middle| m \in \{\text{time, speed}\} \right\}, \tag{16}$$

where m denotes the type of cost. Using the disentangled latent variable $\boldsymbol{z}_i$ as input (with superscripts omitted for simplicity), the travel cost prediction network is formulated as

$$\hat{\boldsymbol{y}}_i = \log\left(1 + \exp\left(\text{MLP}\left(\boldsymbol{z}_i\right)\right)\right), \tag{17}$$

The prediction loss function comprises a MSE loss and a Rank loss. The MSE loss is

$$\mathcal{L}_{mse} = \frac{1}{N_s} \sum_i \left\| \hat{\boldsymbol{y}}_i - \boldsymbol{y}_i \right\|^2, \tag{18}$$



(a) Travel Speed (m/s) (b) Travel Time (s)

Figure 3: Similar local topological structures in New York (left) and Chicago (right).

where N_s is the size of source city dataset.

Rank loss (Burges et al. 2005) is motivated by the challenge of directly predicting absolute travel costs in the cross-city task, where potential biases across different cities can complicate predictions (see Figure 3). This approach emphasizes predicting relative rankings, which is generally less complex than estimating absolute values.

The probability that r_i has a higher travel cost than r_j is

$$\hat{q}_{ij}^{(m)} = \text{Sigmoid}\left(\hat{y}_i^{(m)} - \hat{y}_j^{(m)}\right). \tag{19}$$

The actual ranking label of sample pair (r_i, r_j) is

$$q_{ij}^{(m)} = \begin{cases} 1, & y_i^{(m)} > y_j^{(m)}, \\ 0, & y_i^{(m)} < y_j^{(m)}. \end{cases} \tag{20}$$

The binary cross-entropy loss is calculated as follows

$$\begin{aligned} l\left(\hat{q}, q\right) &= -\left(q \log(\hat{q}) + (1-q)\log(1-\hat{q})\right), \\ \mathcal{L}_{rank} &= \frac{1}{N_s^2}\sum_i \sum_j \sum_m l\left(\hat{q}_{ij}^{(m)}, q_{ij}^{(m)}\right). \end{aligned} \tag{21}$$

The overall loss for the travel cost prediction is

$$\mathcal{L}_{pred} = \mathcal{L}_{mse} + \lambda_r \mathcal{L}_{rank}, \tag{22}$$

where λ_r is a balance weight between the above two loss.

Domain Discrimination To separate domain information from semantic information, we introduce a discriminator to predict the domain label of the road segment. We extract subgraph sample from source city or target city and assign a domain label to each road segment, as follows

$$d_i = \begin{cases} 1, & r_i \in \mathcal{G}^{(src)}, \\ 0, & r_i \in \mathcal{G}^{(tgt)}. \end{cases} \tag{23}$$

Given the latent variable $\boldsymbol{z}_i$ as input, the domain discriminator is formulated as

$$\hat{d}_i = \text{Sigmoid}\left(\text{MLP}\left(\boldsymbol{z}_i\right)\right). \tag{24}$$

Binary cross entropy loss is used for this domain discrimination task, as follows

$$\mathcal{L}_{dis} = -\frac{1}{N_s + N_t}\sum_i \left(d_i \log \hat{d}_i + (1-d_i)\log(1-\hat{d}_i)\right), \tag{25}$$

where N_s and N_t represent the sizes of the source city and target city datasets, respectively.

Disentangled Adversarial Training Adversarial training is used to promote the information decoupling of $z_i^{(s)}$ and $z_i^{(s)}$. The ultimate objective for semantic latent variables is to minimize travel cost prediction loss while maximizing domain discrimination loss, whereas for domain latent variables, the goal is the opposite. By using different representations as input, the loss functions are calculated as follows

$$\begin{aligned} \mathcal{L}_{total}^{(s)} &= \mathcal{L}_{pred}\left(z_i^{(s)}\right) - \lambda_d \mathcal{L}_{dis}\left(z_i^{(s)}\right), \\ \mathcal{L}_{total}^{(d)} &= \lambda_d \mathcal{L}_{dis}\left(z_i^{(d)}\right) - \mathcal{L}_{pred}\left(z_i^{(d)}\right). \end{aligned} \tag{26}$$

As shown in Figure 2, the GRL layer is used to negate the gradient to achieve efficient adversarial training, which performs an identity transformation in the forward propagation and negates the gradient in the back propagation. In addition, the orthogonal loss is introduced to further reduce information coupling, calculated as follows

$$\mathcal{L}_{og} = \frac{1}{N_s + N_t}\sum_i \left(\frac{\boldsymbol{z}_i^{(s)} \cdot \boldsymbol{z}_i^{(d)}}{\left\|\boldsymbol{z}_i^{(s)}\right\| \cdot \left\|\boldsymbol{z}_i^{(d)}\right\|}\right)^2. \tag{27}$$

The total loss function for disentangled domain adaptation combines these losses

$$\mathcal{L}_{total} = \mathcal{L}_{total}^{(s)} + \mathcal{L}_{total}^{(d)} + \lambda_g \mathcal{L}_{og}. \tag{28}$$

3.4 Travel Preference Learning

After completing the travel cost prediction, we can use the shortest path search algorithm to generate trajectories for the target city.

