UniMove: A Unified Model for Multi-city Human Mobility Prediction

Chonghua Han*
Department of Electronic Engineering, BNRist, Tsinghua University, Beijing, China
hanch24@mails.tsinghua.edu.cn

Yuan Yuan*†
Department of Electronic Engineering, BNRist, Tsinghua University, Beijing, China
y-yuan20@tsinghua.org.cn

Yukun Liu
Department of Electronic Engineering, BNRist, Tsinghua University, Beijing, China
liuyk21@mails.tsinghua.edu.cn

Jingtao Ding
Department of Electronic Engineering, BNRist, Tsinghua University, Beijing, China
dingjt15@tsinghua.org.cn

Jie Feng
Department of Electronic Engineering, BNRist, Tsinghua University, Beijing, China
fengjie@tsinghua.edu.cn

Yong Li†
Department of Electronic Engineering, BNRist, Tsinghua University, Beijing, China
liyong07@tsinghua.edu.cn

ABSTRACT

Human mobility prediction is vital for urban planning, transportation optimization, and personalized services. However, the inherent randomness, non-uniform time intervals, and complex patterns of human mobility, compounded by the heterogeneity introduced by varying city structures, infrastructure, and population densities, present significant challenges in modeling. Existing solutions often require training separate models for each city due to distinct spatial representations and geographic coverage. In this paper, we propose UniMove, a unified model for multi-city human mobility prediction, addressing two challenges: (1) constructing universal spatial representations for effective token sharing across cities, and (2) modeling heterogeneous mobility patterns from varying city characteristics. We propose a trajectory-location dual-tower architecture, with a location tower for universal spatial encoding and a trajectory tower for sequential mobility modeling. We also design MoE Transformer blocks to adaptively select experts to handle diverse movement patterns. Extensive experiments across multiple datasets from diverse cities demonstrate that UniMove truly embodies the essence of a unified model. By enabling joint training on multi-city data with mutual data enhancement, it significantly improves mobility prediction accuracy by over 10.2%. UniMove represents a key advancement toward realizing a true foundational model with a unified architecture for human mobility. We release the implementation at https://github.com/tsinghua-fib-lab/UniMove/.

CCS CONCEPTS

• **Computing methodologies** → *Machine learning*; • **Information systems** → **Mobile information processing systems**.

KEYWORDS

Human mobility, unified model, multi-city prediction

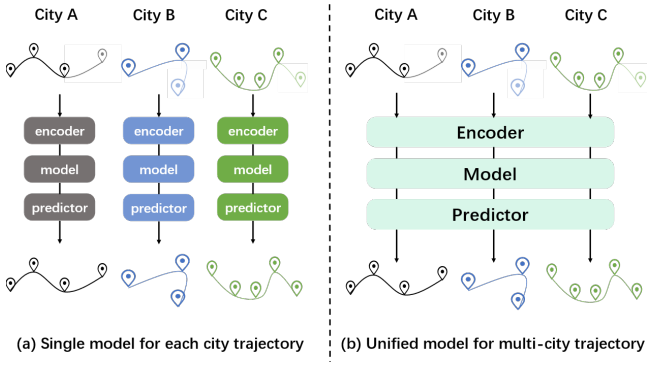

Figure 1: Comparison between existing solutions and our approach.

1 INTRODUCTION

Human mobility [22], which record the movement patterns of individuals across time and space, have gained substantial importance in numerous fields, including urban planning [30, 31, 54, 55], transportation management [41, 42, 53], mobile network optimization [3, 27] and personalized services [18, 48, 52, 56]. Human mobility presents unique challenges in modeling due to their inherent randomness, non-uniform time intervals, and complex patterns [43]. Unlike vehicle trajectories [57], which typically follow more predictable and regular paths with time intervals on the order of seconds, human mobility data is more irregular and context-dependent.

To effectively tackle such data and harness the wealth of mobility information, researchers often treat a mobility trajectory as a sequence of location tokens, with each city represented by a set of unique tokens, similar to a vocabulary in language processing. These tokens are typically discretized using grid-based representations [21, 49], administrative divisions [43], or points of interest (POIs) [8, 26], providing a structured approach to analyzing movement patterns. Then, advanced AI models, such as

*Equal contribution
†Corresponding authors

RNNs [8, 46], transformers [21, 28], and other deep learning architectures [26, 44, 57], can be utilized to learn the underlying patterns and dynamics of mobility trajectories. However, these tokens do not overlap across cities due to the unique geographic coverage of each urban area. As a result, practitioners often need to train separate models for different geographic regions, which limits generalizability and practical applicability. While some exceptions exist, such as transfer learning approaches [12, 38, 39, 58], they either rely on vehicle trajectories (where location representations, such as numerical latitude and longitude, are transferable) [58] or just use more tokens [12] or separate encoders [38] to address geographic differences. Some methods, such as ST-MoE-BERT [13], force the grid cells of different cities into the same number of classes to process mobility data from different cities with a shared encoder, which severely erodes the distinct spatial identities of individual cities.

In this paper, we aim to propose a solution for training a unified model with fully shared parameters across different cities, which means that the model can accommodate the unique geographic characteristics of each urban area, while simultaneously learning shared mobility patterns that benefit from one another. We compare our model with existing practices in Figure 1. However, this is a non-trivial task, and it faces three main challenges: (1) Token sharing across cities: since each city has its own unique geographic coverage, designing a tokenizer and creating a universal spatial representation that can be shared across different cities presents a significant challenge. (2) Variations in city scale and structure: cities differ greatly in terms of scale, population density, and infrastructure. Ensuring that the model can perform well across cities with such diverse urban structures, from large metropolises to smaller towns, is a difficult task. (3) Heterogeneity in mobility data: mobility data varies widely between cities, which presents a challenge in designing a model that can effectively learn from these diverse data sources and enable meaningful knowledge transfer across cities.

To address the challenges above, we propose a unified mobility model **UniMove**. UniMove is a trajectory-location dual-tower architecture. The Location Tower generates universal spatial representations based on location characteristics, utilizing these features to define locations and employing a Deep & Cross Net to enhance their representation. The Trajectory Tower captures mobility patterns from sequential trajectories. To handle diverse movement patterns across cities, we design MoE Transformer blocks that adaptively select expert networks based on data characteristics. Extensive experiments demonstrate that UniMove achieves superior performance compared to baseline models. Moreover, the mutual enhancement among data from multiple cities substantially boosts the model's performance, which suggests that UniMove effectively captures shared mobility patterns across cities and leverages them for mutual benefit.

In summary, our contribution can be summarized as follows:

- To our best knowledge, we are the first to explore the potential for mutual enhancement among human mobility data across different cities.
- We propose a unified human mobility prediction model **UniMove**. We propose a trajectory-location dual-tower architecture. The location tower generates universal spatial representations based on location characteristics while the trajectory tower extracts mobility patterns with MoE Transformers.
- Extensive experiments on multiple real-world datasets demonstrate the superior performance of UniMove, which outperforms state-of-the-art baselines by 8.9% in terms of accuracy. Meanwhile, the unified model facilitates data mutual enhancement and exhibits faster convergence.

