Breaking Data Silos: Towards Open and Scalable Mobility Foundation Models via Generative Continual Learning

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

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

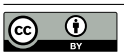
Foundation models have revolutionized fields such as natural language processing and computer vision by enabling general-purpose learning across diverse tasks and datasets. However, building analogous models for human mobility remains challenging due to the privacy-sensitive nature of mobility data and the resulting data silos across institutions. To bridge this gap, we propose MoveGCL, a scalable and privacy-preserving framework for training mobility foundation models via generative continual learning. Without sharing raw data, MoveGCL enables decentralized and progressive model evolution by replaying synthetic trajectories generated from a frozen teacher model, and reinforces knowledge retention through a tailored distillation strategy that mitigates catastrophic forgetting. To address the heterogeneity of mobility patterns, MoveGCL incorporates a Mixture-of-Experts Transformer with a mobility-aware expert routing mechanism, and employs a layer-wise progressive adaptation strategy to stabilize continual updates. Experiments on six real-world urban datasets demonstrate that MoveGCL achieves performance comparable to joint training and significantly outperforms federated learning baselines, while offering strong privacy protection. MoveGCL marks a crucial step toward unlocking foundation models for mobility, offering a practical blueprint for open, scalable, and privacy-preserving model development in the era of foundation models. To facilitate reproducibility and future research, we have released the code and models at https://github.com/tsinghua-fib-lab/MoveGCL.

CCS Concepts

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

Keywords

Human mobility, foundation models, continual learning



SIGSPATIAL '25, Minneapolis, MN, USA

ACM ISBN 979-8-4007-2086-4/2025/11
https://doi.org/10.1145/3748636.3762728

ACM Reference Format:
Yuan Yuan*, Yukun Liu*, Chonghua Han, Jie Feng, and Yong Li†. 2025. Breaking Data Silos: Towards Open and Scalable Mobility Foundation Models via Generative Continual Learning. In *The 33rd ACM International Conference on Advances inGeographic Information Systems (SIGSPATIAL '25), November 3–6, 2025, Minneapolis, MN, USA.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3748636.3762728

1 Introduction

In natural language processing (NLP) [4, 5, 19] and computer vision (CV) [9, 29], the rise of large pre-trained foundation models (e.g., GPT, Sora) [35] has significantly advanced model sharing and general-purpose intelligence through centralized training and open release. However, in the critical domain of human mobility modeling [3, 32], the era of large foundation models has not yet arrived. This is primarily due to the highly sensitive nature of mobility trajectory data, which involves personal privacy [21, 52, 54]. Such data cannot be easily shared or jointly trained across institutions due to privacy constraints and legal restrictions. As a result, mobility datasets often exist in isolated silos [56, 62], with different organizations and researchers relying on their own private datasets, hindering data integration, benchmarking, and collaborative model development.

To address the challenge of data, recent studies have proposed different training strategies to leverage multiple datasets, and Figure 1 compares different approaches. The most common strategy is to train models separately, as shown in Figure 1(a). Recent efforts such as UniTraj [63], TrajBert [40], and TrajFM [26] have explored joint training for unified representation and cross-city generalization (Figure 1(b)), but these models remain tightly coupled with proprietary or limited-quality datasets. TrajFM and TrajBert are typically pre-trained on restricted or private data. While UniTraj uses public data, it suffers from low semantic richness and high sparsity. Federated learning [12, 30] offers a potential solution for distributed mobility model training (Figure 1(c)), but its reliance on frequent synchronization and communication poses challenges for scalability and practical deployment. Consequently, these approaches

*Equal contribution.
†Corresponding author.

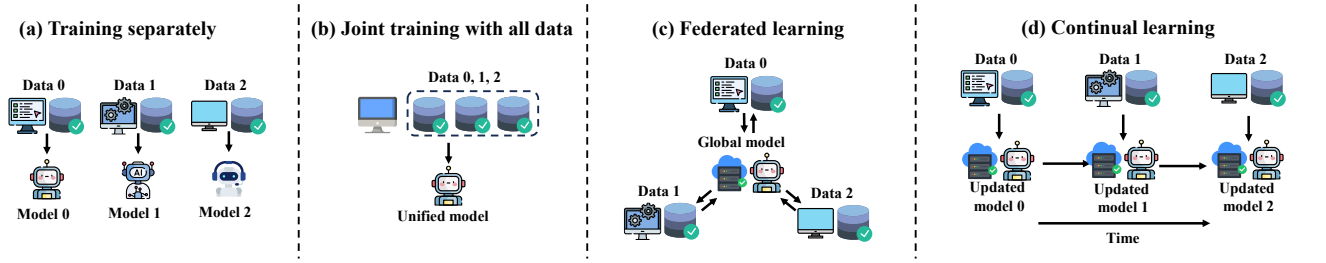


Figure 1: Method comparison with multiple mobility datasets.

fall short of meeting the diverse, dynamic, and multi-source demands of real-world mobility modeling, and cannot support an open ecosystem of shared models as seen in NLP and CV domains.

To truly usher in a new era of shareable and sustainable human mobility foundation models, we argue for a new collaborative paradigm. This paradigm enables multiple data holders to jointly evolve and continually build a foundation model without sharing raw data, while preserving both privacy and generalization capability. However, this vision poses several critical challenges: (1) Privacy constraints. The sensitive nature of mobility trajectory data prohibits direct data sharing. Designing a training framework that supports collaboration without exposing raw data is a fundamental yet unresolved challenge. (2) Catastrophic forgetting. Without access to past training data, the model can easily forget previously learned knowledge when updated with new mobility data, hindering long-term model evolution. (3) Data heterogeneity. Mobility data exhibits substantial variation across regions, populations, and data sources. A practical model must generalize well and dynamically adapt to such heterogeneity.

We propose MoveGCL, a scalable training framework for mobility foundation models based on generative continual learning. MoveGCL allows each data holder to evolve a shared model locally without exposing raw data, thereby ensuring full privacy preservation. Specifically, MoveGCL starts from a pre-trained base model and employs a synthetic trajectory replay mechanism: instead of accessing historical data, each participant generates synthetic trajectories that approximate previously seen mobility patterns. This replay process preserves prior knowledge and mitigates catastrophic forgetting. Furthermore, knowledge distillation is applied during replay to reinforce the model's ability to retain past capabilities while adapting to new data. To handle the diversity of mobility data, MoveGCL adopts a Mixture-of-Experts (MoE) architecture equipped with a mobility pattern-aware expert routing mechanism. This design enables the model to dynamically select expert modules tailored to local mobility characteristics. Together, these innovations make MoveGCL a practical and privacy-preserving solution for collaboratively building generalizable mobility foundation models across distributed, heterogeneous, and privacy-sensitive data sources. In summary, our key contributions are as follows:

- We are the first to formalize a privacy-preserving collaborative training paradigm towards mobility foundation models, enabling decentralized model evolution without raw data sharing.
- We propose MoveGCL, a novel framework based on generative continual learning. Its core components—knowledge distillation, mobility-aware expert routing, and layer-wise progressive adaptation—enable privacy-preserving, scalable, and adaptive trajectory model training across diverse data sources.
- MoveGCL achieves performance comparable to joint training with full data access, and significantly outperforms federated learning baselines without data sharing.

2 Related Work

2.1 Mobility Data

Mobility data includes aggregated flows and individual trajectories [3, 52, 59]. Aggregated flows are relatively easier to obtain and have been widely used in urban analytics [36, 57]. However, individual-level mobility data remain fragmented due to privacy concerns and institutional data silos [21, 52]. Real-world trajectory datasets, such as GeoLife [60], T-Drive [49], NYC Taxi [34], and Foursquare [47], often have limited city coverage, short time spans, and sparse sampling. Some global-scale open datasets, such as the one used in UniTraj, have been introduced, but they suffer from low spatial-temporal resolution and inconsistent data quality. With the advancement of generative AI, synthetic mobility datasets have emerged, such as SynMob [62], YJMob100K [46] and WorldMove [56]. However, the quality of synthetic data still falls short compared to real-world trajectories, particularly in terms of behavioral diversity, temporal continuity, and semantic consistency. In practice, access to real trajectory datasets typically requires signing NDAs, and most published studies do not release the mobility datasets they use due to privacy and legal restrictions [37].