Shortest path search algorithms rely on fixed road cost factors, such as travel speed or time, for route planning. However, focusing on a single cost factor often fails to fully capture users' actual travel preferences, which are influenced by more complex factors (Chen et al. 2018).

To address this, we propose modeling travel preferences as a combination of observable costs and hidden costs. Hidden costs, which account for harder-to-explain factors influencing human choices, are generated using a multi-layer perceptron (MLP) as follows

$$y_i^{(hid)} = \log\left(1 + \exp\left(\text{MLP}(\boldsymbol{z}_i)\right)\right), \tag{29}$$

where $\boldsymbol{z}_i$ is a semantic latent variable. We then estimate overall travel preference using a weighted combination of observable and hidden costs

$$p(r_i) = \sum_m w^{(m)} y_i^{(m)} + y_i^{(hid)}, \tag{30}$$

where $w^{(m)}$ are learnable weights. The smaller the value of $p(r_i)$, the higher the preference for the road segment r_i. We believe this method of combining preferences remains consistent across different cities.

The preference is learned by an unsupervised training. During training, we first randomly initialize the parameters and search for the shortest path. The shortest path found from r_i to r_j is denoted as $\hat{\tau}_{ij} = (r_i, \cdots, r_j)$, whose preference sum is

$$\hat{p}\left(\hat{\tau}_{ij}\right) = \sum_{r_k \in \hat{\tau}_{ij}} p(r_k). \tag{31}$$

The sum of the preference values of the real trajectory τ_{ij} is

$$p\left(\tau_{ij}\right) = \sum_{r_k \in \tau_{ij}} p(r_k). \tag{32}$$

And then the loss function is formulated as

$$\mathcal{L}_{pref} = \frac{1}{\left|\mathcal{T}^{(src)}\right|} \sum_{\tau_{ij} \in \mathcal{T}^{(src)}} \left(p\left(\tau_{ij}\right) - \hat{p}\left(\hat{\tau}_{ij}\right)\right). \tag{33}$$

Through iterative training, the model learns the invariant mapping relationship between travel preferences and various travel costs, which can then be applied to the target city to generate trajectory data. It is worth mentioning that assigning a cost to each road segment is similar to the MaxEnt IRL (Wulfmeier, Ondruska, and Posner 2015). Theoretical analysis can be found in the code repository.

4 Experiments

4.1 Experimental Settings

Experiments are conducted on three real trajectory datasets. The experiments consists of four parts: New City Trajectory Generation, Downstream Task Support, Target City Finetune and Ablation Study.

Datasets and Preprocessing Three real-world trajectory datasets are used to evaluate the performance of our proposed method. These datasets were collected in the three cities, namely Beijing(BJ), Xi'an(XA) and Chengdu(CD).

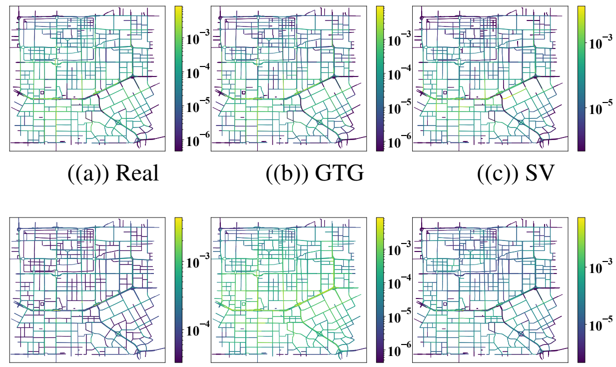
Road network of the three cities are collected from the OpenStreetMap (OpenStreetMap contributors 2017), and road segment trajectories are obtained by performing map-matching algorithm (Yang and Gidofalvi 2018).

Comparative Baselines Baseline models can be categorized into two types, knowledge-driven and data-driven.