2 RELATED WORK

2.1 Human Mobility Prediction

Human mobility prediction has been extensively studied using various modeling techniques, each aiming to capture the intricate dynamics of human movement. Real-world applications of human mobility prediction span a variety of domains, including smart cities [36, 47], traffic management [41, 42], personalized recommendations [18, 52, 56], and public health (e.g., predicting the spread of diseases) [4, 11, 25]. Early methods relied heavily on probabilistic models, such as the Markov model [6, 9] and EPR-based models [16, 29], where mobility patterns are derived from the transition probabilities between discrete locations. These approaches, while effective in some contexts, often fail to capture the complex, dynamic nature of human behavior [8, 43], especially when dealing with non-uniform, irregular time intervals between events. Recently, deep learning models have significantly advanced mobility prediction. Models such as recurrent neural networks (RNNs) [5], attention networks [8, 44], Transformers [7, 28], Graph Neural Networks (GNNs) [32, 33], diffusion models [20, 21, 57] have been employed to capture the temporal dependencies inherent in mobility data. However, these models typically require training with large-scale data. Since mobility patterns are highly localized and influenced by unique geographic, cultural, and infrastructural factors, they necessitate training separate models for each city's data.

2.2 Location Encoding

Location encoding [24] plays a crucial role in human mobility prediction by transforming spatial data into a form that can be processed by deep neural networks. Location encoders can be divided into two main categories: continuous location encoding and discrete location encoding. For continuous encoding, locations are treated as continuous values, often represented by continuous coordinates (e.g., latitude, longitude) [45, 46, 51, 59]. These continuous values can be directly embedded into the model through continuous embeddings [2, 51, 59] or positional encodings [44, 46], which are typically processed via neural networks to capture spatial relationships in a continuous space. In discrete location encoding, the continuous spatial domain is often divided into predefined locations, such as Points of Interest (POI) [8, 26], Areas of Interest (AOI) [50], administrative districtc [43], or grids [21, 43]. Each of these locations is assigned a discrete identifier, and the locations are then represented using techniques like one-hot encoding [1]. In this way, each location is transformed into a unique token, much like words in language models. Discrete encoding simplifies the representation, making it more computationally efficient while still allowing the model to process and distinguish between different locations effectively.

2.3 Cross-City Modeling of Human Mobility

Cross-city modeling of human mobility aims to leverage knowledge and patterns learned from one city to improve mobility modeling in another, addressing the challenges posed by data scarcity and the unique characteristics of individual cities [10, 14, 15, 38, 39]. For instance, He *et al.* [14] propose a top-down paradigm that utilizes abundant annotated data to plan mobility for a new city. Additionally, meta-learning frameworks [35] have been developed to enable models to adapt quickly to new cities with limited data [10, 38]. However, existing approaches still face significant limitations. For example, they often fail to address the challenge of using a single shared model (with identical parameters) across different cities. Current methods either rely on separate location embedding layers for each city [38] or are restricted to handling vehicle trajectories with numerical coordinate values [58, 59]. Unlike vehicle trajectories, which primarily rely on numerical GPS coordinates (e.g., latitude and longitude), human mobility trajectories are inherently richer in semantics, encompassing contextual information such as PoIs, activities, and urban infrastructure. This requires to address the tokenization of locations across cities, ensuring that models can effectively generalize and adapt to diverse urban environments.

3 PRELIMINARY

3.1 Human Mobility Data

Human mobility datasets are non-uniform trajectories with variable lengths. The relevant definitions are provided as follows:

Definition of Location: A location is a grid area within a city, uniquely identified by a specific location ID. The characteristics of each location include its geographical coordinates, the distribution of Points of Interest (POI), and popularity. The latitude and longitude coordinates pinpoint the exact location of the area within the city. The distribution of POIs reflects the functional nature of the area. The popularity, measured by the number of visits during a certain period, is a key indicator of its appeal and activity level.

Definition of Mobility Trajectory: The mobility trajectory of a user, denoted as S_u, can be represented as $\{\text{loc}_1, t_1; \text{loc}_2, t_2; \ldots; \text{loc}_n, t_n\}$, where loc_i represents the location ID where the user stays, and t_i represents the start time when the user first appears at location loc_i. Since precise time information at the millisecond level is redundant and even noisy for our task, we divide a day into 48 time slots at half-hour intervals and map t_i to the nearest discrete time point. As each user's mobility pattern is different, although the trajectory sequences are within the same time window T, the length n of each sequence is not fixed, and the time interval between adjacent locations t_i and t_{i+1} in each trajectory is also not fixed.

3.2 Mobility Prediction

In the context of mobility prediction, a typical scenario involves leveraging a variable number of historical records to forecast the subsequent future location. Specifically, given a sequence of historical mobility trajectories denoted as $\{S_t\}_{t=1}^{H}$, where H represents the length of the historical time steps and is not fixed, the objective is to predict the future location loc_{H+1}, This task can be formally defined as learning a mapping function $\mathcal{F}$ that transforms the historical sequence into the future location, which can be mathematically represented as:

$$\hat{loc}_{H+1} = \mathcal{F}(S_{1:H}) \tag{1}$$

where $\hat{loc}_{H+1}$ signifies the predicted location for the next stay point, and $S_{1:H}$ denotes the historical records spanning the time period from 1 to H. The primary goal is to accurately predict the next future location based on the historical data provided.

The mobility prediction problem bears resemblance to the next token prediction in the close task of natural language processing(NLP). However, unlike NLP, where multiple datasets can utilize the same vocabulary, each city's mobility trajectory dataset varies in the number and features of locations. When applied to multiple datasets, existing methods require different encoders and predictors for each city, and train the model from scratch on the new dataset. It can be represented as:

$$\hat{loc}_{H+1} = \mathcal{F}_i(S_{1:H}) \quad S \in \mathcal{D}_i, \mathcal{D} = \{\mathcal{D}_1, \mathcal{D}_2, \ldots\} \tag{2}$$

where $\mathcal{D}$ is a collection of mobility trajectory datasets from multiple cities. Our motivation is to develop a prediction model that can be applied to datasets from multiple cities without any modification, which can be mathematically represented as:

$$\hat{loc}_{H+1} = \mathcal{F}(S_{1:H}) \quad S \in \mathcal{D}, \mathcal{D} = \{\mathcal{D}_1, \mathcal{D}_2, \ldots\} \tag{3}$$

Table 1: Details of POI Categories

Auto Service
Auto Dealers
Auto Repair
Motocycle Service
Food & Beverages
Shopping
Daily Life Service
Sports & Recreation
Medical Service
Accommodation Service
Tourist Attraction
Commercial House
Governmental Organization
Science/Culture & Education Service

4 METHODOLOGY

4.1 Overall Framework

We propose a unified mobility prediction framework UniMove, which adopts a dual-tower structure, as shown in the Figure 2. The Trajectory Tower predicts the intent embedding $\mathbf{I} \in \mathbb{R}^{B\times T\times d}$ of the next location based on the user's historical trajectories. The Location Tower generates candidate locations embedding $\mathbf{L} \in \mathbb{R}^{N\times d}$ based on the characteristics of locations within the city. the predicted location probability distribution $\mathbf{M} \in \mathbb{R}^{B\times T\times N}$ is obtained by element-wise multiplying $\mathbf{L}$ and $\mathbf{I}$. The framework comprises three main components:

- **Location Encoder**:The Location Encoder takes the regional features of a location as input. Compared to existing methods

