GAMS: Geospatial Knowledge Enhanced Agentic Framework for Urban Human Mobility Behavior Simulation

Anonymous ACL submission

Abstract

Recently, researchers have explored leveraging the commonsense knowledge and reasoning capabilities of large language models to accelerate human mobility simulation. However, these methods suffer from several critical shortcomings, including inadequate modeling of urban spaces, poor integration with individual mobility patterns, and weak alignment with collective mobility distributions. To address these limitations, we propose **G**eospatial Knowledge Enhanced **A**gentic framework for **M**obility **S**imulation (**GAMS**), an agentic framework that leverages the language based geospatial foundation model to simulate human mobility in urban space. **GAMS** comprises three core modules: MobExtractor, which extracts template-based mobility profiles and synthesizes new ones during simulation; GeoGenerator, which generate trajectory points considering collective mobility knowledge and urban geospatial knowledge; and TrajEnhancer, which captures individual mobility regularities from real-world trajectories and refine to generate style-aligned synthetic trajectories. Experiments on two real-world datasets show that **GAMS** achieves superior performance over various state-of-the-art methods, with more than a 17% improvement by capturing diverse mobility regularities. Our codes and datasets are open-sourced via https://anonymous.4open.science/r/CAMS-1992/.

1 Introduction

Human mobility simulation is a critical real-world task with widespread applications across many domains (Pappalardo et al., 2023), such as supporting the implementation of the 15-minute city concept in urban development by modeling residents' daily activities (Zheng et al., 2023), optimizing transportation strategies through travel behavior simulation, and validating intervention policies in epidemic prevention and control. Given its significant value, the research community has studied this problem extensively for many years, resulting in a range of effective solutions. Early efforts, such as mechanism-based models like TimeGeo (Jiang et al., 2016), have gradually been supplemented—and surpassed—by recent deep learning approaches, such as MoveSim (Feng et al., 2020), ActSTD (Yuan et al., 2022), and DSTPP (Yuan et al., 2023) and on. Despite remarkable progress, key challenges remain—particularly concerning the spatial transferability of methods, as well as the controllability and interpretability of the generated mobility behaviors.

To address these challenges, recent research has explored integrating LLMs into mobility simulation, leveraging their role-playing (Gao et al., 2024; Wang et al., 2024b; Gao et al., 2023; Piao et al., 2025), commonsense knowledge (OpenAI, 2022; Touvron et al., 2023; Ding et al., 2024) and reasoning capabilities (Wei et al., 2022; Xu et al., 2025) to achieve promising results. The most crucial and challenging aspect of applying LLMs to mobility simulation lies in effectively incorporating spatial information. Existing work (Gurnee and Tegmark, 2024; Shao et al., 2024) has shown that simply utilizing general-purpose LLMs is insufficient for accurately understanding urban space. As a result, studies such as CoPB (Shao et al., 2024) and LLMob (Wang et al., 2024a) have proposed specific mechanisms within their frameworks to mitigate this limitation and harness the strengths of LLMs for sequential modeling and reasoning. However, these approaches typically combine spatial knowledge and LLMs in a relatively independent manner, fusing information in an ad hoc fashion. Moreover, spatial knowledge is often simplified to facilitate model comprehension, and the integration process remains largely unidirectional, lacking feedback-driven optimization or iterative reasoning updates.

Recently, geospatial LLMs such as CityGPT (Feng et al., 2025b) and LAMP (Balsebre et al., 2024) have emerged, directly enhancing

general LLMs with urban spatial knowledge through post-training and achieving impressive results on geospatial tasks such as urban spatial knowledge question answering. In these works, they convert the urban spatial knowledge into the language format and train the general model to enhance the urban spatial knowledge. This progress offers a new perspective on incorporating spatial knowledge into LLMs and enables deeper collaboration between spatial knowledge and spatial reasoning in downstream task.