- Knowledge-Driven Methods: The knowledge-driven methods are manually proposed by researchers based on the analysis of trajectory data. Subsequently, they integrate proposed rules with a random walk algorithm to generate trajectory data. This type of baseline includes Random Walk (Grover and Leskovec 2016)(RW), Density-EPR (Pappalardo et al. 2015)(DE) and Spatial-EPR (Pappalardo, Rinzivillo, and Simini 2016)(SE).
- Data-Driven Methods: TrajGen (Cao and Li 2021)(TG), SeqGAN (Yu et al. 2017)(SG), SVAE (Huang et al. 2019)(SV), MoveSim (Feng et al. 2020)(MS), TS-TrajGen (Jiang et al. 2023b)(TT), DiffTraj (Zhu et al. 2023)(DT) and VOLUNTEER (Long et al. 2023)(VO).

Evaluation Metrics Evaluation metrics are divided into two categories, macroscopic and microscopic.

- Macro metrics: we use the Jensen-Shannon divergence(JSD) to evaluate the similarity between the generated trajectory dataset and the real dataset across three statistical features: travel distance(Distance), radius of gyration(Radius), and road segment visit frequency(LocFreq).
- Micro metrics: we calculate the sequence distance between the each generated trajectory and its corresponding real trajectory to evaluate similarity. We use four types of sequence distance: Hausdorff, DTW, EDT and EDR.



((a)) Real ((b)) GTG ((c)) SV

((d)) DE ((e)) DT ((f)) VO

Figure 4: The visualization of road segment visit frequency on the XA dataset, brighter color means higher frequency.

4.2 New City Trajectory Generation Experiment

The overall performance result is shown in Table 1. In our experiments across each dataset, the best results are highlighted in bold, while the second-best results are underlined.

The proposed method demonstrates superior performance compared to all baseline models on the three real-world trajectory datasets, evaluated from both macro and micro perspectives. Significant improvements are observed across all metrics with the proposed approach. Unlike other deep learning baseline methods that require training on the target city's data to produce accurate trajectories, GTG can operate effectively without training in target city. This demonstrates the model's generalization capability.

Additionally, all evaluation metrics exhibit smaller values, suggesting that the generated trajectory data closely resembles the real trajectory data. Figure 4 shows the generated datasets and real datasets for some baselines. GTG achieves the highest similarity to the real dataset.

4.3 Downstream Task Support

Trajectory data is frequently used in many downstream tasks, such as next-hop prediction (Feng et al. 2018; Sun et al. 2020), trajectory classification (Wang et al. 2018a; Chen et al. 2019), and trajectory recovery (Wang et al. 2019a). The downstream task experiment serves as a supplementary verification of the model's generalization ability and underscores the practical significance of the trajectory generation task. In this experiment, the trajectory generation model is employed to produce pre-training data for the downstream task, thereby enhancing the performance.

Target	Metric	RW	DE	SE	TG	SG	SV	MS	TT	DT	VO	GTG[1]	GTG[2]
BJ	Distance	0.0434	0.1509	0.1662	0.2980	0.2320	0.0080	0.1826	0.0107	0.0173	0.1893	**0.0006**	0.0006
	Radius	0.1191	0.1332	0.1543	0.2180	0.2070	0.0098	0.1450	0.0071	0.0019	0.0518	**0.0001**	0.0002
	LocFreq	0.198	0.076	0.196	0.187	0.038	0.044	0.410	0.095	0.216	0.145	**0.041**	0.043
	Hausdorff	2.682	3.833	3.807	7.228	10.469	3.294	7.523	1.092	3.305	5.680	**0.292**	0.315
	DTW	51.54	58.88	58.49	185.17	149.49	61.99	145.38	24.27	74.26	89.82	**4.89**	5.27
	EDT	28.17	29.85	29.90	42.93	27.31	26.20	31.07	18.13	31.42	29.17	**8.73**	9.17
	EDR	0.813	0.904	0.906	0.820	0.903	0.751	0.942	0.423	0.889	0.850	**0.190**	0.205
XA	Distance	0.0524	0.2122	0.2233	0.4271	0.1369	0.0584	0.3018	0.0085	0.0386	0.1450	0.0044	**0.0040**
	Radius	0.0708	0.1422	0.1612	0.4199	0.0655	0.0268	0.1795	0.0011	0.0030	0.0588	**0.0002**	0.0002
	LocFreq	0.263	0.104	0.264	0.403	0.126	0.099	0.290	0.097	0.180	0.207	0.042	**0.040**
	Hausdorff	2.300	2.406	2.627	2.852	2.319	1.597	3.161	0.349	1.006	2.178	0.188	**0.187**
	DTW	34.02	36.35	38.04	56.34	26.20	17.66	30.69	5.28	12.09	23.52	**2.32**	2.34
	EDT	24.21	30.28	30.36	34.11	17.67	13.42	17.95	7.10	14.88	15.85	**4.29**	4.23
	EDR	0.818	0.888	0.893	0.711	0.808	0.595	0.871	0.243	0.668	0.680	0.142	**0.135**
CD	Distance	0.0646	0.2083	0.2069	0.3596	0.1420	0.0405	0.2560	0.0136	0.0440	0.1380	0.0051	**0.0049**
	Radius	0.0958	0.1127	0.0980	0.1259	0.0619	0.0221	0.1598	0.0017	0.0053	0.0162	**0.0002**	0.0002
	LocFreq	0.280	0.110	0.267	0.210	0.103	0.051	0.374	0.069	0.159	0.098	0.027	**0.026**
	Hausdorff	1.434	1.981	1.980	2.104	2.433	1.273	3.126	0.220	1.033	1.530	**0.117**	0.125
	DTW	18.87	24.82	24.32	28.75	28.65	13.94	35.44	2.54	13.52	15.62	**1.19**	1.34
	EDT	19.42	25.64	25.83	20.16	18.27	14.02	18.42	6.69	17.48	15.01	**3.59**	3.63
	EDR	0.778	0.865	0.869	0.720	0.801	0.620	0.900	0.213	0.679	0.657	**0.109**	0.117