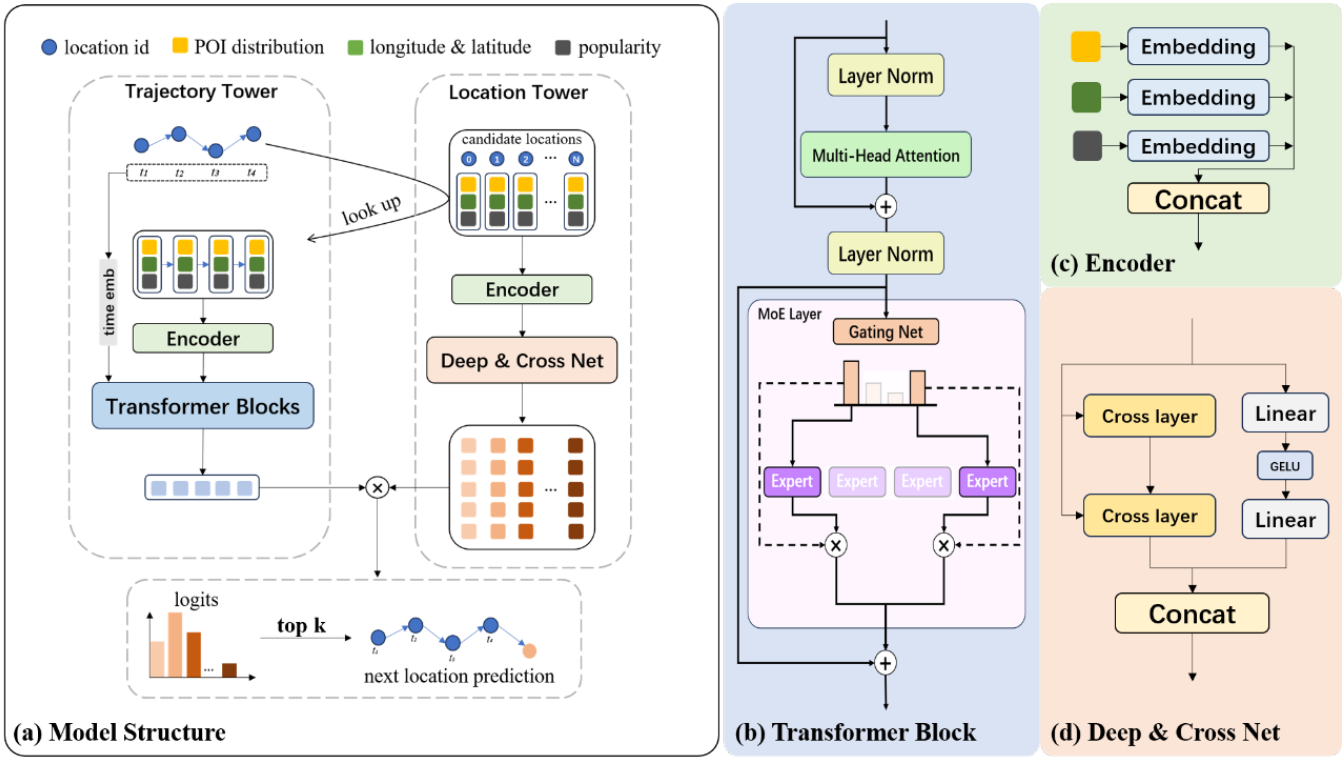

Figure 2: The overall architecture of UniMove. (a)Trajectory Tower and Location Tower:Trajectory Tower predicts the latent embedding of the next location based on historical trajectories, Location Tower generates the feature representations for all candidate locations. (b)MoE(Mixture of Experts) Transformer Block. (c)Location Encoder. (d)Deep & Cross Net for further capturing features from different locations.

Table 2: Details of Popularity Rank and Value

Popularity Rank	R value
<1%	0
1%-5%	1
5%-10%	2
10%-20%	3
20%-40%	4
40%-60%	5
60%-80%	6
>80%	7

that directly use location IDs as feature inputs, this approach can be applied to extract features from locations in different cities without altering any parameters

- **Deep & Cross Net**:We introduce DCN in the Location Tower to model the features of locations within the city. This approach not only enables automatic learning of complex interactions between features but also further extracts unique characteristics of different locations through feature crossing, thereby providing more complex feature representations for subsequent prediction tasks.
- **MoE Transformer Block**: To address the heterogeneous distribution of mobility data between different cities, UniMove utilize MoE transformer, which assigns different data distributions to specialized expert models. This reduces conflicts caused by data heterogeneity and improves the model's generalization capability.

4.2 Location Encoder

Each location is defined as a grid area, and the location embedding is obtained from three features: POI distribution **P**, geographical coordinates **G**, and popularity rank **R**.

- POI distribution$\mathbf{P} \in \mathbb{R}^{2c}$, where c represents the number of POI categories. The details of POI categories are shown in Table 1. The POI distribution is encoded into the form $[n_1, n_2, \ldots, n_c, p_1, p_2, \ldots, p_c]$, where n_i denotes the number of POIs in the i-th category, and $p_i = \frac{n_i}{\sum_{j=1}^{c} n_j}$
- Geographical coordinates$\mathbf{G} \in \mathbb{R}^2$ refer to the latitude and longitude of the location's center point.
- $\mathbf{R} \in \mathbb{R}^1$ is a discrete integer value.Based on the flow data of all locations within the city, each location is ranked according to its popularity.The **R** value is assigned based on the percentile ranking: locations in the top 1% are assigned a value of 1, those between 1% and 5% are assigned a value of 2, and so on.The details of popularity rank and value are shown in Table 2.

We then create three embedding layers to obtain the following embeddings: POI embedding $\mathbf{E}_p \in \mathbb{R}^{B\times T\times \frac{d}{2}}$, geographical embedding $\mathbf{E}_g \in \mathbb{R}^{B\times T\times \frac{d}{4}}$, and flow rank embedding $\mathbf{E}_r \in \mathbb{R}^{B\times T\times \frac{d}{4}}$. The location embedding $\mathbf{E}_l \in \mathbb{R}^{B\times T\times d}$ is defined as the concatenation of these three embeddings:

$$\mathbf{E}_l = \text{concat}(\mathbf{E}_p, \mathbf{E}_g, \mathbf{E}_r) \tag{4}$$

4.3 Deep & Cross Net

Location Tower introduces the Deep & Cross Net(DCN) to further capture features from different locations. It consists of two main parts: the Cross layer and the Deep layer. The design of the Cross layer is inspired by [37], aiming to capture high-order feature interactions through feature crossing. Suppose the input feature vector is location embedding $\mathbf{E_l}$, with a dimension of d. The Cross layer calculates as follows:

$$\mathbf{E}_{\text{cross}} = \sum_{i=1}^{d} \mathbf{E}_i \odot \mathbf{W}_i \mathbf{E}_l + \mathbf{b}_i \tag{5}$$

where $\odot$ represents element-wise multiplication, and $\mathbf{W}_i$ is the learnable weight matrix. In this way, the Cross layer can effectively capture the nonlinear relationships between features. The Deep layer further processes features through a multi-layer perceptron (MLP) structure. Its calculation method is as follows:

$$\mathbf{E}_{\text{deep}} = (\text{GELU}(\mathbf{W}_1\mathbf{E_l} + \mathbf{b}_1))\mathbf{W}_2 + \mathbf{b}_2 \tag{6}$$

where $\mathbf{W}_1$ and $\mathbf{W}_2$ are learnable weight matrices, $\mathbf{b}_1$ and $\mathbf{b}_2$ are bias terms, and GELU is an activation function that introduces nonlinearity. The candidate locations embedding $\mathbf{L} \in \mathbb{R}^{N\times d}$ is obtained by concatenating the outputs of the Cross layer and the Deep layer. The specific expression is as follows:

$$\mathbf{L} = \text{Concat}(\mathbf{E}_{\text{cross}}, \mathbf{E}_{\text{deep}}) \tag{7}$$

4.4 MoE Transformer Blocks

The Trajectory Tower is composed of MoE transformer blocks that integrate multi-head masked attention and Mixture-of-Experts (MoE) layers. It is designed to model mobility patterns and predict the intent embedding $\mathbf{I}$ for the next location based on the user's historical trajectory. This process can be formulated as:

$$\mathbf{I} = \text{Transformer}(\mathbf{E}_l + \mathbf{E}_t), \tag{8}$$

where $\mathbf{E}_l$ represents the embedding of the user's historical trajectory and $\mathbf{E}_t \in \mathbb{R}^{B\times T\times d}$ denotes the embedding derived from the timestamp information at each trajectory point.