In this paper, we introduce **GAMS**, an agentic framework that achieves more controllable, accurate, and generalizable human mobility simulation. The framework operates through three synergistic core components. First, **MobExtractor** analyzes raw trajectory data to extract and summarize high-level mobility profiles for enabling the controllable generation of mobility. Next, **GeoGenerator** leverages a geospatially-aware LLM with specific mechanisms to translate these abstract patterns into realistic location sequences with collective distribution constrains. Finally, **TrajEnhancer** ensures the output's fidelity, it employs direct preference optimization to perform **trajectory style alignment**, conforming the generated paths to real-world data patterns and enforcing spatio-temporal consistency. A multi-dimensional feedback mechanism unifies this three-stage pipeline, enabling **GAMS** to iteratively refine the generation process. This continuous improvement loop enhances both the realism and adaptability of the simulated human mobility. Besides, built upon an enhanced geospatial-knowledge enhanced LLM, **GAMS** natively integrates urban spatial knowledge into the reasoning and generation process of LLMs.

In summary, our contributions are:

- To our knowledge, **GAMS** is the first to integrate an geospatial foundation model with rich geospatial knowledge and multi-dimensional feedback for controllable and generalized mobility simulation.
- For controllable generation, we design a dual-phase architecture based MobExtractor, which condenses mobility patterns into compact linguistic representations, then synthesizes new user-specific patterns through profile-aware feature fusion.
- To generate mobility patterns that align with collective urban mobility regularities, we build a geospatial foundation model with fine-grained urban geographic knowledge fine-tuning and tailored mechanism design.
- We introduce multi-dimensional feedback in the framework to progressively align the generated trajectories with real-world trajectories, ensuring that the simulated mobility better conforms to individual mobility regularity and style.
- Experimental results on two real-world datasets demonstrate that the proposed **GAMS** framework significantly outperforms existing methods in human mobility simulation, achieving more than a 17% improvement in a composite metric capturing various mobility regularities.

2 Related Work

Mobility Simulation. Mobility simulation includes synthesizing data based on statistical laws, generating simulated data in virtual space, and generating mobility data in real urban spaces. On the basis of macroscopic statistical laws (Brockmann et al., 2006; Roth et al., 2011; Liang et al., 2012), researchers proposed a series of mobility simulation models to depict individual behavior mechanism (Pappalardo and Simini, 2018; Wang et al., 2019; Pappalardo and Simini, 2018; Jiang et al., 2016). While these mechanism models are concise but fail to capture complex human mobility patterns and model the impact of urban structure. With the rapid development of deep learning, different model structures were designed to model the complex dynamics of mobility behaviors (Feng et al., 2020; Long et al., 2023; Liu et al., 2024). However, these deep learning methods face challenges of data sparsity, poor transferability and low explainability.

LLM for Geospatial Tasks. Since LLMs are geospatially knowledgeable (OpenAI, 2022; Bhandari et al., 2023; Touvron et al., 2023), researchers pay attention to leverage LLM in geography and urban science field by solving domain-specific tasks like geospatial understanding tasks (Brown et al., 2020; Mai et al., 2023; Roberts et al., 2023; Feng et al., 2025b) and geospatial prediction tasks (Wang et al., 2023; Beneduce et al., 2025; Feng et al., 2025a; Gong et al., 2024). LLMs can achieve good results in global-scale or national-scale tasks with simple prompt engineering (Manvi et al., 2024b,a) or a trained linear layer (Gurnee and Tegmark, 2024). However, when breaks down to city scale, well-designed agentic frameworks and fine-tuning