Table 1: New City Trajectory Generation Results

GTG[1] refers to the GTG model trained on XA for BJ, BJ for CD, and CD for XA. GTG[2] refers to the model trained on BJ for XA, XA for CD and CD for BJ.

Specifically, the trajectory next-location prediction task is selected as the downstream application. This task, which aims to predict the next location in a trajectory given several observed locations, is widely applicable in POI recommendation systems. The models DeepMove (Feng et al. 2018) and LSTPM (Sun et al. 2020), implemented via LibCity (Wang et al. 2021a), are utilized for this purpose. These downstream models are trained using the trajectory data generated by the proposed model, and their performance is subsequently tested on real trajectory data.

The results, as presented in Table 2, are evaluated using three metrics: Accuracy (ACC), Normalized Discounted Cumulative Gain at 3 (NDCG@3), and Mean Reciprocal Rank at 3 (MRR@3). The performance of the data generated by GTG in training downstream tasks is found to be second only to that of real data. The baseline model demonstrates inferior performance, indicating that the trajectory data generated by our approach is more effective in supporting downstream tasks.

4.4 Target City Fine-tune

Considering that the gradual collection of trajectory data is a more realistic application scenario, it would be helpful if the trajectory generation capability of the model could further adjust in the target city. We fine-tune the model using the trajectory of the target city to test its improved generation ability. Fine-tuning phase training include travel cost prediction and preference learning. The experimental results of fine-tuning in XA are shown in the Table 3. The results in other cities can be found in our code repository.

From Table 3, we can see that using target city data for fine-tuning improves the model's performance. Before applying the model to a new city, collecting a small amount of trajectory data for fine-tuning can achieve good generation results without incurring excessive costs.

4.5 Ablation Study

To validate the effectiveness of submodules, we conduct the following ablation studies.

(a) w/o Cost: we removed the cost prediction module, which means that instead of using combined costs shown in formula 30, we only use hidden costs as road weights to generate trajectories.

(b) w/o Pref: we removed the preference learning module, meaning that we now only use supervise learning to predict observable costs and use their sum as weights for trajectory generation.

(c) w/o SS: we removed the *Space Syntax* feature extraction module to test the impact of *Space Syntax* features., which means that only the basic features of road segments are input into the model.

The ablation study results in target city XA are shown in Table 4. Upon the removal of the aforementioned submodules, a notable decline in the model's performance was observed, with the cost prediction module having the most pronounced impact. Ablation experiments that excluded the cost prediction and preference learning modules demonstrated that the integration of travel cost components more accurately captures the invariant travel patterns of humans. Additionally, experiments that removed the *Space Syntax* feature extraction module revealed that *Space Syntax* significantly contributes to the cross-city trajectory generation.

5 Related Work

Existing trajectory generation works can be divided into two categories: Knowledge-Driven methods and Data-Driven methods.