4.4.1 Masked multi-head attention. The masked multi-head attention mechanism in our model incorporates two types of masks to ensure proper information flow and handle variable-length trajectories:

- **Causal Mask**: The causal mask ensures that the prediction at the i_{th} location is constrained to utilize information solely from the first $i-1$ locations. Specifically, it enforces a unidirectional dependency structure, where each location is influenced exclusively by its preceding locations. This design is essential for modeling sequential patterns in trajectories, as it effectively prevents the leakage of future information and preserves the temporal order inherent in the trajectory data.
- **Padding Mask**: Since the lengths of user trajectories vary, we standardize the input length for the model by padding all trajectories to a fixed length of T. To achieve this, we append a special termination token (represented by 1) at the end of each trajectory to signify its termination point. Subsequently, the remaining positions are filled with padding tokens (represented by 0) to reach the desired length. This padding strategy ensures that the model can process trajectories of different lengths while maintaining the integrity of the original data structure.

4.4.2 Mixture of Experts (MoE). The MoE is composed of a gating network and multiple expert networks, which serve as an alternative to the Feed-Forward Network (FFN) in the Transformer architecture. The gating network computes the activation probabilities for each expert based on the input x, producing a probability distribution vector $G(x)$. The output of the gating network is calculated as follows:

$$G(x) = \text{Softmax}(\text{TopK}(x \cdot W_g)) \tag{9}$$

where W_g is the weight matrix of the gating network, and $\text{TopK}(\cdot)$ denotes the operation of selecting the top k largest values from the input vector, with the remaining values set to $-\infty$ to ensure that only the top k experts are activated.

During the training of MoE model, it is common for the model to favor a single expert, leading to load imbalance among the experts. To address this issue, noise is added to the output of the gating neural network. This helps to balance the load among different experts by preventing the model from consistently routing inputs to the same expert. Specifically, standard normal noise is added to the logits produced by the gating network.This noise is scaled by a factor derived from a separate linear layer, which allows for adjustable noise levels, which can be expressed as follows:

$$\text{noise} = \mathcal{N}(0,1) \times \text{Softplus}(x \cdot W_n) \tag{10}$$

where $\mathcal{N}(0,1)$ represents the standard normal noise, and W_n is the weight matrix of the additional noise linear layer. The noisy logits are then used to select the top-k experts, ensuring that the selection process is more randomized and less biased towards any particular expert. So the final logits of the gating network can be expressed as

$$G(x) = \text{Softmax}(\text{TopK}(x \cdot W_g + \text{noise})) \tag{11}$$

This method not only promotes load balancing but also enhances the model's robustness and generalization ability. Each expert network produces an output $E_i(x)$, where i indicates the i-th expert. For each input x, only the k selected experts by the gating network will compute their outputs.

Ultimately, the output of the MoE layer H_{moe} is the weighted sum of the outputs from the selected k experts, with the weights determined by the gating network's output:

$$H_{\text{moe}} = \sum_{i=1}^{k} G_i(x) \cdot E_i(x) \tag{12}$$

This sparse activation mechanism ensures that only a subset of experts participate in the computation, the dynamic selection by the gating network allows the model to choose the most suitable experts based on the characteristics of the input, enhancing the model's adaptability and generalization capability.

Algorithm 1 Training

Require: Trajectory Dataset $D = \{D_1, D_2, \ldots, D_M\}$, Mobility prediction model F, and loss function L.
Ensure: Learnable parameters θ for the model F

for $epoch \in \{1, 2, \ldots, N_{iter}\}$ **do**
  Randomly sample a mini-batch x from D_m.
  Trajectory data $x_i = (P_i, G_i, R_i, t_i)$.
  location data $l = (P^N, G^N, R^N)$,
  $E_p, E_g, E_r, E_t \leftarrow \text{emb}(P_i), \text{emb}(G_i), \text{emb}(R_i), \text{emb}(t_i)$
  $I = \text{Transformer}(\text{concat}(E_p, E_g, E_r) + E_t)$
  $M_p, M_g, M_r \leftarrow \text{emb}(P^N), \text{emb}(G^N), \text{emb}(R^N)$
  $L = \text{DCN}(\text{concat}(M_p, M_g, M_r))$
  logits $= I \odot L$
  Compute the Cross Entropy loss $\mathcal{L} \leftarrow L(logits, x)$
  Update the model's parameters $\theta \leftarrow update(\mathcal{L}; \theta)$
end for

4.5 Training

As summarized in the algorithm 1, during the training process, a batch of trajectory data is extracted from the dataset of a random city. The user's historical trajectory data is fed into the Transformer-based Trajectory Tower to derive the intent embedding$\mathbf{I} \in \mathbb{R}^{B\times T\times d}$for the next location prediction. Concurrently, the features of all locations within the city are input into the Location Tower to generate the location embedding $\mathbf{L} \in \mathbb{R}^{N\times d}$. The final **logits** $\in \mathbb{R}^{B\times T\times N}$ is computed through element-wise multiplication of L and I, with the logits calculation being as follows:

$$\text{logits}_{b,t,n} = \sum_{i=1}^{d} I_{b,t,i} \cdot L_{n,i} \tag{13}$$

and the loss is expressed as:

$$Loss = \text{Cross-Entropy Loss}(\text{logits}, x) \tag{14}$$

where $\mathbf{x} \in \mathbb{R}^{B\times T}$ is the trajectory location id.

5 PERFORMANCE EVALUATIONS

5.1 Experimental Settings

5.1.1 Datasets. We conduct extensive experiments on three real-world mobility datasets: Shanghai, Nanchang, and Lhasa. Detailed information on the data sets is summarized in Table 3. For pre-processing the trajectory data in these datasets, we applied a sliding window of three days to each user's trajectory and filtered out trajectories with fewer than five trajectory points. In the location pre-processing phase, we map GPS points to predefined grid area IDs with a granularity of 500 m × 500 m and record features such as the POI distribution within each grid area. For every location, POI data is sourced from Gaode Map and consists of 14 major categories. The longitude and latitude of all locations in the same city are normalized to have a mean of 0 and a standard deviation of 1. We discretize the popularity rank based on quantiles. For temporal pre-processing, we organized the time data into fixed half-hourly intervals. Finally, we divided the data into training, validation, and testing sets based on different users, with a ratio of 6 : 2 : 2.

Table 3: Basic statistics of mobility data.

City	Duration	Location	Trajectory
Shanghai	14 days	5451	483200
Nanchang	7 days	2055	26400
Lhasa	28 days	166	48160

5.1.2 Metrics. Evaluation metric Acc@K is utilized to measure the proportion of samples that are correctly predicted within the top K highest-probability locations by the model.The formula for Acc@k can be expressed as follows:

$$\text{Acc@}k = \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}\left(y_i \in \{f(x_i)_1, f(x_i)_2, \ldots, f(x_i)_k\}\right) \tag{15}$$

where $\mathbb{I}(\cdot)$ is the indicator function (1 if the condition is true, 0 otherwise), N is the total number of samples, y_i is the true label for the i-th sample, $f(x_i)_1, \ldots, f(x_i)_k$ are the top k predictions made by the model for the sample x_i.