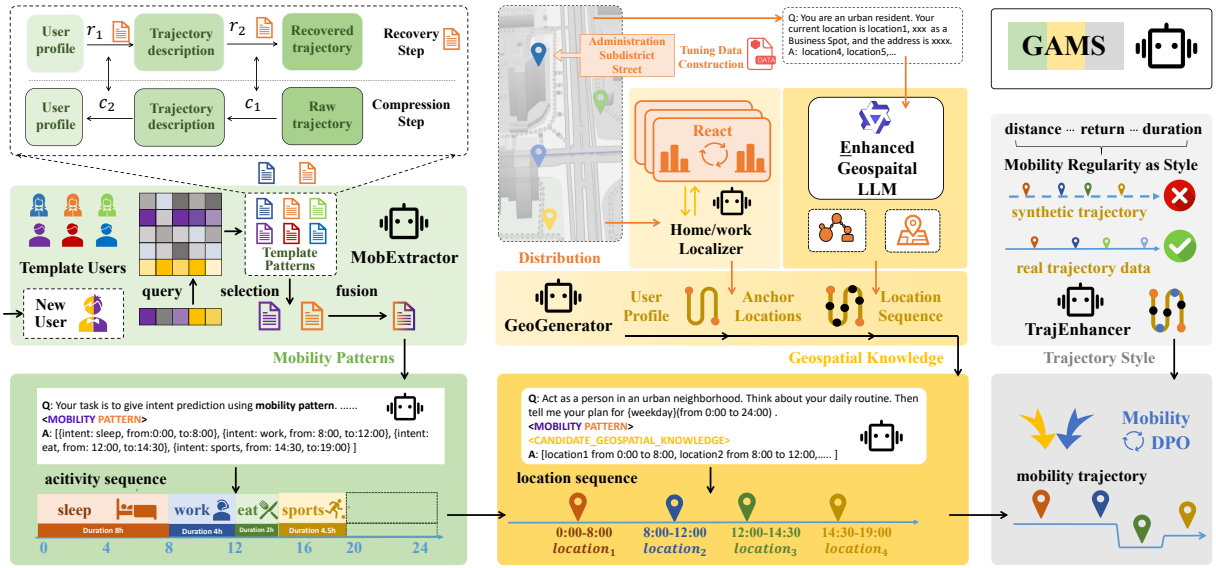


Figure 1: The framework of **GAMS**, with components arranged from left to right: MobExtractor, GeoGenerator, and TrajEnhancer.

methods are required to enable LLMs to acquire urban structural knowledge (Feng et al., 2025b; Balsebre et al., 2024) and enhance task-specific performance via geospatial knowledge alignment. However, they are primarily designed for question-answering tasks and performs poorly on complex, structured tasks like trajectory generation, without an agentic framework that orchestrates multiple specialized modules to leverage their knowledge effectively for simulation.

LLM for Mobility Simulation. With the successful application of LLM in geospatial tasks, researchers are exploring the potential of applying LLMs to human mobility simulation (Wang et al., 2024a; Shao et al., 2024; Wang et al., 2024a; Li et al., 2024; Jiang et al., 2024; Yan et al., 2024). They extract individual knowledge from the user profile and historical trajectories, then synthesize simulated data (Wang et al., 2024a), map simulated data to real urban spaces using mechanistic models (Shao et al., 2024; Li et al., 2024), or generate real-world trajectories based on the given urban spatial information (Wang et al., 2024a). They perform well in few-shot scenarios and exhibit good transferability. However, they insufficiently model real urban structures, and fail to capture collective mobility patterns.

3 Methodology

GAMS is an agentic framework for generating trajectories in real urban spaces by incorporating urban-knowledgeable LLM with multi-dimensional feedback. As shown in Figure 1, our methodology is a three-stage pipeline that progressively generates realistic human mobility behaviors. Details of each components are introduced below.