Knowledge-driven methods models human mobility based on prior knowledge and statistical data. Gravity Model (Zipf 1946) and Intervening Opportunities Model (Stouffer 1940) generate trajectories by modeling the

Model	Data	BJ ACC	BJ NDCG	BJ MRR	XA ACC	XA NDCG	XA MRR	CD ACC	CD NDCG	CD MRR
DeepMove	Real	0.81	0.88	0.86	0.88	0.94	0.93	0.89	0.95	0.93
	RW	0.60	0.75	0.71	0.62	0.83	0.78	0.62	0.85	0.80
	DE	0.01	0.04	0.03	0.01	0.11	0.08	0.02	0.11	0.08
	SE	0.01	0.02	0.02	0.01	0.03	0.03	0.01	0.05	0.03
	TG	0.56	0.61	0.60	0.46	0.47	0.46	0.69	0.74	0.73
	SG	0.67	0.73	0.71	0.64	0.70	0.68	0.82	0.89	0.87
	SV	0.64	0.68	0.67	0.78	0.81	0.80	0.82	0.88	0.87
	MS	0.01	0.02	0.02	0.11	0.13	0.13	0.11	0.13	0.18
	TT	0.65	0.73	0.71	0.71	0.78	0.76	0.75	0.83	0.81
	DT	0.14	0.20	0.18	0.39	0.52	0.49	0.41	0.51	0.49
	VO	0.22	0.31	0.29	0.42	0.49	0.47	0.60	0.69	0.67
	GTG [1]	0.72	0.80	0.78	**0.82**	0.87	**0.85**	**0.84**	**0.90**	**0.89**
	GTG [2]	**0.73**	**0.80**	**0.79**	0.80	**0.85**	0.84	0.83	0.89	0.88
LSTPM	Real	0.85	0.93	0.91	0.90	0.96	0.95	0.89	0.96	0.94
	RW	0.68	0.86	0.82	0.64	0.85	0.80	0.61	0.84	0.79
	DE	0.01	0.05	0.04	0.01	0.10	0.08	0.00	0.11	0.08
	SE	0.00	0.03	0.02	0.00	0.03	0.02	0.00	0.05	0.04
	TG	0.66	0.70	0.69	0.62	0.66	0.65	0.78	0.84	0.83
	SG	0.80	0.87	0.85	0.75	0.82	0.81	0.85	0.93	0.91
	SV	0.79	0.86	0.84	0.83	0.88	0.87	0.87	0.93	0.92
	MS	0.04	0.05	0.05	0.13	0.16	0.15	0.15	0.17	0.17
	TT	0.74	0.85	0.83	0.77	0.85	0.84	0.80	0.89	0.87
	DT	0.17	0.26	0.23	0.43	0.56	0.53	0.45	0.58	0.55
	VO	0.30	0.38	0.36	0.49	0.56	0.54	0.64	0.73	0.71
	GTG [1]	**0.80**	**0.88**	**0.87**	**0.86**	**0.91**	**0.90**	**0.87**	**0.94**	**0.92**
	GTG [2]	0.78	0.87	0.60	0.85	0.90	0.89	0.87	0.93	0.92

Table 2: Downstream Task Support Experiment Result

City	#Traj (10^3)	Distance (10^{-3})	Radius (10^{-3})	LocFreq	Hausdorff	DTW	EDT	EDR
BJ → XA	0.0	4.039	0.228	0.040	0.187	2.34	4.23	0.135
	0.1	4.679	0.825	0.040	0.194	2.52	4.36	0.143
	0.4	4.875	0.306	0.044	0.188	2.30	4.38	0.147
	1.6	4.233	0.334	0.036	0.184	2.21	4.17	0.139
	6.4	3.976	0.139	0.029	0.174	2.07	4.00	0.129
	12.8	4.034	0.104	0.027	0.170	1.95	3.96	0.125
CD → XA	0.0	4.375	0.206	0.042	0.188	2.32	4.29	0.142
	0.1	4.468	0.345	0.050	0.192	2.45	4.67	0.148
	0.4	5.951	0.464	0.067	0.207	2.72	5.05	0.159
	1.6	4.285	0.175	0.043	0.181	2.13	4.38	0.139
	6.4	4.103	0.219	0.033	0.171	1.95	4.09	0.130
	12.8	4.128	0.253	0.029	0.170	1.93	4.01	0.128

Table 3: Target City Fine-tune Experiment Results in XA

City	Method	Distance (10^{-3})	Radius (10^{-3})	LocFreq	Hausdorff	DTW	EDT	EDR
BJ → XA	w/o Cost	19.171	5.066	0.119	0.264	3.78	8.51	0.260
	w/o Pref	4.356	0.293	0.047	0.198	2.54	4.61	0.146
	w/o SS	3.685	0.673	0.041	0.199	2.57	4.49	0.144
	GTG	4.039	0.228	0.040	0.187	2.34	4.23	0.135
CD → XA	w/o Cost	17.745	3.763	0.118	0.274	3.52	6.98	0.227
	w/o Pref	4.641	1.052	0.062	0.219	3.00	5.03	0.164
	w/o SS	5.261	2.817	0.059	0.228	3.15	4.95	0.163
	GTG	4.375	0.206	0.042	0.188	2.32	4.29	0.142