2.2 Mobility Foundation Models

Inspired by the success of foundation models in NLP and CV, recent efforts have explored pre-trained models for urban and mobility domains [6, 7, 15, 55, 58, 61]. Early studies primarily focused on aggregated mobility data, leveraging mobility flows across cities to build unified spatio-temporal representations, and have demonstrated strong zero-shot transfer capabilities [23, 24, 50, 51]. In contrast, individual-level mobility foundation models are less developed. Researchers have explored multi-scale mobility modeling [31, 52, 59], aiming to capture both micro-level behaviors and macro-level patterns essential for generalization. Attempts such as UniTraj [63], TrajBert [40], and TrajFM [26] have explored learning from open trajectory datasets, but these datasets often consist of short-term or non-representative mobility traces that do not reflect regular human movement patterns. As a result, current models struggle to

capture the full complexity and diversity of real-world individual mobility. Recently, LLMs have also widely utilized in generating human mobility [10, 14, 16, 17, 38], but the gap between natural language and trajectory data suggests that mobility still requires native foundation models, which can later be aligned with LLMs to bridge symbolic reasoning and physical behavior modeling.

2.3 Continual Learning

Continual learning [18, 42, 42], also known as lifelong learning, aims to enable models to learn from a sequence of tasks or data streams without forgetting previously acquired knowledge [48]. A central challenge in continual learning is catastrophic forgetting [22, 44], where the model's performance on earlier tasks degrades as it learns new ones. Continual learning methods are typically categorized into three main types. Regularization-based methods introduce constraints on parameter updates to preserve important knowledge from earlier tasks, as seen in approaches like Elastic Weight Consolidation (EWC) and Synaptic Intelligence (SI). Replay-based methods mitigate forgetting by either storing a subset of previous data (experience replay) or generating pseudo-data (generative replay) to simulate past learning. Parameter isolation methods, on the other hand, allocate different parts of the model to different tasks, using techniques such as dynamic networks or task-specific masking to reduce interference between tasks.

3 Preliminaries

3.1 Data Format

In our setting, human mobility data is represented as sequences of spatiotemporal tokens, where each token corresponds to a visited location at a specific time. The spatial domain is typically discretized into a uniform grid (500m × 500m resolution), and the temporal domain is segmented into fixed-length intervals (30 minutes). Each individual trajectory can be formulated as a sequence $z_1, z_2, \ldots, z_T$, where $z_t = (l_t, t_t)$ denotes the location and timestamp of a mobility event at time step t. These sequences capture rich behavioral patterns across time and space and form the foundation for model training.

3.2 Model Training via Next-Token Prediction

Following the standard practice in language modeling, the training objective for mobility foundation models is formulated as a next-location prediction task. Given a partial trajectory $\{z_1, \ldots, z_{t-1}\}$, the model aims to predict the next location l_t, where l_t represents the spatial component of the upcoming step in the trajectory. Formally, the training objective is defined as maximizing the log-likelihood of the observed sequence:

$$\mathcal{L} = \sum_{t=1}^{T} \log P(l_t \mid z_1, \ldots, z_{t-1}; \theta), \tag{1}$$

where θ denotes the model parameters. This objective allows the model to learn rich dependencies across spatial locations, temporal patterns, and contextual semantics, and serves as the core pretraining strategy for mobility foundation models.

3.3 Training Pipeline for Mobility Foundation Model Development

We adopt a continual learning paradigm to train the mobility foundation model. To simulate learning on highly heterogeneous data, each round of continual learning introduces the dataset of a new city to the model. Initially, the base model f_{base} is trained using the base dataset $\mathcal{D}_{\text{base}}$. During continual learning, at the beginning of the n-th round, the model has already been trained on the dataset $\mathcal{D}_{\text{all},n-1} = \mathcal{D}_{\text{base}} \cup \mathcal{D}_{\text{continual},n-1}$, where $\mathcal{D}_{\text{continual},n-1} = \bigcup_{i=1}^{n-1} d_i$, and each d_i represents the dataset introduced in the i-th round. However, the historical dataset $\mathcal{D}_{\text{all},n-1}$ is no longer accessible. The model update at round n is performed solely based on the current data d_n and a copy of the model from the previous round, denoted as $f_{\text{old},n}$.The continual learning process can be formalized as the following optimization objective:

$$\theta_{\text{new},n} = \arg\min_{\theta} \mathcal{L}\left(\theta;\, d_n;\, f_{\text{old},n}\right), \quad \text{s.t.} \quad \theta \leftarrow \theta_{\text{old},n}, \tag{2}$$

where θ denotes the model parameters, and $\theta_{\text{old},n}$ and $\theta_{\text{new},n}$ correspond to the parameters of $f_{\text{old},n}$ and $f_{\text{new},n}$, respectively. The loss function$\mathcal{L}(\theta; d_n; f_{\text{old},n})$ incorporates constraint terms derived from the previous model $f_{\text{old},n}$ to mitigate catastrophic forgetting. Rather than reinitializing from scratch, we optimize $\theta_{\text{new},n}$ starting from $\theta_{\text{old},n}$. Specifically, when $n = 0$, we set $f_{\text{old},n} = f_{\text{base}}$; for $n > 0$, we have $f_{\text{old},n} = f_{\text{new},\,n-1}$.

4 MoveGCL

In this section, we introduce the overall framework of MoveGCL, which is shown in Figure 2. MoveGCL is built upon the paradigm of generative continual learning, as detailed in Section 4.1. We then present the core model architecture in Section 4.2, which is designed to support modular scalability and cross-city adaptability. Finally, we elaborate on the training strategy in Section 4.3, including the design of layer-wise progressive adaptation mechanism to stabilize continual updates and mitigate forgetting.

4.1 Generative Continual Learning

Generative Replay with Teacher Model. To retain knowledge from previously visited cities without storing real-world mobility trajectory data, we design a generative replay strategy, as illustrated in Figure 2(a). At each stage, we keep a copy of the previously trained model f_{old}, referred to as the *teacher model*. This teacher model represents the model trained on earlier mobility datasets. It remains frozen during subsequent learning and serves as a knowledge source to guide the student model f_{new} when learning new cities.

To simulate past mobility behaviors, we employ f_{old} to generate synthetic trajectory data $\tilde{x}_{\text{old}}^{c_i}$. First, we extract a trajectory

$$x_{\text{new}} = \left[(l'_0,\, t'_0),\, (l'_1,\, t'_1),\, \ldots,\, (l'_L,\, t'_L)\right], \tag{3}$$

from the new dataset, where L denotes its length and $\{(l'_i,\, t'_i)\}_{i=0}^{L}$ are the location–time pairs. Next, for a specific previously observed city c_i, we sample an initial location from the empirical distribution of actual locations in city c_i conditioned on length L:

$$l_0 \;\sim\; \rho_{\text{loc}|L}^{c_i}, \tag{4}$$

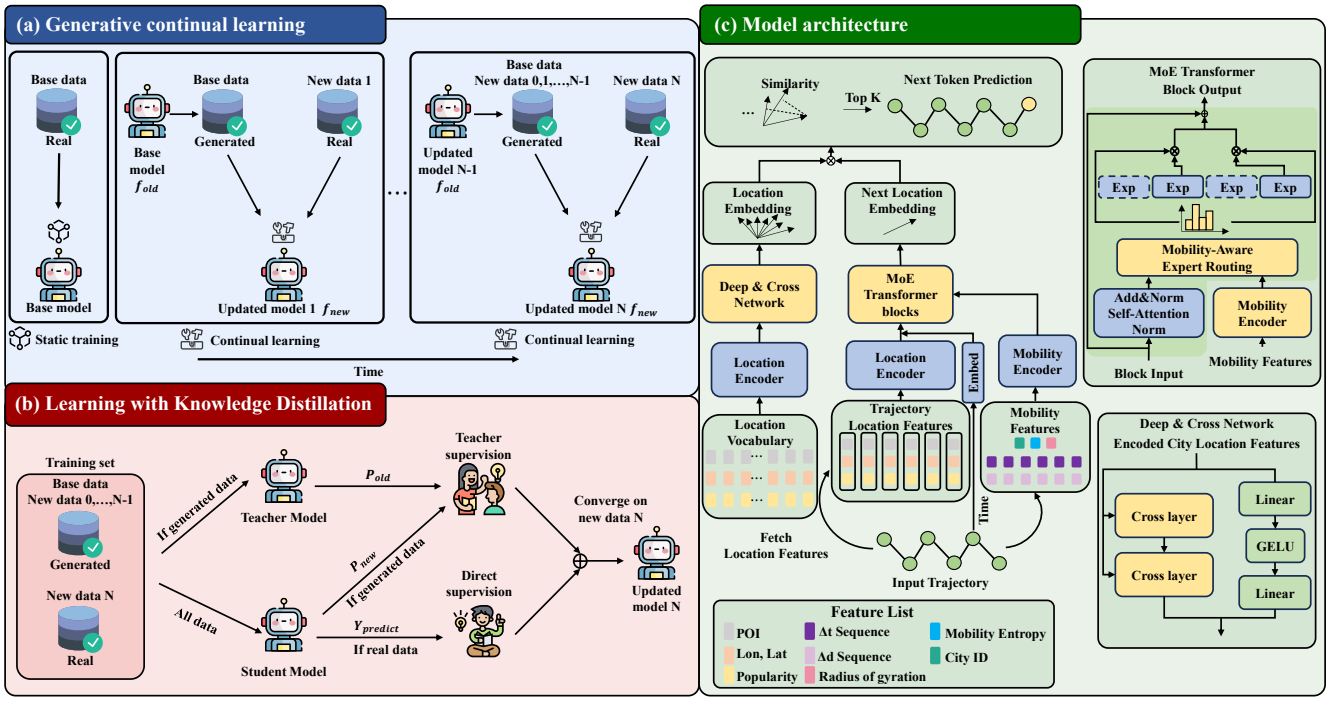


Figure 2: Overview of the MoveGCL framework: (a) the overall workflow; (b) the implementation of generative continual learning; (c) the model architecture.