5.1.3 Baselines. To evaluate the performance of our method, we compared it with state-of-the-art models.The details are as follows:

- **Linear**:Linear model employs the linear regression to predict the probability distribution of the next location.
- **Markov** [9]:The Markov model serves as a statistical framework designed to depict how states evolve over time. It predicts future locations by computing the probabilities of transitions.
- **LSTM** [17]:LSTM networks excel at processing sequential data and are adept at capturing long-term dependencies, making them naturally suitable for location prediction tasks.
- **Transformer** [34]:Transformer is a deep learning architecture based on the self-attention mechanism, capable of efficiently processing sequential data and capturing long-range dependencies.
- **DeepMove** [8]:This approach is particularly effective in addressing the challenges of predicting human movement patterns, which are influenced by various factors such as time, location, and personal preferences
- **TrajBert** [28]:TrajBERT is a trajectory recovery method based on BERT, which recovers implicit sparse trajectories through spatial-temporal refinement.
- **STAN** [23]: STAN employs a spatio-temporal bi-attention architecture to capture non-adjacent location correlations and non-consecutive visit dependencies.
- **GETNext** [40]: GETNext integrates Graph Convolutional Networks (GCN) and Transformer to model global POI transition patterns via a trajectory flow map. It fuses spatio-temporal contexts with time-aware category embeddings for next POI recommendation.
- **CTLE** [19]: The Context and Time aware Location Embedding (CTLE) model dynamically generates context-specific location embeddings using bidirectional Transformers and temporal encoding to capture multi-functional properties of locations under varying contexts and temporal dynamics.

5.1.4 Implementation Details. The transformer blocks utilized in this study comprises 8 layers, with an embedding dimension of 512. The MoE layer within the model consists of 4 experts and employs

Table 4: Performance comparison of different prediction models in terms of Acc@k, where bold denotes best results.

	Nanchang			Lhasa			Shanghai		
Model	**Acc@1**	**Acc@3**	**Acc@5**	**Acc@1**	**Acc@3**	**Acc@5**	**Acc@1**	**Acc@3**	**Acc@5**
Linear	0.125	0.300	0.398	0.253	0.500	0.622	0.105	0.239	0.326
Markov	0.166	0.353	0.402	0.288	0.557	0.688	0.154	0.303	0.406
LSTM	0.224	0.416	0.541	0.366	0.601	0.702	0.206	0.353	0.459
Transformer	0.225	0.448	0.543	0.380	0.614	0.715	0.232	0.404	0.454
DeepMove	0.243	0.420	0.503	0.401	0.610	0.690	0.214	0.365	0.445
TrajBert	0.249	0.423	0.505	0.353	0.571	0.672	0.246	0.411	**0.489**
CTLE	0.252	0.435	0.508	0.413	0.588	0.682	0.260	0.405	0.445
STAN	0.244	0.422	0.507	0.384	0.600	0.685	0.247	0.409	0.469
GETNext	0.258	0.438	0.522	0.408	0.619	0.713	0.262	0.400	0.477
UniMove(separate)	0.261	0.455	0.506	0.421	0.557	0.679	0.244	0.405	0.462
UniMove	**0.338**	**0.509**	**0.573**	**0.425**	**0.646**	**0.722**	**0.263**	**0.412**	0.476

Table 5: Results of the ablation study.

	Nanchang			Lhasa			Shanghai		
Model	**Acc@1**	**Acc@3**	**Acc@5**	**Acc@1**	**Acc@3**	**Acc@5**	**Acc@1**	**Acc@3**	**Acc@5**
w/o dcn	0.316	0.492	0.560	0.414	0.632	0.720	0.243	0.379	0.421
w/o moe	0.321	0.497	0.562	0.405	0.617	0.705	0.261	0.403	0.461
UniMove	**0.338**	**0.509**	**0.573**	**0.425**	**0.646**	**0.722**	**0.263**	**0.412**	**0.476**

a top-k gating mechanism, where $k = 2$. Training was conducted using the AdamW optimizer, with a learning rate set to 3×10^{-4}, for a total of 50 epochs. Early stopping was implemented to terminate training if the validation loss did not improve for 3 consecutive epochs. The batch size was fixed at 16.

5.2 Overall Performance

The overall performance of UniMove is compared with that of the baselines, and the results are presented in Table 4. Based on the experimental findings, we have reached the following conclusions:

- **UniMove outperforms the state-of-the-art baselines in the mobility prediction task.** By integrating the trajectory data of multiple cities for joint training, UniMove can more accurately capture the complex patterns of human mobility, thereby effectively improving the model's prediction performance. Compared with the SOTA baselines, the accuracy of UniMove has been significantly improved, with the highest increase reaching 35.7% in terms of Acc@1 for Nanchang dataset.
- **Data from different cities can complement each other, thereby enhancing the performance of UniMove.** Compared with training UniMove solely on a single dataset, incorporating mobility trajectory data from other cities can significantly enhance the model's prediction accuracy for target city trajectories. Specifically, across multi-city datasets, the performance metrics ACC@1, ACC@3, and ACC@5 have improved by 12.7%, 9.8%, and 7.5%, respectively.
- **The performance improvement of UniMove is more pronounced in the city with sparse mobility data.** For Nanchang city with limited data and incomplete mobility patterns, UniMove can effectively leverage the data resources from data-rich cities with complete mobility patterns, such as Shanghai.UniMove achieves better performance improvements on the sparse Nanchang dataset than on the Shanghai dataset. Specifically, on the Shanghai dataset, UniMove achieves an average improvement of 18.1%, while on the Nanchang dataset, the metrics improved by an average of 4.1%.

5.3 Ablation Study

To investigate the critical roles of different design modules in MobFormer, we conducted ablation studies on three datasets. Specifically, we removed the feature-crossing DCN and the MoE layer from the model architecture. Since completely removing the DCN would lead to the model's inability to capture higher-order location features, we only removed the cross-layer of the DCN and replaced it with an MLP. For the MoE layer, we replaced it with an FFN (Feed-Forward Network) layer.

As shown in Table 5, the experimental results indicate that when replacing the DCN with an MLP, the model performance deteriorated on datasets with a larger number of locations. This phenomenon demonstrates that the DCN has a unique advantage in capturing the differences between locations, allowing it to more

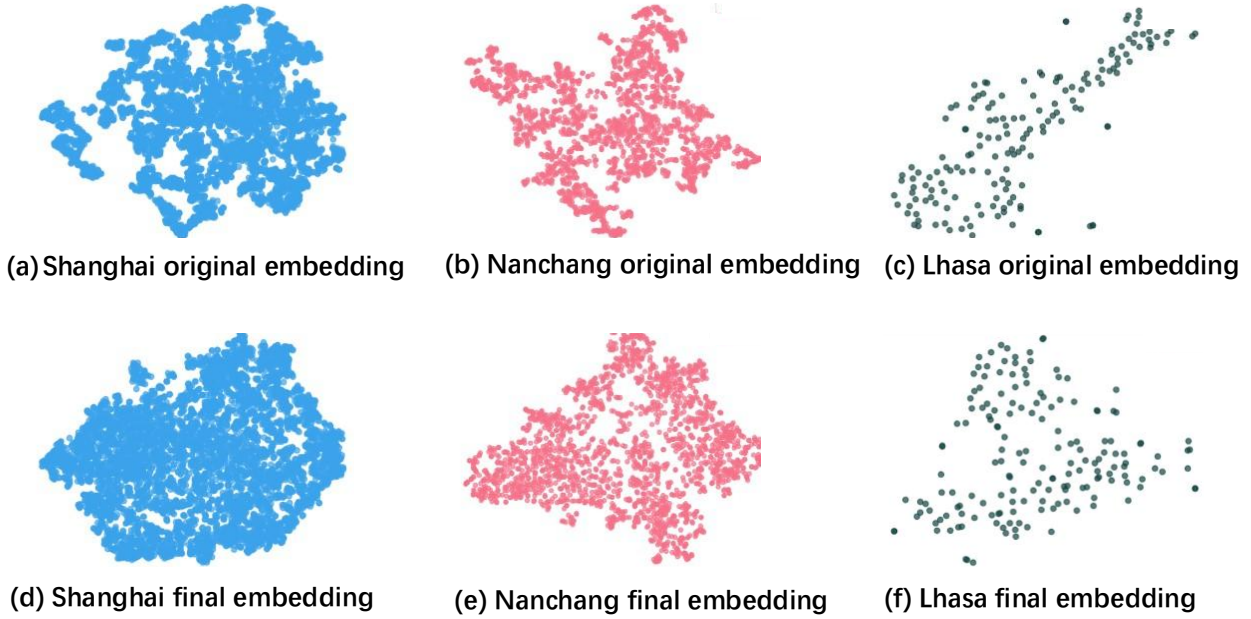

Figure 3: Locations embedding of three cities based on t-SNE visualization. (a)-(c) are original embeddings, and (d)-(f) are final embeddings learned by DCN.

accurately characterize the complex relationships between different locations, which is crucial for model performance.