Table 4: Ablation Study Results in XA

relationship between inter-regional mobility intensity and population and economic data. The EPR model (Pappalardo et al. 2015; Pappalardo, Rinzivillo, and Simini 2016) regards trajectory generation as a process of multiple explorations and returns, and the probability of an individual visiting a location is related to regional attractiveness. This type of method emerged along with traditional modeling methods. It has a coarse granularity but has certain generalization capabilities.

Data-driven methods uses large amounts of trajectory data to train neural network models, capturing complex travel patterns. According to different architectures, these algorithms can be divided into algorithms based on Seq2Seq (Park et al. 2018), GAN (Yu et al. 2017), VAE (Long et al. 2023; Huang et al. 2019), and Diffusion (Zhu et al. 2023). The Seq2Seq method uses RNN, TCN, Transformer to model the movement state of individual, then predict the choice of individuals facing multiple optional road segments. As an improvement to Seq2Seq, Neural A* (Jiang et al. 2023b) searches multiple trajectories simultaneously to provide the most likely trajectory. Similarly, the VAE method models the individual's movement state and travel prefernece through latent variables. In order to better generate trajectories with composite human movement distribution, the GAN method leverage a discriminator to identify the authenticity of the trajectory. Different from the method of generating trajectory component by component, Diffusion innovatively treats the trajectory of human movement as a special texture and generates the entire trajectory at once using image generation. This type of method focuses on statistical learning and has generalization capabilities in the same city but cannot be generalized to other cities.

In other areas of urban data mining, such as traffic flow prediction and OD (origin-destination) demand prediction (Ji et al. 2023; Jiang et al. 2023a; Wang et al. 2022), there have been numerous active studies on cross-city transfer learning. (Wang et al. 2018b) calculates matching scores between regions using other data and forces the model to generate similar representations for matched region pairs by adding a matching regularization term. CrossTRes (Jin, Chen, and Yang 2022) conducts transfer learning using data from multiple cities. To avoid negative transfer caused by data from dissimilar cities, this method introduces a weighted score for each city to weight the losses from multiple cities. Yu Zheng et al. (He et al. 2020) proposed an OD transfer learning model primarily focused on generating travel demand for new cities. Research on tranfer learning in other fields inspired us to propose GTG .

6 Conclusion

In this paper, we propose a novel, generalizable trajectory generation model that leverages invariant human mobility patterns. The model incorporates *Space Syntax* theory as a feature input and innovatively applies methods such as inductive graph convolution and disentangled learning to capture these complex mobility patterns. Across a wide range of experimental scenarios, baseline models, and evaluation metrics, this method consistently and significantly outperforms the baselines, which demonstrates the model's exceptional generalization ability. In the future work, we will further enhance the model by allowing trajectory generation without training on the target city's road network.

Acknowledgements

Prof. Jingyuan Wang's work was supported by the National Natural Science Foundation of China (No. 72171013, 72222022, 72242101), and the Special Fund for Health Development Research of Beijing (2024-2G-30121).

References

Brody, S.; Alon, U.; and Yahav, E. 2022. How Attentive are Graph Attention Networks? In *Proceedings of the 10th International Conference on Learning Representations*.

Burges, C.; Shaked, T.; Renshaw, E.; Lazier, A.; Deeds, M.; Hamilton, N.; and Hullender, G. 2005. Learning to Rank Using Gradient Descent. In *Proceedings of the 22nd International Conference on Machine Learning*, 89–96.

Cai, R.; Li, Z.; Wei, P.; Qiao, J.; Zhang, K.; and Hao, Z. 2019. Learning Disentangled Semantic Representation for Domain Adaptation. In *Proceedings of the 33rd International Joint Conference on Artificial Intelligence*, 2060. NIH Public Access.

Cao, C.; and Li, M. 2021. Generating Mobility Trajectories with Retained Data Utility. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2610–2620.

Chen, L.; Gao, Y.; Fang, Z.; Miao, X.; Jensen, C. S.; and Guo, C. 2019. Real-time Distributed Co-movement Pattern Detection on Streaming Trajectories. *Proceedings of the VLDB Endowment*, 12(10): 1208–1220.