where $\rho^{c_i}_{\text{loc}|L}$ denotes the empirical distribution of initial locations in previously observed city c_i given a trajectory length L. We then replace l'_0 with the sampled l_0 and generate the pseudo old-city trajectory

$$\tilde{x}^{c_i}_{\text{old}} = \left[(l_0, t'_0), (l_1, t'_1), \ldots, (l_L, t'_L)\right],$$

by drawing subsequent locations autoregressively:

$$(l_1, l_2, \ldots, l_L) \sim f_{\text{old}}\Big(\cdot \mid l_0, \{t'_i\}_{i=0}^{L}\Big), \tag{5}$$

where each t'_i is taken from the time distribution of the extracted new-data trajectory.

These pseudo-trajectories reflect the mobility patterns learned in earlier stages. We combine all pseudo-trajectories with the real data from the current city, to construct the full training set:

$$\mathcal{D}_{\text{train}} = \bigcup_{i=1}^{N} \alpha\, \tilde{\mathcal{X}}^{c_i}_{\text{old}} \cup \mathcal{X}_{\text{new}}, \tag{6}$$

where $\{c_1, c_2, \ldots, c_N\}$ denotes the N previously observed cities, $\tilde{\mathcal{X}}^{c_i}_{\text{old}}$ denotes the set of pseudo-trajectories generated for city c_i, and $\mathcal{X}_{\text{new}}$ is the set of real trajectories from the current city. Coefficient $\alpha > 0$ specifies ratio between the number of pseudo-trajectories in each previous city and the number of real trajectories in $\mathcal{X}_{\text{new}}$.

This allows the student model to learn current-city behaviors while retaining knowledge of previously learned cities.

Distilling Knowledge to Preserve Mobility Patterns. To further strengthen the model's ability to preserve prior knowledge, we introduce a knowledge distillation loss that transfers behavioral patterns from the teacher model to the student model, as depicted in Figure 2(b). For each generated pseudo-trajectory $\tilde{x}_{\text{old}}$, we extract the predicted mobility distributions from both models:

$$P_{\text{old}}(\cdot \mid \tilde{x}_{\text{old}}) = f_{\text{old}}(\tilde{x}_{\text{old}}), \quad P_{\text{new}}(\cdot \mid \tilde{x}_{\text{old}}) = f_{\text{new}}(\tilde{x}_{\text{old}}). \tag{7}$$

We minimize the Kullback–Leibler (KL) divergence between the teacher's and student's predicted distributions:

$$\mathcal{L}_{\text{KD}} = \mathbb{E}_{\tilde{x}_{\text{old}} \sim f_{\text{old}}}\left[\text{KL}\left(P_{\text{old}}(\cdot \mid \tilde{x}_{\text{old}}) \,\|\, P_{\text{new}}(\cdot \mid \tilde{x}_{\text{old}})\right)\right]. \tag{8}$$

For the new data x_{new}, we compute the task loss as the cross-entropy between the model's predicted distribution and the true labels, referred to as $\mathcal{L}_{\text{cross-entropy}}$. The total training objective of the student model is a weighted sum of the task loss on new-city data and the distillation loss:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{cross-entropy}} + \lambda \cdot \mathcal{L}_{\text{KD}}, \tag{9}$$

where λ is a hyperparameter that balances learning new knowledge and retaining previously acquired behaviors.

4.2 Model Architecture

To enable scalable and adaptive learning across heterogeneous urban environments, our model is designed with a modular architecture that integrates flexible location encoders/decoders, a Mixture-of-Experts (MoE) Transformer backbone, and a mobility-aware expert routing mechanism, as shown in Figure 2(c). This design ensures the model's scalability and adaptability in multi-city continual learning scenarios.

Unified Location Encoder. Conventional location representations often rely on discrete location IDs, which are inherently city-specific and hinder cross-city generalization. To overcome this limitation, we design a continuous location representation that embeds each location into a shared latent space, capturing transferable semantic and spatial properties across cities. This unified representation facilitates knowledge sharing and supports incremental learning across heterogeneous urban environments. Concretely, each location $l \in \mathcal{L}$ is represented by a feature vector $\mathbf{z}_l \in \mathbb{R}^d$ constructed from three key components:

$$\mathbf{z}_l = \phi_{\text{POI}}(l) \oplus \phi_{\text{lat-lon}}(l) \oplus \phi_{\text{hot}}(l), \tag{10}$$

where $\phi_{\text{POI}}(l) \in \mathbb{R}^{d_1}$ denotes the Point-of-Interest (POI) embedding, capturing semantic land-use attributes such as residential, commercial, educational, or recreational functions, often indicative of mobility intent and purpose; $\phi_{\text{hot}}(l) \in \mathbb{R}^{d_2}$ is the mobility heat embedding, derived from public available OD flows at each location, which reflects the functional centrality of that location; $\phi_{\text{lat-lon}}(l) \in \mathbb{R}^{d_3}$ is the normalized latitude-longitude embedding, representing the relative spatial position of the location within the city boundary.

The overall location representation $\mathbf{z}_l$ is obtained via concatenation ($\oplus$) of these features, and is further transformed by a shared multi-layer perceptron (MLP). This design is sufficient and generalizable because it captures three complementary views of spatial semantics: (1) semantic functions via POI types, (2) actual mobility signals via visitation popularity, and (3) captures where the location sits in the urban layout. Together, they provide a compact yet expressive embedding that generalizes well across different cities with varied spatial structures.

Mixture-of-Experts Transformer. The Mixture-of-Experts (MoE) architecture comprises a router network and multiple expert networks, serving as a replacement for the Feed-Forward Network (FFN) within the Transformer [33]. The output of the MoE layer, $F_{\text{MoE}}(x)$, is the weighted sum of the selected expert outputs, where the weights are given by the router network's output:

$$F_{\text{MoE}}(x) \;=\; \sum_{i=1}^{k} R_i(x) \;\cdot\; E_i(x). \tag{11}$$

Here, x denotes the input to the MoE module, k is the number of selected experts, $R_i(x)$ is the output of the router network for expert i (detailed in Section 4.2), $E_i(x)$ is the output of expert i.

MoveGCL is built upon Mixture-of-Experts (MoE) Transformer blocks, in which each expert module is responsible for capturing specific mobility patterns. During continual learning, we introduce new experts to accommodate knowledge from new cities, and design layer-wise progressive adaptation training strategy (detail in Section 4.3). This partial parameter update strategy injects new knowledge without overwriting existing capabilities, thus alleviating catastrophic forgetting. The modularity of the MoE block also supports elastic expansion of the model as more cities are introduced.

Mobility-Aware Expert Routing. For each input trajectory x, we extract a set of mobility behavior descriptive features and encode them into a mobility feature descriptor vector $\mathbf{z}_m \in \mathbb{R}^d$. This feature set comprises: the jump distance d_{jump} (distance between the current point and the previous point in the trajectory); the waiting time t_{wait} (time difference between arrivals at the current and previous points in the trajectory); the quantized radius of gyration r_{gyr}; the quantized location entropy H_{loc}; and the city identifier ID_{city}. These features are embedded via their respective encoders, where d_{jump} and t_{wait} are processed by a Transformer-based continuous feature encoder, and r_{gyr}, H_{loc}, and ID_{city} are handled by discrete embedding modules. Finally, the five feature embeddings are concatenated to form the mobility behavior vector:

$$\mathbf{z}_m \;=\; \begin{bmatrix} \phi_{d_{\text{jump}}}\big(d_{\text{jump}}(x)\big),\ \phi_{t_{\text{wait}}}\big(t_{\text{wait}}(x)\big),\ \phi_{r_{\text{gyr}}}\big(r_{\text{gyr}}(x)\big), \\ \phi_{H_{\text{loc}}}\big(H_{\text{loc}}(x)\big),\ \phi_{\text{ID}_{\text{city}}}\big(\text{ID}_{\text{city}}\big) \end{bmatrix}. \tag{12}$$

Here, $\phi_{d_{\text{jump}}}$ and $\phi_{t_{\text{wait}}}$ denote the Transformer encoders for continuous mobility features, while $\phi_{r_{\text{gyr}}}$, $\phi_{H_{\text{loc}}}$, and $\phi_{\text{ID}_{\text{city}}}$ represent the embedding encoders for discrete features.