Moreover, after removing the MoE layer, the performance degradation is more significant on the Nanchang dataset with a smaller data volume. This clearly indicates that the MoE layer plays a key role in handling the heterogeneity of data from different cities. It can effectively integrate data with similar mobility pattern trajectories, thereby enhancing the model's adaptability and generalization ability for cross-city data.

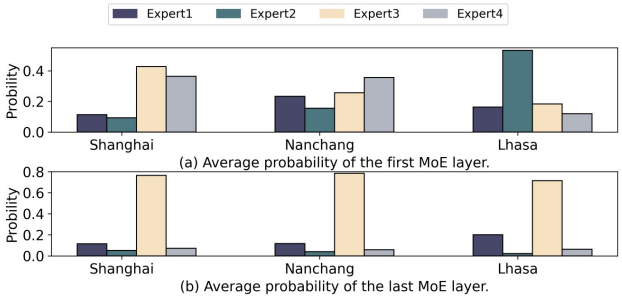

Figure 4: Expert selection distribution in MOE Layers across three city datasets.

5.4 Visualization of Location Embeddings

The location tower extracts the representations of all locations within a city. Given that the distribution of location features varies significantly across different cities, to analyze how the location tower handles this heterogeneity, we select the original embeddings of each city after passing through the Location Encoder and the final location embeddings learned by DCN and use t-SNE dimensionality reduction for visualization.

Figure 3 shows t-SNE visualization analysis of location embeddings for cities, it was observed that after learned by Deep & Cross Network (DCN), the distribution of locations across different cities exhibited higher similarity. This indicates that DCN is capable of effectively extracting similar location feature patterns across different cities, thereby providing strong support for cross-city location analysis and modeling. Meanwhile, within the same city, the distribution of locations is no longer as concentrated as in the original embeddings, and the differences between various locations become more pronounced.

5.5 In-depth Analysis of MoE

To thoroughly investigate whether MoE can capture the common mobility patterns across different cities and to understand its mechanism for handling data heterogeneity, we have conducted a detailed statistical analysis of the expert selection probabilities of different MoE layers across various mobility data.

Figure 4 illustrate the average probabilities of the gating network outputs in the first and last layer for three datasets. It can be observed that there are significant differences in expert selection

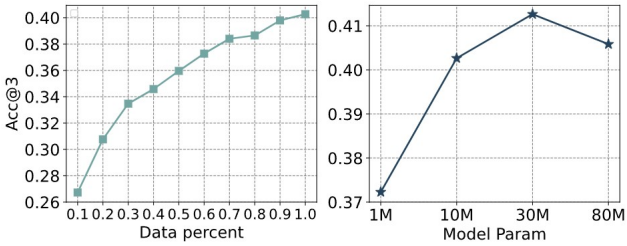

Figure 5: Data scalability(left) and model scalability(right).

Table 6: Acc@1 results of the hyperparameter study.

Expert/Top k	Nanchang	Lhasa	Shanghai
4/2	0.338	0.425	0.263
4/1	0.293	0.403	0.255
4/3	0.334	0.426	0.263
3/2	0.311	0.412	0.260
6/2	0.332	0.426	0.273

among various cities for the first layer, suggesting that the MOE dynamically selects the expert networks to be activated based on the different characteristics across cities. Meanwhile, the last layer exhibit more similar probability distributions, indicating that despite significant heterogeneity among different cities, similar patterns exist at higher-order dimensions.

5.6 Scalability

The high performance of UniMove is attributed to the joint training on multiple city datasets, which provides a richer information base. To verify the performance bottlenecks of the model under different parameter sizes and data volumes, we conducted experiments by testing the model's performance with varying parameter sizes and training data volumes.

Figure 5 shows the impact of varying model parameter scales and dataset sizes on prediction performance. We have identified two characteristics: (i) As the training data volume increases progressively, the model's prediction performance exhibits a continuous upward trend. However, once the data volume reaches a certain threshold, the performance improvement gradually slows down and stabilizes. (ii) When the model's parameter scale is increased, the performance improves incrementally. However, for the dataset used in this study, overly complex model structures lead to faster convergence, which in turn prevents the model from achieving optimal performance.

5.7 Hyperparameter of MoE Analysis

In our exploration of how different hyperparameters within the Mixture of Experts (MoE) framework impact model performance, we conducted experiments by varying the number of experts as well as the top-k values. The experimental outcomes are displayed in Table 6.

Changing the number of experts alters the model's parameter. From the table, a reduction in the number of experts tends to have

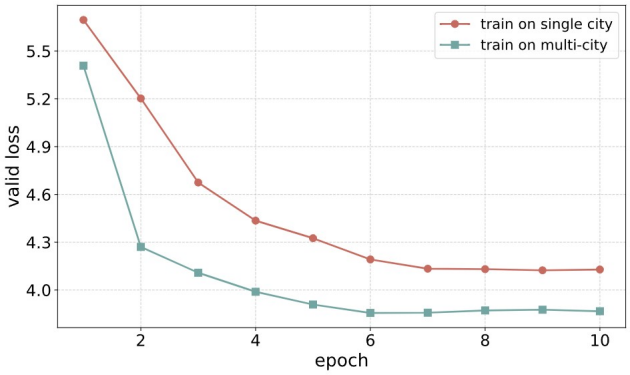

Figure 6: Comparison of training efficiency between single city and multi-city in terms of validation loss.

a detrimental effect on the model's performance. When the number of experts is decreased, the model's capacity to capture complex patterns and diverse features in the data is diminished, thereby leading to suboptimal results. Conversely, increasing the number of experts does not necessarily lead to further improvements in performance. This result is consistent with the analysis in Section 5.6. Due to the limited diversity and quantity of the current training data, adding more experts or model parameters does not enhance the model's performance. Furthermore, the influence of top k and the number of experts on model performance resembles each other. Under the current training data, reducing the top k value leads to a performance drop, while increasing it doesn't significantly boost model performance.

5.8 Training Efficiency

To deeply explore if the multi-city joint training method can boost model performance and training efficiency, and whether it can help the model quickly grasp universal patterns while significantly improving prediction accuracy, we compared the validation loss of the UniMove trained on single city and multi-city datasets. Comparing the validation loss of the UniMove model trained on single-city and multi-city datasets showed in Figure 6 , it demonstrates that joint training across multiple cities significantly enhances the training efficiency of the model. Specifically, within the same epoch, the reduction in loss value is much greater for multi-city training. The complementarity and similarity of data from different cities (such as variations in traffic patterns and user behaviors) provide a richer set of training information. This enables the model to rapidly capture universal patterns and achieve the optimal performance attainable by single-dataset training within fewer epochs. Moreover, it effectively improves the performance of prediction accuracy.