Chen, L.; Gao, Y.; Li, X.; Jensen, C. S.; and Chen, G. 2017. Efficient Metric Indexing for Similarity Search and Similarity Joins. *IEEE Transactions on Knowledge and Data Engineering*, 29(3): 556–571.

Chen, L.; Zhong, Q.; Xiao, X.; Gao, Y.; Jin, P.; and Jensen, C. S. 2018. Price-and-time-aware Dynamic Ridesharing. In *Proceedings of the 34th International Conference on Data Engineering*, 1061–1072. IEEE.

Chiang, W.-L.; Liu, X.; Si, S.; Li, Y.; Bengio, S.; and Hsieh, C.-J. 2019. Cluster-GCN: An efficient algorithm for training deep and large graph convolutional networks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 257–266.

Ding, X.; Chen, L.; Gao, Y.; Jensen, C. S.; and Bao, H. 2018. UlTraMan: A Unified Platform for Big Trajectory Data Management and Analytics. *Proceedings of the VLDB Endowment*, 11(7): 787–799.

Feng, J.; Li, Y.; Zhang, C.; Sun, F.; Meng, F.; Guo, A.; and Jin, D. 2018. DeepMove: Predicting Human Mobility with Attentional Recurrent Networks. In *Proceedings of the 2018 World Wide Web Conference*, 1459–1468.

Feng, J.; Yang, Z.; Xu, F.; Yu, H.; Wang, M.; and Li, Y. 2020. Learning to Simulate Human Mobility. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 3426–3433.

Grover, A.; and Leskovec, J. 2016. Node2Vec: Scalable Feature Learning for Networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 855–864.

Hamilton, W.; Ying, Z.; and Leskovec, J. 2017. Inductive Representation Learning on Large Graphs. *Advances in Neural Information Processing Systems*, 30.

He, T.; Bao, J.; Li, R.; Ruan, S.; Li, Y.; Song, L.; He, H.; and Zheng, Y. 2020. What Is the Human Mobility in a New City: Transfer Mobility Knowledge Across Cities. In *Proceedings of the 29th Web Conference*, 1355–1365.

Hettige, K. H.; Ji, J.; Xiang, S.; Long, C.; Cong, G.; and Wang, J. 2024. AirPhyNet: Harnessing Physics-Guided Neural Networks for Air Quality Prediction. *arXiv preprint arXiv:2402.03784*.

Hillier, B.; Leaman, A.; Stansall, P.; and Bedford, M. 1976. Space Syntax. *Environment and Planning B: Planning and design*, 3(2): 147–185.

Huang, D.; Song, X.; Fan, Z.; Jiang, R.; Shibasaki, R.; Zhang, Y.; Wang, H.; and Kato, Y. 2019. A Variational Autoencoder Based Generative Model of Urban Human Mobility. In *Proceedings of the 2nd IEEE Conference on Multimedia Information Processing and Retrieval*, 425–430.

Ji, J.; Wang, J.; Huang, C.; Wu, J.; Xu, B.; Wu, Z.; Zhang, J.; and Zheng, Y. 2023. Spatio-temporal Self-supervised Learning for Traffic Flow Prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 4356–4364.

Ji, J.; Wang, J.; Wu, J.; Han, B.; Zhang, J.; and Zheng, Y. 2022. Precision CityShield Against Hazardous Chemicals Threats via Location Mining and Self-supervised Learning. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*.

Jiang, J.; Han, C.; Zhao, W. X.; and Wang, J. 2023a. PDFormer: Propagation Delay-aware Dynamic Long-range Transformer for Traffic Flow Prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 4365–4373.

Jiang, W.; Zhao, W. X.; Wang, J.; and Jiang, J. 2023b. Continuous Trajectory Generation Based on Two-stage GAN. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 4374–4382.

Jin, Y.; Chen, K.; and Yang, Q. 2022. Selective Cross-City Transfer Learning for Traffic Prediction via Source City Region Re-Weighting. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 731–741.

Kong, X.; Chen, Q.; Hou, M.; Wang, H.; and Xia, F. 2023. Mobility Trajectory Generation: a Survey. *Artificial Intelligence Review*, 56(Suppl 3): 3057–3098.

Liu, Z.; Wang, J.; Li, Z.; and He, Y. 2024. Full Bayesian Significance Testing for Neural Networks in Traffic Forecasting. In *Proceedings of the 33rd International Joint Conference on Artificial Intelligence*.

Long, Q.; Wang, H.; Li, T.; Huang, L.; Wang, K.; Wu, Q.; Li, G.; Liang, Y.; Yu, L.; and Li, Y. 2023. Practical synthetic human trajectories generation based on variational point processes. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 4561–4571.