Within each layer of the MoE transformer blocks, we introduce a routing network based on a linear transformation to compute the routing weights for each expert, leveraging both the mobility descriptor feature vector $\mathbf{z}_m$ and the output of the self-attention submodule at that layer. The routing weights for layer i are computed as:

$$R_i(x) = \text{softmax}\Big(\text{TopK}\big(W_{i,r}\,(\mathbf{z}_m \oplus X_i(x)) \;+\; b_i\big)\Big), \tag{13}$$

where $\mathbf{z}_m$ is the mobility feature descriptor vector defined above; $X_i(x)$ denotes the output of the self-attention submodule; $W_{i,r}$ is a learnable projection matrice; b_i is the bias term; and $\text{TopK}(\cdot)$ retains only the top K values (corresponding to the K highest-scoring experts), setting the remaining expert scores to $-\infty$ so that their weights after the softmax operation are effectively zero.

This routing strategy achieves two objectives: (1) it promotes functional specialization by directing similar motion patterns to consistent expert subsets; (2) it enables the model to discover and transfer shared mobility structures across cities, thereby enhancing generalization in multi-city scenarios. Additionally, this mobility-aware routing provides a structured inductive bias that accelerates model adaptation during incremental learning, allowing new experts to specialize rapidly with minimal interference to retained knowledge.

Similarity–Based Decoder. The next-location prediction is performed by computing the similarity between the final output of the Mixture-of-Experts (MoE) Transformer blocks and the representation vectors of all candidate locations in the city. These location representations are generated using a Deep & Cross Network (DCN) [43], which consists of both a Cross layer and a Deep layer to capture feature interactions and nonlinear transformations.

The Cross network captures inter-location correlations within a city by applying element-wise interactions over location embeddings E_l, producing:

$$E_{\text{cross}} = \sum_{i=1}^{d} E_i \odot W_i E_l + b_i, \tag{14}$$

where $\odot$ denotes element-wise multiplication, and W_i, b_i are learnable parameters. The Deep layer refines the same E_l using a two-layer MLP:

$$E_{\text{deep}} = \text{GELU}\left(\left(W_1 E_l + b_1\right) W_2 + b_2\right), \tag{15}$$

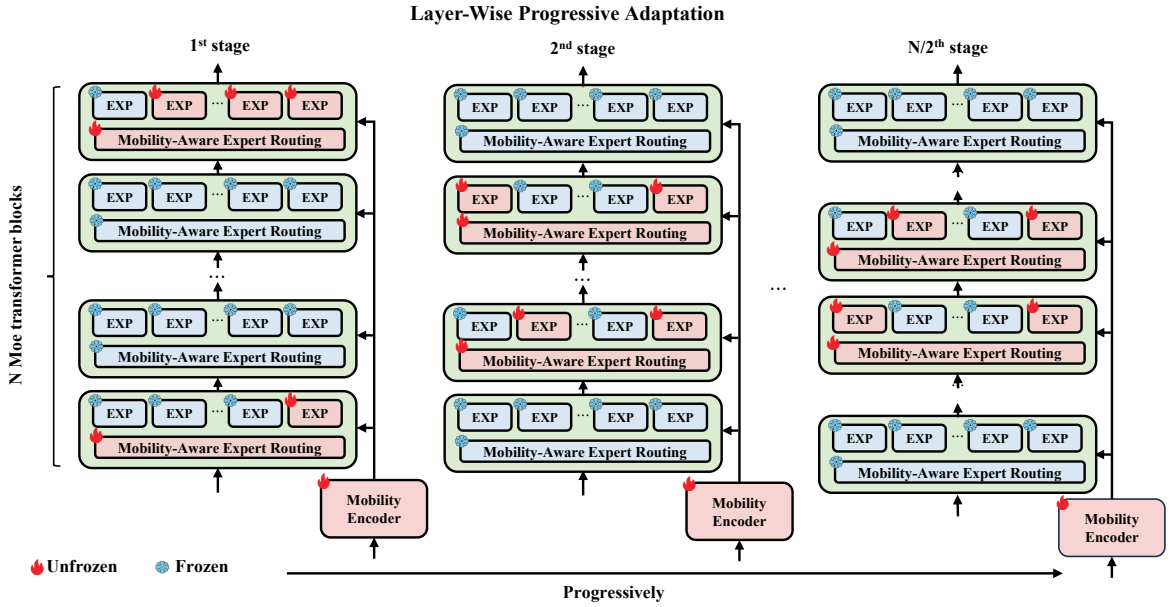


Figure 3: Illustration of the layer-wise progressive adaptation

with weights W_1, W_2, biases b_1, b_2, and GELU activation. The DCN output is the concatenation of both branches:

$$E_{\text{DCN}} = E_{\text{cross}} \oplus E_{\text{deep}}. \tag{16}$$

For next-location prediction, the user's historical trajectory is encoded by MoE transformer blocks to produce a prediction vector P; separately, the Deep & Cross Network (DCN) encodes every candidate location in the city to produce location representations E_{DCN}. The similarity score is computed as:

$$\text{Score}_{\text{similarity}} = \sum_{i=1}^{d} P \cdot E_{\text{DCN},i}, \tag{17}$$

where d is the number of locations. Higher similarity scores indicate higher probabilities of being the next location. This similarity-based decoding strategy ensures scalability, as it decouples the prediction process from a fixed output space. Instead of classifying over a static set of locations, the model performs representation-level matching, allowing it to generalize across cities with different spatial layouts and dynamically varying numbers of candidate locations.

4.3 Layer-Wise Progressive Adaptation

To ensure a balance between previously and newly learned knowledge, MoveGCL employs a layer-wise progressive adaptation strategy, where model parameters are updated in stages, as illustrated in Figure 3. For a model composed of N layers of MoE transformer blocks, the total number of training epochs E is evenly divided into $N/2$ stages, with each stage lasting $\frac{\text{E}}{N/2}$ epochs. At each stage, a pair of symmetrically positioned MoE transformer blocks—one near the input side and the other near the output side—are unfrozen for fine-tuning, while the remaining layers remain frozen. The specific process is as follows:

- In Stage 1, the outermost layers (closest to the input and output) are unfrozen.
- In Stage 2, the second closest layers to the input and output are unfrozen.
- ...
- In Stage N/2, the two central layers of the model are unfrozen.

Within each stage, only a subset of parameters in the unfrozen layers is updated—the routing modules, newly added experts and previously trained experts that were not frequently activated in the prior generative continual learning phase. To facilitate adaptation to the mobility features across different datasets, parameters of the mobility feature encoder are updated continuously during all stages. Furthermore, to prevent large parameter shifts during the initial stage of parameter updating, all previously trained experts in the input-side MoE transformer layer are kept frozen during Stage 1.

5 Results

5.1 Experimental Settings

Datasets. We utilize human mobility datasets from multiple cities to evaluate the performance of MoveGCL. Specifically, the datasets cover over eight hundred thousand users and feature a relatively high sampling rate compared to currently available public datasets. Detailed statistics and descriptions are provided in Appendix Table 4. For each city, we randomly sample 120,000 trajectories for training, 40,000 for validation, and 40,000 for testing.

Baselines. We compare our method with a diverse set of baselines, including traditional mobility models, federated learning-based approaches, and joint learning models with privacy-preserving mechanisms.

Table 1: Performance comparison between MoveGCL and baseline methods across different cities and Acc@k metrics.