6 CONCLUSION

In this work, we propose UniMove, a transformative approach to human mobility prediction that unlocks the potential for more robust and scalable modeling paradigms. By enabling joint training across multi-city data and promoting mutual enhancement across diverse urban environments, UniMove not only improves prediction

accuracy but also advances the concept of a unified foundational model. Furthermore, the ability to leverage shared mobility insights across cities with varying structures and characteristics paves the way for more generalized and adaptable models that can be applied on a global scale.

While UniMove currently focuses on mobility prediction, its next-token prediction training framework lays a promising foundation for future expansion into other tasks. We plan to extend the model's capabilities to include mobility generation, trajectory recovery, and anomaly detection. Additionally, this architecture holds the potential for integration with large language models, enabling a more seamless understanding of mobility data in natural language contexts. This synergy could facilitate more intuitive human-computer interactions and open new possibilities for urban decision-making.

ACKNOWLEDGMENTS

This work was supported in part by the National Natural Science Foundation of China under 62476152 and U24B20180.

REFERENCES

[1] Jason Brownlee. 2020. Ordinal and one-hot encodings for categorical data. *Machinelearningmastery. com* (2020).

[2] Chu Cao and Mo Li. 2021. Generating mobility trajectories with retained data utility. In *KDD*. 2610–2620.

[3] Haoye Chai, Yuan Yuan, and Yong Li. 2025. MobiWorld: World Models for Mobile Wireless Network. *arXiv preprint arXiv:2507.09462* (2025).

[4] Serina Chang, Emma Pierson, Pang Wei Koh, Jaline Gerardin, Beth Redbird, David Grusky, and Jure Leskovec. 2021. Mobility network models of COVID-19 explain inequities and inform reopening. *Nature* 589, 7840 (2021), 82–87.

[5] Jianwei Chen, Jianbo Li, and Ying Li. 2020. Predicting human mobility via long short-term patterns. *Computer Modeling in Engineering & Sciences* 124, 3 (2020), 847–864.

[6] Meng Chen, Yang Liu, and Xiaohui Yu. 2014. Nlpmm: A next location predictor with markov modeling. In *PAKDD*. Springer, 186–197.

[7] Riccardo Corrias, Martin Gjoreski, and Marc Langheinrich. 2023. Exploring transformer and graph convolutional networks for human mobility modeling. *Sensors* 23, 10 (2023), 4803.

[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 *WWW*. 1459–1468.

[9] Sébastien Gambs, Marc-Olivier Killijian, and Miguel Núñez del Prado Cortez. 2012. Next place prediction using mobility markov chains. In *Proceedings of the first workshop on measurement, privacy, and mobility*. 1–6.

[10] Vinayak Gupta and Srikanta Bedathur. 2022. Doing more with less: Overcoming data scarcity for POI recommendation via cross-region transfer. *TITS* 13, 3 (2022), 1–24.

[11] Zhenyu Han, Haohao Fu, Fengli Xu, Zhen Tu, Yang Yu, Pan Hui, and Yong Li. 2021. Who will survive and revive undergoing the epidemic: Analyses about poi visit behavior in Wuhan via check-in records. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 5, 2 (2021), 1–20.

[12] Haoyu He, Haozheng Luo, and Qi R Wang. 2024. ST-MoE-BERT: A Spatial-Temporal Mixture-of-Experts Framework for Long-Term Cross-City Mobility Prediction. In *Proceedings of the 2nd ACM SIGSPATIAL International Workshop on Human Mobility Prediction Challenge*. 10–15.

[13] Haoyu He, Haozheng Luo, and Qi R. Wang. 2024. ST-MoE-BERT: A Spatial-Temporal Mixture-of-Experts Framework for Long-Term Cross-City Mobility Prediction. *CoRR* abs/2410.14099 (2024). https://doi.org/10.48550/arXiv.2410.14099

[14] Tianfu He, Jie Bao, Ruiyuan Li, Sijie Ruan, Yanhua Li, Li Song, Hui He, and Yu Zheng. 2020. What is the human mobility in a new city: Transfer mobility knowledge across cities. In *Proceedings of The Web Conference 2020*. 1355–1365.

[15] Renhe Jiang, Xuan Song, Zipei Fan, Tianqi Xia, Zhaonan Wang, Quanjun Chen, Zekun Cai, and Ryosuke Shibasaki. 2021. Transfer urban human mobility via poi embedding over multiple cities. *ACM Transactions on Data Science* 2, 1 (2021), 1–26.

[16] Shan Jiang, Yingxiang Yang, Siddharth Gupta, Daniele Veneziano, Shounak Athavale, and Marta C González. 2016. The TimeGeo modeling framework for urban mobility without travel surveys. *Proceedings of the National Academy of Sciences* 113, 37 (2016), E5370–E5378.

[17] Dejiang Kong and Fei Wu. 2018. HST-LSTM: A hierarchical spatial-temporal long-short term memory network for location prediction.. In *Ijcai*, Vol. 18. 2341–2347.

[18] Defu Lian, Yong Ge, Fuzheng Zhang, Nicholas Jing Yuan, Xing Xie, Tao Zhou, and Yong Rui. 2015. Content-aware collaborative filtering for location recommendation based on human mobility data. In *2015 IEEE international conference on data mining*. IEEE, 261–270.

[19] Yan Lin, Huaiyu Wan, Shengnan Guo, and Youfang Lin. 2021. Pre-training context and time aware location embeddings from spatial-temporal trajectories for user next location prediction. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 35. 4241–4248.

[20] Qingyue Long, Can Rong, Huandong Wang, and Yong Li. 2025. One Fits All: General Mobility Trajectory Modeling via Masked Conditional Diffusion. *arXiv preprint arXiv:2501.13347* (2025).

[21] Qingyue Long, Yuan Yuan, and Yong Li. 2024. A Universal Model for Human Mobility Prediction. *arXiv preprint arXiv:2412.15294* (2024).

[22] Massimiliano Luca, Gianni Barlacchi, Bruno Lepri, and Luca Pappalardo. 2021. A survey on deep learning for human mobility. *ACM Computing Surveys (CSUR)* 55, 1 (2021), 1–44.

[23] Yingtao Luo, Qiang Liu, and Zhaocheng Liu. 2021. Stan: Spatio-temporal attention network for next location recommendation. In *WWW*. 2177–2185.

[24] Gengchen Mai, Krzysztof Janowicz, Yingjie Hu, Song Gao, Bo Yan, Rui Zhu, Ling Cai, and Ni Lao. 2022. A review of location encoding for GeoAI: methods and applications. *International Journal of Geographical Information Science* 36, 4 (2022), 639–673.

[25] Pierre Nouvellet, Sangeeta Bhatia, Anne Cori, Kylie EC Ainslie, Marc Baguelin, Samir Bhatt, Adhiratha Boonyasiri, Nicholas F Brazeau, Lorenzo Cattarino, Laura V Cooper, et al. 2021. Reduction in mobility and COVID-19 transmission. *Nature communications* 12, 1 (2021), 1–9.

[26] Chenyang Shao, Fengli Xu, Bingbing Fan, Jingtao Ding, Yuan Yuan, Meng Wang, and Yong Li. 2024. Beyond imitation: Generating human mobility from context-aware reasoning with large language models. *arXiv preprint arXiv:2402.09836* (2024).

[27] Zhi Sheng, Daisy Yuan, Jingtao Ding, and Yong Li. 2025. Unveiling the power of noise priors: Enhancing diffusion models for mobile traffic prediction. *arXiv preprint arXiv:2501.13794* (2025).