OpenStreetMap contributors. 2017. Planet Dump Retrieved from https://planet.osm.org . https://www.openstreetmap.org.

Pappalardo, L.; Rinzivillo, S.; and Simini, F. 2016. Human Mobility Modelling: Exploration and Preferential Return Meet the Gravity Model. *Procedia Computer Science*, 83: 934–939.

Pappalardo, L.; Simini, F.; Rinzivillo, S.; Pedreschi, D.; Giannotti, F.; and Barabási, A.-L. 2015. Returners and Explorers Dichotomy in Human Mobility. *Nature Communications*, 6(1): 8166.

Park, S. H.; Kim, B.; Kang, C. M.; Chung, C. C.; and Choi, J. W. 2018. Sequence-to-Sequence Prediction of Vehicle Trajectory via LSTM Encoder-Decoder Architecture. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, 1672–1678.

Stouffer, S. A. 1940. Intervening Opportunities: A Theory Relating Mobility and Distance. *American Sociological Review*, 5(6): 845–867.

Sun, K.; Qian, T.; Chen, T.; Liang, Y.; Nguyen, Q. V. H.; and Yin, H. 2020. Where to Go Next: Modeling Long-and short-term User Preferences for Point-of-interest Recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 214–221.

Wang, J.; He, X.; Wang, Z.; Wu, J.; Yuan, N. J.; Xie, X.; and Xiong, Z. 2018a. CD-CNN: A Partially Supervised Cross-domain Deep Learning Model for Urban Resident Recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Wang, J.; Ji, J.; Jiang, Z.; and Sun, L. 2022. Traffic Flow Prediction Based on Spatiotemporal Potential Energy Fields. *IEEE Transactions on Knowledge and Data Engineering*, 35(9): 9073–9087.

Wang, J.; Jiang, J.; Jiang, W.; Li, C.; and Zhao, W. X. 2021a. LibCity: An Open Library for Traffic Prediction. In *Proceedings of the 29th International Conference on Advances in Geographic Information Systems*, 145–148.

Wang, J.; Lin, X.; Zuo, Y.; and Wu, J. 2021b. DGeye: Probabilistic Risk Perception and Prediction for Urban Dangerous Goods Management. *ACM Transactions on Information Systems (TOIS)*, 39(3): 1–30.

Wang, J.; Wang, X.; and Wu, J. 2018. Inferring Metapopulation Propagation Network for Intra-city Epidemic Control and Prevention. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery and data mining*.

Wang, J.; Wu, N.; Lu, X.; Zhao, W. X.; and Feng, K. 2019a. Deep Trajectory Recovery with Fine-grained Calibration using Kalman Filter. *IEEE Transactions on Knowledge and Data Engineering*, 33(3): 921–934.

Wang, J.; Wu, N.; and Zhao, W. X. 2021. Personalized Route Recommendation with Neural Network Enhanced Search Algorithm. *IEEE Transactions on Knowledge and Data Engineering*, 34(12): 5910–5924.

Wang, J.; Wu, N.; Zhao, W. X.; Peng, F.; and Lin, X. 2019b. Empowering A* Search Algorithms with Neural Networks for Personalized Route Recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 539–547.

Wang, L.; Geng, X.; Ma, X.; Liu, F.; and Yang, Q. 2018b. Cross-City Transfer Learning for Deep Spatio-Temporal Prediction. *ArXiv Preprint arXiv:1802.00386*.

Wu, N.; Wang, J.; Zhao, W. X.; and Jin, Y. 2019. Learning to Effectively Estimate the Travel Time for Fastest Route Recommendation. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*.

Wulfmeier, M.; Ondruska, P.; and Posner, I. 2015. Maximum Entropy Deep Inverse Reinforcement Learning. *ArXiv Preprint arXiv:1507.04888*.

Yang, C.; and Gidofalvi, G. 2018. Fast Map Matching, an Algorithm Integrating Hidden Markov Model with Precomputation. *International Journal of Geographical Information Science*, 32(3): 547–570.

Yu, L.; Zhang, W.; Wang, J.; and Yu, Y. 2017. SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*, volume 31.

Zhu, Y.; Ye, Y.; Zhang, S.; Zhao, X.; and Yu, J. 2023. DiffTraj: Generating GPS Trajectory with Diffusion Probabilistic Model. In *Proceedings of 37th Conference on Neural Information Processing Systems*.

Zipf, G. K. 1946. The P1*P2/D Hypothesis: On the Intercity Movement of Persons. *American sociological review*, 11(6): 677–686.