Model	Atlanta		Chicago		Los Angeles		New York		Seattle		Washington D.C	
	Acc@1	Acc@3	Acc@1	Acc@3	Acc@1	Acc@3	Acc@1	Acc@3	Acc@1	Acc@3	Acc@1	Acc@3
Markov	0.183	0.325	0.146	0.260	0.103	0.201	0.115	0.275	0.202	0.318	0.162	0.347
LSTM	0.231	0.373	0.194	0.334	0.131	0.275	0.169	0.312	0.259	0.420	0224	0.383
Transformer	0.210	0.353	0.175	0.300	0.124	0.268	0.156	0.318	0.235	0.397	0.192	0.356
DeepMove	0.242	0.393	0.203	0.344	0.147	0.274	0.177	0.329	0.278	0.444	0.247	0.408
TrajBert	0.214	0.370	0.183	0.316	0.146	0.277	0.159	0.310	0.234	0.402	0.207	0.367
CLET	0.263	0.422	0.200	0.341	0.138	0.275	0.156	0.313	0.289	0.454	0.232	0.390
PMF	0.248	0.381	0.151	0.249	0.112	0.190	0.150	0.257	0.269	0.418	0.217	0.349
LightTR	0.269	0.402	0.168	0.269	0.130	0.216	0.168	0.281	0.296	0.444	0.242	0.376
MoveGCL (FullTune)	0.188	0.304	0.125	0.206	0.064	0.114	0.208	0.329	0.199	0.318	0.147	0.257
MoveGCL (ExpertTune)	0.192	0.310	0.132	0.215	0.062	0.108	0.207	0.327	0.195	0.322	0.147	0.259
MoveGCL (WSC→A→L→N)	0.282	0.421	0.197	0.306	0.157	0.254	0.206	0.328	0.324	0.478	0.273	0.413

- **Traditional approach:** This includes Markov models [13] that fit separate transition matrices for different datasets.
- **Deep mobility models:** We include LSTM [20], Transformer [41], DeepMove [11], TrajBert [40], and CLET [25] as representative baselines. For each dataset, we train a separate model to ensure fair comparison under the same training conditions.
- **Fedarated learning models:** We evaluate against PMF [12] and LightTR [30], which leverage federated learning frameworks for human mobility prediction while maintaining data decentralization. We apply their federated learning methods to train our model.
- **Joint models with privacy protection:** These methods enable continual learning without accessing previously seen data. Specifically, we consider two variants of our model: **MoveGCL (FullTune)** unfreezes all experts and routers in the MoE Transformer while keeping the rest of the model frozen, and fine-tunes using only the new city's data. **MoveGCL (ExpertTune)** incrementally adds one new expert per layer in the MoE Transformer, unfreezes all experts and routers, and fine-tunes on the new city's data while keeping all other parameters fixed.

Parameter Settings. The key parameters of our framework fall into three main categories.

- For the model architecture, we set the temporal embedding dimension to 48. In the mobility encoder, the embedding dimensions of d_{jump} and t_{wait} are both 128, the embedding dimensions of r_{gyr} and location entropy H_{loc} are 64, and the embedding dimension of ID_{city} is 32, with the self-attention modules for both d_{jump} and t_{wait} using a hidden dimension of 64. In the trajectory location encoder and the city location encoder, the embedding dimensions of ϕ_{POI}, ϕ_{hot}, and $\phi_{\text{lat-lon}}$ are 256, 128, 128, respectively. The hidden dimension of each MoE transformer block is set to 512. The initial number of experts in each MoE transformer block is 4, and the model comprises 6 layers of MoE transformer blocks.
- For the base model training phase, the initial model is obtained by training on three cities' datasets. During this phase, we use a batch size of 16 and train for 30 epochs. The initial learning rate is set to 1.2×10^{-5}, and the learning rate decays in a stepwise fashion during training.
- For generative continual learning, whenever we introduce a new dataset (i.e., a new city), we add one new expert to every MoE transformer block. The initial learning rate for this phase is 1.2×10^{-4}, the batch size is 128, and training also runs for 30 epochs, with the learning rate decaying stepwise throughout. The generative coefficient α is set to 20%. The balance coefficient λ for $\mathcal{L}_{\text{total}}$ is set to 1.

5.2 Overall Performance

Table 1 presents the performance of MoveGCL compared to state-of-the-art baseline methods.

- **MoveGCL consistently outperforms traditional deep learning models trained independently on each dataset, demonstrating strong cross-city scalability.** On average, it achieves a 8% improvement in Acc@1, highlighting its ability to generalize across diverse urban environments. This result validates the promise of mobility foundation models, which unify knowledge across cities and reduce redundancy. In contrast, training separate models for each city not only increases computational and deployment costs, but also fails to leverage shared mobility patterns across domains.
- **MoveGCL surpasses privacy-preserving federated learning approaches in both accuracy and stability.** Compared to domain-specific baselines such as PVM [12] and LightTR [30], MoveGCL achieves significantly higher performance. This advantage stems from its unified generative continual learning framework, which maintains global generalization without suffering from the synchronization overhead and convergence instability inherent in federated setups. This further supports its practicality in real-world multi-party mobility modeling scenarios.
- **MoveGCL effectively balances adaptation to new data while retaining knowledge from previously seen data.** We benchmark against two continual learning strategies—FullTune and ExpertTune—which either fine-tune or expand the model on new

Table 2: Performance comparison across cities and Acc@k metrics for evaluating order invariance.

Model	Atlanta		Chicago		Los Angeles		New York		Seattle		Washington D.C	
	Acc@1	Acc@3	Acc@1	Acc@3	Acc@1	Acc@3	Acc@1	Acc@3	Acc@1	Acc@3	Acc@1	Acc@3
WSC→A→L→N	0.282	0.421	0.197	0.306	0.157	0.254	0.206	0.328	0.324	0.478	0.273	0.413
AWN→L→S→C	0.284	0.423	0.197	0.308	0.150	0.246	0.200	0.326	0.317	0.469	0.265	0.407
WAL→N→S→C	0.285	0.426	0.194	0.304	0.151	0.245	0.188	0.306	0.321	0.472	0.267	0.407
Joint Training	0.288	0.428	0.192	0.302	0.156	0.250	0.198	0.320	0.322	0.475	0.270	0.410

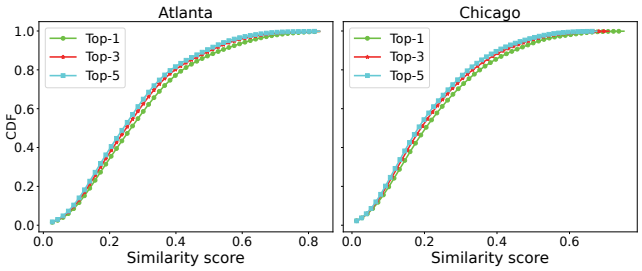


Figure 4: Similarity score distribution in uniqueness testing.

datasets. While these methods partially adapt to new cities, they suffer from severe performance degradation on previously seen data, indicating catastrophic forgetting. In contrast, MoveGCL preserves prior knowledge and achieves higher performance on both old and new datasets, demonstrating its ability to support continuous model evolution without sacrificing stability.

5.3 Order Invariance in Continual Learning

To assess the robustness of MoveGCL in real-world deployment scenarios, we investigate its sensitivity to the order in which data from different cities is introduced during continual learning. In practice, the arrival of mobility data is often dictated by external factors such as data access regulations, infrastructure development cycles, or institutional collaborations. As a result, foundation models intended for long-term, large-scale deployment must remain robust to such variations in data sequencing.

We simulate this scenario by reversing the order of datasets used in the continual learning phase. As shown in Table 2, the performance of MoveGCL remains remarkably consistent, with the vast majority of metrics showing deviations of less than 5% across original and reversed sequences. This empirical finding confirms the order-invariant learning behavior of our model, demonstrating that MoveGCL can integrate new city data without disrupting previously acquired knowledge, even when the order of exposure varies significantly. This property is especially crucial for building scalable and unified mobility foundation models, which must support progressive, privacy-preserving knowledge accumulation in non-i.i.d. settings where data arrives incrementally and asynchronously. Order robustness is thus a key enabler for deploying foundation models that can continually evolve while ensuring stable performance and generalization across diverse urban contexts.

5.4 Privacy Evaluation

As MoveGCL is built on a generative continual learning framework that does not retain raw data from previous cities, a key question is whether the synthetic data used for replay may inadvertently leak private information from the original training data. To rigorously evaluate the privacy-preserving properties of our approach, following the methodology in [52, 53], we conduct a comprehensive analysis from three complementary perspectives:

- **Uniqueness Testing** [8, 45]: To evaluate the degree of similarity between the generated data and the real data.
- **Membership Inference Attack** [27, 39]: Given a trained model and a set of samples, it assesses whether an classifier can accurately determine which samples were included in the model's training set based on the model's outputs.
- **Differential Privacy** [1, 2]: To ensure that the model does not depend on a small subset of training examples, we remove a minimal set of training samples and evaluate whether the distribution of model outputs undergoes an obvious change.