[28] Junjun Si, Jin Yang, Yang Xiang, Hanqiu Wang, Li Li, Rongqing Zhang, Bo Tu, and Xiangqun Chen. 2023. Trajbert: Bert-based trajectory recovery with spatial-temporal refinement for implicit sparse trajectories. *IEEE Transactions on Mobile Computing* (2023).

[29] Chaoming Song, Tal Koren, Pu Wang, and Albert-László Barabási. 2010. Modelling the scaling properties of human mobility. *Nature physics* 6, 10 (2010), 818–823.

[30] Hongyuan Su, Yu Zheng, Jingtao Ding, Depeng Jin, and Yong Li. 2024. MetroGNN: Metro Network Expansion with Reinforcement Learning. In *Companion Proceedings of the ACM on Web Conference 2024*. 650–653.

[31] Hongyuan Su, Yu Zheng, Yuan Yuan, Yuming Lin, Depeng Jin, and Yong Li. 2025. Reinforcement Learning with Adaptive Reward Modeling for Expensive-to-Evaluate Systems. In *Forty-second International Conference on Machine Learning*.

[32] Hao Sun, Changjie Yang, Liwei Deng, Fan Zhou, Feiteng Huang, and Kai Zheng. 2021. Periodicmove: Shift-aware human mobility recovery with graph neural network. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 1734–1743.

[33] Fernando Terroso-Sáenz and Andrés Muñoz. 2022. Nation-wide human mobility prediction based on graph neural networks. *Applied Intelligence* (2022), 1–17.

[34] A Vaswani. 2017. Attention is all you need. *Advances in Neural Information Processing Systems* (2017).

[35] Ricardo Vilalta and Youssef Drissi. 2002. A perspective view and survey of meta-learning. *Artificial intelligence review* 18 (2002), 77–95.

[36] Anqi Wang, Anshu Zhang, Edwin HW Chan, Wenzhong Shi, Xiaolin Zhou, and Zhewei Liu. 2020. A review of human mobility research based on big data and its implication for smart city development. *ISPRS International Journal of Geo-Information* 10, 1 (2020), 13.

[37] Ruoxi Wang, Bin Fu, Gang Fu, and Mingliang Wang. 2017. Deep & cross network for ad click predictions. In *Proceedings of the ADKDD'17*. 1–7.

[38] Yu Wang, Tongya Zheng, Yuxuan Liang, Shunyu Liu, and Mingli Song. 2024. Cola: Cross-city mobility transformer for human trajectory simulation. In *WWW*. 3509–3520.

[39] Shuai Xu and Donghai Guan. 2024. CrossPred: A Cross-City Mobility Prediction Framework for Long-Distance Travelers via POI Feature Matching. In *CIKM*. 4148–4152.

[40] Song Yang, Jiamou Liu, and Kaiqi Zhao. 2022. GETNext: trajectory flow map enhanced transformer for next POI recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on research and development in information retrieval*. 1144–1153.

[41] Yuan Yuan, Jingtao Ding, Jie Feng, Depeng Jin, and Yong Li. 2024. Unist: A prompt-empowered universal model for urban spatio-temporal prediction. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and*

Data Mining. 4095–4106.

[42] Yuan Yuan, Jingtao Ding, Chonghua Han, Depeng Jin, and Yong Li. 2024. A Foundation Model for Unified Urban Spatio-Temporal Flow Prediction. *arXiv preprint arXiv:2411.12972* (2024).

[43] Yuan Yuan, Jingtao Ding, Depeng Jin, and Yong Li. 2025. Learning the complexity of urban mobility with deep generative network. *PNAS nexus* 4, 5 (2025), pgaf081.

[44] Yuan Yuan, Jingtao Ding, Chenyang Shao, Depeng Jin, and Yong Li. 2023. Spatio-temporal diffusion point processes. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 3173–3184.

[45] Yuan Yuan, Jingtao Ding, Huandong Wang, and Depeng Jin. 2024. Generating Daily Activities with Need Dynamics. *ACM Transactions on Intelligent Systems and Technology* 15, 2 (2024), 1–28.

[46] Yuan Yuan, Jingtao Ding, Huandong Wang, Depeng Jin, and Yong Li. 2022. Activity trajectory generation via modeling spatiotemporal dynamics. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 4752–4762.

[47] Yuan Yuan, Chonghua Han, Jingtao Ding, Depeng Jin, and Yong Li. 2024. Urbandit: A foundation model for open-world urban spatio-temporal learning. *arXiv preprint arXiv:2411.12164* (2024).

[48] Yuan Yuan, Huandong Wang, Jingtao Ding, Depeng Jin, and Yong Li. 2023. Learning to simulate daily activities via modeling dynamic human needs. In *Proceedings of the ACM Web Conference 2023*. 906–916.

[49] Yuan Yuan, Yuheng Zhang, Jingtao Ding, and Yong Li. 2025. WorldMove, a global open data for human mobility. *arXiv preprint arXiv:2504.10506* (2025).

[50] Jun Zhang, Wenxuan Ao, Depeng Jin, Li Liu, and Yong Li. 2023. A City-level High-performance Spatio-temporal Mobility Simulation System. In *Proceedings of the 1st ACM SIGSPATIAL International Workshop on Sustainable Mobility*. 23–32.

[51] Yuheng Zhang, Yuan Yuan, Jingtao Ding, Jian Yuan, and Yong Li. 2024. Noise Matters: Diffusion Model-based Urban Mobility Generation with Collaborative Noise Priors. *arXiv preprint arXiv:2412.05000* (2024).

[52] Vincent Zheng, Bin Cao, Yu Zheng, Xing Xie, and Qiang Yang. 2010. Collaborative filtering meets mobile recommendation: A user-centered approach. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 24. 236–241.

[53] Yu Zheng, Qianyue Hao, Jingwei Wang, Changzheng Gao, Jinwei Chen, Depeng Jin, and Yong Li. 2024. A Survey of Machine Learning for Urban Decision Making: Applications in Planning, Transportation, and Healthcare. *Comput. Surveys* 57, 4 (2024), 1–41.

[54] Yu Zheng, Yuming Lin, Liang Zhao, Tinghai Wu, Depeng Jin, and Yong Li. 2023. Spatial planning of urban communities via deep reinforcement learning. *Nature Computational Science* 3, 9 (2023), 748–762.

[55] Yu Zheng, Hongyuan Su, Jingtao Ding, Depeng Jin, and Yong Li. 2023. Road planning for slums via deep reinforcement learning. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 5695–5706.

[56] Yu Zheng and Xing Xie. 2011. Learning travel recommendations from user-generated GPS traces. *ACM Transactions on Intelligent Systems and Technology (TIST)* 2, 1 (2011), 1–29.

[57] Yuanshao Zhu, Yongchao Ye, Shiyao Zhang, Xiangyu Zhao, and James Yu. 2023. Difftraj: Generating gps trajectory with diffusion probabilistic model. *Advances in Neural Information Processing Systems* 36 (2023), 65168–65188.

[58] Yuanshao Zhu, James Jianqiao Yu, Xiangyu Zhao, Qidong Liu, Yongchao Ye, Wei Chen, Zijian Zhang, Xuetao Wei, and Yuxuan Liang. 2024. Controltraj: Controllable trajectory generation with topology-constrained diffusion model. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 4676–4687.

[59] Yuanshao Zhu, James Jianqiao Yu, Xiangyu Zhao, Xuetao Wei, and Yuxuan Liang. 2024. UniTraj: Universal Human Trajectory Modeling from Billion-Scale Worldwide Traces. *arXiv preprint arXiv:2411.03859* (2024).