Uniqueness Testing. We randomly extract a subset of trajectories from the training set and use an autoregressive process to generate new trajectories conditioned on each sampled trajectory. We then compute the pairwise similarity between each original sample and its corresponding generated trajectory.If the lengths of two trajectories are different, the similarity score is defined as 0. If they are of equal length, the similarity score is calculated as the proportion of positions where both the timestamp and location ID exactly match. For each generated trajectory, we compute its similarity score with the top-1, top-3, and top-5 most similar real trajectories.

Figure 4 presents the cumulative distribution of similarity scores. As shown in the figure, over 95% of the generated trajectories do not have any corresponding real trajectory with a similarity score higher than 50%. This indicates that the model's outputs are based on the knowledge it has acquired, rather than directly copying trajectories from the training set.

Membership Inference Attack. Following the experimental setup in [28, 39], we use the similarity between generated trajectories and their corresponding real input trajectories as the classification feature. For each trajectory x, MoveGCL generates $\tilde{x}$ autoregressively conditioned on x, and we compute a similarity score $s(x, \tilde{x})$ to form the input to the classifier. The classifier is then tasked with determining whether x was included in the model's training set. Positive samples consist of real-world trajectories that were used during training, while negative samples are real trajectories from the same city that were held out. We evaluate the attack success rate, defined as the proportion of samples for which the classifier correctly infers membership status. We employ three widely used

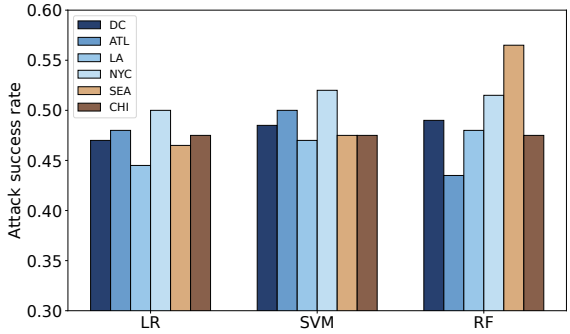


Figure 5: Success rate in membership inference attack

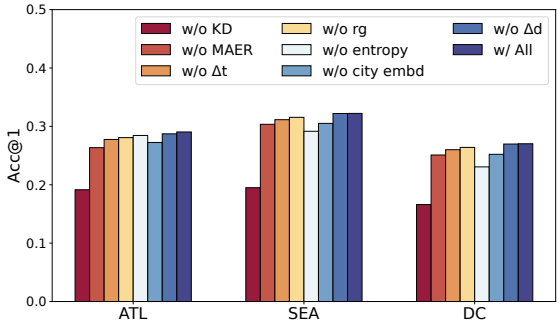


Figure 6: Ablation study. "w/o KD" denotes removal of the knowledge distillation loss; "w/o MAER" denotes removal of mobility feature from MoE transformer's router inputs.

Table 3: Differential Privacy statistics by city.

ϵ	Mean	Median	75th Percentile
Atlanta	2.671	0.706	2.212
Chicago	2.919	0.752	2.504
Los Angeles	2.988	0.593	2.572
New York	3.394	0.713	2.001
Seattle	2.934	0.655	1.870
Washington D.C	3.037	0.600	1.787

classification algorithms: Logistic Regression (LR), Support Vector Machine (SVM) and Random Forest (RF).

Figure 5 shows the attack results. As observed, the success rates across all datasets are approximately 50%, indicating that the classifier can hardly determine whether a trajectory was part of the training data or not based on the generated sample. These results indicate that our model is not easily susceptible to membership inference attacks.

Differential Privacy. For any pair of datasets D and D' that differ by only a small number of training trajectories, a model M is said to satisfy (ε, δ)-differential privacy if the following condition holds:

$$\mathbb{P}\big[M(z;\, D) = z\big] \;\leq\; e^{\varepsilon}\, \mathbb{P}\big[M(z;\, D') = z\big] \;+\; \delta, \tag{18}$$

where $\mathbb{P}\big[M(z;\, D) = z\big]$ denotes the probability of observing output z when the model is trained on dataset D, and $\mathbb{P}\big[M(z;\, D') = z\big]$ is defined analogously for dataset D'. Smaller values of ε and δ imply stronger privacy guarantees, since the model's output distribution becomes less dependent on any single trajectory.

In our experiment, we randomly select a subset of trajectories and consider two training scenarios: one in which this subset is included in the training data (D), and one in which it is excluded (D'). For each scenario, we train M on the corresponding dataset and then use each selected trajectory as a conditioning input to generate multiple synthetic trajectories. We compute a similarity score between each generated trajectory and its original conditioning trajectory. These similarity scores are modeled as two Gaussian distributions corresponding to $\mathbb{P}\big[M(z;\, D)\big]$ and $\mathbb{P}\big[M(z;\, D')\big]$, respectively. Finally, we estimate the privacy-budget parameters ε from these distributions. As shown in Table 3, without applying any additional privacy-preserving mechanisms, MoveGCL naturally achieves a privacy budget of $\varepsilon \approx 1$–2 for 75% of randomly sampled trajectories. This level is generally considered an acceptable operating point for generative models [27]; for example, Apple adopts a privacy budget of $\varepsilon = 4.0$ [*].

5.5 Ablation Studies

In this section, we conduct two sets of incremental ablation studies based on MoveGCL (WSC→A→L→N). The first set focuses on the input features of the Mobility-Aware Expert Routing module. We selectively remove or adjust different dimensions of the location feature to evaluate the contribution and necessity of each type of input in guiding expert routing. The second set targets the incremental learning mechanism itself. We remove the knowledge distillation strategy designed to mitigate catastrophic forgetting in GCL, and instead train the model using only the conventional cross-entropy loss. This setup allows us to assess the effectiveness of knowledge distillation in preserving previously learned knowledge.

As shown in Figure 6, removing any input feature from the Mobility-Aware Expert Routing module leads to a noticeable performance drop. Similarly, disabling the knowledge distillation strategy in GCL also results in a significant decline in model performance. These findings highlight the critical role of each input feature in expert selection, as well as the importance of knowledge distillation in ensuring model stability during continual learning.

5.6 In-Depth Analysis

To better understand why MoveGCL is capable of unifying diverse mobility datasets and effectively handling substantial inter-city heterogeneity, we conduct an in-depth analysis of the location embedding layer to examine whether MoveGCL can learn shared spatial representations across cities. To this end, we extract the location embeddings for each city at two stages: (1) after the initial encoder, and (2) after the Deep & Cross Network (DCN). By comparing these two sets of embeddings, we evaluate the role of DCN in aligning spatial semantics across heterogeneous urban environments.

As shown in Figure 7, DCN aligns location embedding distributions in different cities much more closely than the original encoder output. This indicates that DCN successfully captures shared location-feature patterns across urban areas, thereby boosting the model's ability to generalize in cross-city settings. Moreover, within

[*]https://www.apple.com/privacy/docs/Differential_Privacy_Overview.pdf

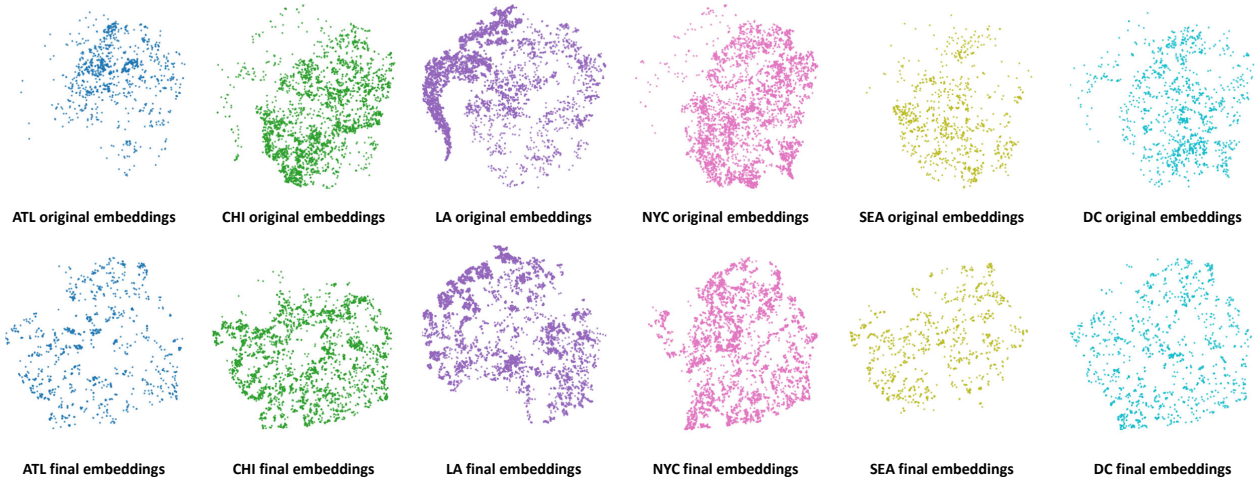


Figure 7: City location embeddings before (original) and after (final) DCN.

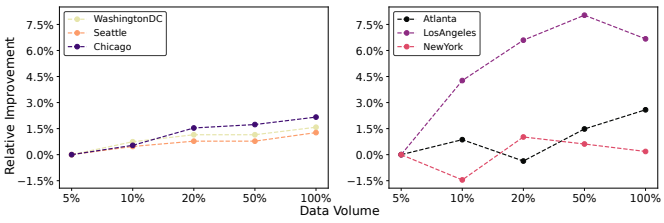


Figure 8: Acc@1 changes at different generated data ratios (α), relative to $\alpha = 5\%$.

each city, the DCN-processed embeddings become less densely clustered than their original counterparts, indicating a marked increase in separability among individual locations and further enhancing the model's capacity to encode location semantics.

5.7 Impact of Replay Volume

In generative continual learning, synthetic data replay serves as a key mechanism for preserving previously acquired knowledge without accessing raw data. A critical hyperparameter in this process is the volume of generated data used during training on new cities. While too little replay data may result in catastrophic forgetting, excessive generation increases computational costs and may introduce noise or redundancy. Understanding this trade-off is essential for building scalable and efficient mobility foundation models. To explore this, we vary the amount of generated data and evaluate its impact on both knowledge retention (for previously seen cities) and adaptation to new cities.

As shown in Figure 8, the performance on the base cities (WSC) consistently improves with more generated data, indicating that replay volume directly affects the ability to retain past knowledge. In contrast, performance on the newly introduced cities (A, L, N) remains largely stable regardless of replay volume, with no consistent trend of improvement or degradation. These results suggest that while synthetic replay is crucial for mitigating forgetting, it has limited effect on new knowledge acquisition. Thus, allocating a moderate amount of generated data offers a practical balance—sufficient to preserve prior knowledge without incurring unnecessary overhead—supporting the long-term scalability.

6 Conclusion

In this work, we present MoveGCL, a scalable and privacy-preserving framework for training mobility foundation models via generative continual learning. By enabling decentralized model evolution without sharing raw data, it addresses key challenges in real-world human mobility modeling, including data silos, privacy constraints, and heterogeneous mobility distributions. MoveGCL represents a significant step toward realizing mobility foundation models by offering a practical and generalizable framework that facilitates collaborative learning across cities and institutions. It paves the way for long-term, privacy-safe, and adaptive modeling of human movement, with broad implications for urban planning, transportation optimization, and evidence-based policy making. The development of more large-scale, semantically rich, and geographically diverse mobility datasets will be crucial for further improving the model's generalization and robustness. We encourage the broader research community and data-holding institutions to join this collaborative effort, contributing to the creation of open, inclusive, and powerful spatiotemporal foundation models for the mobility domain.

Acknowledgments

This work was supported in part by the National Key Research and Development Program of China under grant 2024YFC3307603 and the National Natural Science Foundation of China under 62476152.

References

[1] Martin Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. 2016. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS)*. ACM, 308–318.

[2] Galen Andrew, Steve Chien, and Nicolas Papernot. 2019. TensorFlow Privacy: Learning with Differential Privacy for Training Machine Learning Models. https://blog.tensorflow.org/2019/03/introducing-tensorflow-privacy-learning.html. Accessed: 2025-06-06.

[3] Hugo Barbosa, Marc Barthelemy, Gourab Ghoshal, Charlotte R James, Maxime Lenormand, Thomas Louail, Ronaldo Menezes, José J Ramasco, Filippo Simini, and Marcello Tomasini. 2018. Human mobility: Models and applications. *Physics Reports* 734 (2018), 1–74.

[4] Xiao Bi, Deli Chen, Guanting Chen, Shanhuang Chen, Damai Dai, Chengqi Deng, Honghui Ding, Kai Dong, Qiushi Du, Zhe Fu, et al. 2024. DeepSeek LLM: Scaling Open-Source Language Models with Longtermism. *CoRR* (2024).
[5] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems* 33 (2020), 1877–1901.
[6] Haoye Chai, Yuan Yuan, and Yong Li. 2025. MobiWorld: World Models for Mobile Wireless Network. *arXiv preprint arXiv:2507.09462* (2025).
[7] Shushman Choudhury, Abdul Rahman Kreidieh, Ivan Kuznetsov, and Neha Arora. 2024. Towards a Trajectory-powered Foundation Model of Mobility. In *Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Spatial Big Data and AI for Industrial Applications*. 1–4.
[8] Yves-Alexandre de Montjoye, César A. Hidalgo, Michel Verleysen, and Vincent D. Blondel. 2013. Unique in the crowd: The privacy bounds of human mobility. *Scientific Reports* 3 (2013), 1376. doi:10.1038/srep01376
[9] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. 2024. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*.
[10] Jie Feng, Yuwei Du, Jie Zhao, and Yong Li. 2024. Agentmove: Predicting human mobility anywhere using large language model based agentic framework. *arXiv preprint arXiv:2408.13986* (2024).
[11] Jie Feng, Yong Li, Zeyu Yang, Qiang Qiu, and Depeng Jin. 2022. Predicting Human Mobility With Semantic Motivation via Multi-Task Attentional Recurrent Networks. *IEEE Transactions on Knowledge and Data Engineering* 34, 5 (2022), 2360–2374. doi:10.1109/TKDE.2020.3006048
[12] Jie Feng, Can Rong, Funing Sun, Diansheng Guo, and Yong Li. 2020. PMF: A privacy-preserving human mobility prediction framework via federated learning. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 4, 1 (2020), 10:1–10:21. doi:10.1145/3381006
[13] 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*. ACM, Bern, Switzerland, 3–8.
[14] Letian Gong, Yan Lin, Yiwen Lu, Xuedi Han, Yichen Liu, Shengnan Guo, Youfang Lin, Huaiyu Wan, et al. 2024. Mobility-llm: Learning visiting intentions and travel preference from human mobility data with large language models. *Advances in Neural Information Processing Systems* 37 (2024), 36185–36217.
[15] Chonghua Han, Yuan Yuan, Kaiyan Chen, Jingtao Ding, and Yong Li. 2025. TrajMoE: Spatially-Aware Mixture of Experts for Unified Human Mobility Modeling. *arXiv preprint arXiv:2505.18670* (2025).
[16] WANG JIAWEI, Renhe Jiang, Chuang Yang, Zengqing Wu, Ryosuke Shibasaki, Noboru Koshizuka, Chuan Xiao, et al. 2024. Large language models as urban residents: An llm agent framework for personal mobility generation. *Advances in Neural Information Processing Systems* 37 (2024), 124547–124574.
[17] Chenlu Ju, Jiaxin Liu, Shobhit Sinha, Hao Xue, and Flora Salim. 2025. TrajLLM: A Modular LLM-Enhanced Agent-Based Framework for Realistic Human Trajectory Simulation. *arXiv preprint arXiv:2502.18712* (2025).
[18] Gyuhak Kim, Changnan Xiao, Tatsuya Konishi, Zixuan Ke, and Bing Liu. 2022. A theoretical study on solving continual learning. *Advances in neural information processing systems* 35 (2022), 5065–5079.
[19] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems* 35 (2022), 22199–22213.
[20] Dejiang Kong and Fei Wu. 2018. HST-LSTM: A hierarchical spatial-temporal long-short term memory network for location prediction. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI)*. AAAI Press, Stockholm, Sweden, 2341–2347.
[21] Xiangjie Kong, Qiao Chen, Mingliang Hou, Hui Wang, and Feng Xia. 2023. Mobility trajectory generation: a survey. *Artificial Intelligence Review* 56, Suppl 3 (2023), 3057–3098.
[22] Xilai Li, Yingbo Zhou, Tianfu Wu, Richard Socher, and Caiming Xiong. 2019. Learn to grow: A continual structure learning framework for overcoming catastrophic forgetting. In *International conference on machine learning*. PMLR, 3925–3934.
[23] Zhonghang Li, Long Xia, Lei Shi, Yong Xu, Dawei Yin, and Chao Huang. 2024. Opencity: Open spatio-temporal foundation models for traffic prediction. *arXiv preprint arXiv:2408.10269* (2024).
[24] Zhonghang Li, Lianghao Xia, Jiabin Tang, Yong Xu, Lei Shi, Long Xia, Dawei Yin, and Chao Huang. 2024. Urbangpt: Spatio-temporal large language models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 5351–5362.
[25] Yuxuan Lin, Hongxu Wan, Shenglin Guo, and Yantao 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. doi:10.1609/aaai.v35i5.16548
[26] Yan Lin, Tonglong Wei, Zeyu Zhou, Haomin Wen, Jilin Hu, Shengnan Guo, Youfang Lin, and Huaiyu Wan. 2024. TrajFM: A vehicle trajectory foundation model for region and task transferability. *arXiv preprint arXiv:2408.15251* (2024).
[27] Zinan Lin, Alankar Jain, Chen Wang, Giulia Fanti, and Vyas Sekar. 2020. Using gans for sharing networked time series data: Challenges, initial promise, and open questions. In *Proceedings of the ACM internet measurement conference*. 464–483.
[28] Zinan Lin, Alankar Jain, Chen Wang, Giulia Fanti, and Vyas Sekar. 2020. Using GANs for Sharing Networked Time Series Data: Challenges, Initial Promise, and Open Questions. In *Proceedings of the ACM Internet Measurement Conference (IMC)*. 464–483.
[29] Yixin Liu, Kai Zhang, Yuan Li, Zhiling Yan, Chujie Gao, Ruoxi Chen, Zhengqing Yuan, Yue Huang, Hanchi Sun, Jianfeng Gao, et al. 2024. Sora: A review on background, technology, limitations, and opportunities of large vision models. *arXiv preprint arXiv:2402.17177* (2024).
[30] Ziqiao Liu, Hao Miao, Yan Zhao, Chenxi Liu, Kai Zheng, and Huan Li. 2024. LightTR: A Lightweight Framework for Federated Trajectory Recovery. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*. 4422–4434. doi:10.1109/ICDE60146.2024.00337
[31] Qingyue Long, Yuan Yuan, and Yong Li. 2024. A Universal Model for Human Mobility Prediction. *arXiv preprint arXiv:2412.15294* (2024).
[32] 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.
[33] Saeed Masoudnia and Reza Ebrahimpour. 2014. Mixture of experts: a literature survey. *Artificial Intelligence Review* 42 (2014), 275–293.
[34] New York City Taxi and Limousine Commission. 2023. TLC Trip Record Data. https://www.nyc.gov/site/tlc/about/tlc-trip-record-data.page. [Online; accessed 16-November-2023].
[35] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*. PmLR, 8748–8763.
[36] Can Rong, Jingtao Ding, and Yong Li. 2024. An interdisciplinary survey on origin-destination flows modeling: Theory and techniques. *Comput. Surveys* 57, 1 (2024), 1–49.
[37] Markus Schläpfer, Lei Dong, Kevin O'Keeffe, Paolo Santi, Michael Szell, Hadrien Salat, Samuel Anklesaria, Mohammad Vazifeh, Carlo Ratti, and Geoffrey B West. 2021. The universal visitation law of human mobility. *Nature* 593, 7860 (2021), 522–527.
[38] 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. *CoRR* (2024).
[39] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. 2017. Membership inference attacks against machine learning models. In *Proceedings of the 2017 IEEE Symposium on Security and Privacy (SP)*. IEEE, 3–18.
[40] 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* 23, 5 (2023), 4849–4860.
[41] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NeurIPS)*. Curran Associates Inc., Long Beach, California, USA, 6000–6010.
[42] Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. 2024. A comprehensive survey of continual learning: Theory, method and application. *IEEE transactions on pattern analysis and machine intelligence* 46, 8 (2024), 5362–5383.
[43] 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.
[44] Buddhi Wickramasinghe, Gobinda Saha, and Kaushik Roy. 2023. Continual learning: A review of techniques, challenges, and future directions. *IEEE Transactions on Artificial Intelligence* 5, 6 (2023), 2526–2546.
[45] Fengli Xu, Zhen Tu, Yong Li, Pengyu Zhang, Xiaoming Fu, and Depeng Jin. 2017. Trajectory recovery from ash: User privacy is not preserved in aggregated mobility data. In *Proceedings of the 26th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 1241–1250.
[46] Takahiro Yabe, Kota Tsubouchi, Toru Shimizu, Yoshihide Sekimoto, Kaoru Sezaki, Esteban Moro, and Alex Pentland. 2024. YJMob100K: City-scale and longitudinal dataset of anonymized human mobility trajectories. *Scientific Data* 11, 1 (2024), 397.
[47] Dingqi Yang, Bingqing Qu, Jie Yang, and Philippe Cudre-Mauroux. 2019. Revisiting user mobility and social relationships in lbsns: a hypergraph embedding approach. In *The world wide web conference*. 2147–2157.
[48] Jaehong Yoon, Wonyong Jeong, Giwoong Lee, Eunho Yang, and Sung Ju Hwang. 2021. Federated continual learning with weighted inter-client transfer. In *International conference on machine learning*. PMLR, 12073–12086.
[49] Jing Yuan, Yu Zheng, Chengyang Zhang, Wenlei Xie, Xing Xie, Guangzhong Sun, and Yan Huang. 2010. T-drive: driving directions based on taxi trajectories.

In *Proceedings of the 18th SIGSPATIAL International conference on advances in geographic information systems*. 99–108.
[50] 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.
[51] Yuan Yuan, Jingtao Ding, Chonghua Han, Zhi Sheng, Depeng Jin, and Yong Li. 2024. UniFlow: A Foundation Model for Unified Urban Spatio-Temporal Flow Prediction. *arXiv preprint arXiv:2411.12972* (2024).
[52] 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.
[53] 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), 29:1–29:28. doi:10.1145/3637493
[54] 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.
[55] 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).
[56] Yuan Yuan, Yuheng Zhang, Jingtao Ding, and Yong Li. 2025. WorldMove, a global open data for human mobility. *arXiv preprint arXiv:2504.10506* (2025).
[57] Junbo Zhang, Yu Zheng, and Dekang Qi. 2017. Deep spatio-temporal residual networks for citywide crowd flows prediction. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 31.
[58] Weijia Zhang, Jindong Han, Zhao Xu, Hang Ni, Hao Liu, and Hui Xiong. 2024. Urban foundation models: A survey. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 6633–6643.
[59] Yuheng Zhang, Yuan Yuan, Jingtao Ding, Jian Yuan, and Yong Li. 2025. Noise Matters: Diffusion Model-based Urban Mobility Generation with Collaborative Noise Priors. In *Proceedings of the ACM on Web Conference 2025*. 5352–5363.
[60] Yu Zheng, Hao Fu, Xing Xie, Wei-Ying Ma, and Quannan Li. [n. d.]. Geolife GPS trajectory dataset-User Guide, geolife gps trajectories 1.July 2011, geolife GPS trajectories 1.1. *July 2011 geolife GPS trajectories 1.1* ([n. d.]).
[61] Zhen Zhou, Ziyuan Gu, Xiaobo Qu, Pan Liu, Zhiyuan Liu, and Wenwu Yu. 2024. Urban mobility foundation model: A literature review and hierarchical perspective. *Transportation Research Part E: Logistics and Transportation Review* 192 (2024), 103795.
[62] Yuanshao Zhu, Yongchao Ye, Ying Wu, Xiangyu Zhao, and James Yu. 2023. Synmob: Creating high-fidelity synthetic gps trajectory dataset for urban mobility analysis. *Advances in Neural Information Processing Systems* 36 (2023), 22961–22977.
[63] 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).

A Dataset

Table 4: Basic statistics of mobility data.

City	User	Trajectory	Location
Atlanta	114941	2348218	1175
Chicago	148000	8051522	4166
Los Angeles	161544	16844127	6198
New York	170321	15766369	4988
Seattle	88569	3362353	1046
Washington D.C	134442	11024181	1361

B Evaluation Metrics

We adopt top-k accuracy as our evaluation metric, defined as

$$\text{acc@}k \;=\; \frac{1}{N}\sum_{i=1}^{N}\mathbf{1}\big(x_i \in f_k(x_i)\big)\,, \tag{19}$$

where N is the total number of samples, x_i is the ground-truth label for the i^{th} sample, $f_k(x_i)$ denotes the set of the model's top-k predicted labels for sample i, and $\mathbf{1}(\cdot)$ is the indicator function that equals 1 if its argument is true and 0 otherwise. We report on acc@1 and acc@3 to assess the performance of the model.