3.1 MobExtractor: Semantic Mobility Pattern Extraction for Controllable Simulation

As shown in the left part of Figure 1, to enable controllable mobility simulation through user profiles, we design MobExtractor with a "compress-then-recover" architecture to extract semantic mobility pattern descriptions. This structure facilitates diverse and flexible pattern extraction without relying on manually designed prompts. Without the compression stage, generating activity sequences from raw user profiles would require extensive human effort in crafting effective prompts, which is both labor-intensive and difficult to scale. Instead, we leverage an LLM to first compress raw trajectories into compact linguistic representations, and then recover the trajectories by referencing these compressed prompts. This reconstruction process enables us to automatically derive high-quality activity sequence generation templates. Based on numerous user profiles paired with their corresponding activity pattern templates, we can generate personalized activity sequences for new users by measuring their similarity to existing templates.

Building on this framework, MobExtractor implements a dual-phase pipeline: compression and generation. In the compression phase, mobility patterns from template users are distilled into com-

pact linguistic representations that capture both shared routines (e.g., commuting, dining) and individual deviations (e.g., irregular visit times, unique location preferences). These representations are derived through LLM-based semantic abstraction, allowing the model to identify generic mobility regularities from limited data—alleviating the reliance on large-scale trajectory datasets. In the generation phase, for new users, we first retrieve semantically similar templates based on profile embeddings, then synthesize personalized activity sequences via profile-aware feature fusion and variational encoding. This approach enables scalable and data-efficient pattern generation, leveraging LLMs' world knowledge to generalize across users while preserving individual characteristics.

Mobility pattern recovery. As shown in Figure 1, in the mobility patterns reconstruction phase, the model learns high-level correlations between user profiles, semantic trajectory descriptions, and raw mobility patterns through a dual-phase compression-reconstruction process. The model automatically distill observed patterns and correlations into interpretable natural language rules, including $\mathbf{c}_1$, $\mathbf{c}_2$ in compression stage and $\mathbf{r}_1$, $\mathbf{r}_2$ in reconstruction stage.

- **Compression.** In compression stage, the model learns compression patterns that map raw trajectory data to user profile representations, i.e., (1) how to derive users' behavioral habits and motivations by analyzing statistical patterns in their historical trajectories, (2) how to identify user's mobility pattern from raw trajectory, habits, motivations and address information, (3) how to identify profile-influencing features from trajectory descriptions. The compression patterns obtained in this stage are denoted as $\mathbf{c}_1$ and $\mathbf{c}_2$,
- **Reconstruction.** During reconstruction, the model acquires reconstruction patterns that map user profiles back to raw trajectories, i.e., (1) how to identify components most predictive of trajectory description from user profile based on key profile determinants identified in $\mathbf{c}_2$, (2) how to generate user's raw trajectory from trajectory description and candidate POIs based on $\mathbf{c}_1$. With the above compression patterns $\mathbf{c}_1$ and $\mathbf{c}_2$ as reference, reconstruction stage generate $\mathbf{r}_1$ and $\mathbf{r}_2$, which are preserved to condition the trajectory generation process for new users.

Mobility pattern generation. As shown in Figure 1, in the generation phase, the model generates mobility patterns for any users with only profile information. To enhance the model's generalization capability, we retrieve the top K most similar template users (training users) for each new user (test user) and then combine these template patterns to generate mobility pattern in the forms of activity sequence. We compare following two strategies for retrieving similarity individuals.

- **Language-based**: Use LLM to select the top K most similar users based on semantic user profile characteristics, then directly output the ID and similarity score of each selected user.
- **Embedding-based**: Find similar users based on similarity scores of user profile embeddings(Yu et al., 2024). First, we construct a template user profile embedding matrix $\mathbf{E}_{\text{template}} \in \mathbb{R}^{m \times d}$, using the profiles of m template users. Then we encode user profile of new user into an embedding $\mathbf{e}_{1\times d}$, computing cosine similarities sim_ibetween $\mathbf{e}_{1\times d}$ and $\mathbf{E}_{\text{template}}$. Finally, we retrieve the top K users $\mathcal{T}$ with highest similarity scores.

After acquiring similar users, we sequentially perform the following steps: (1) $\mathbf{c}_2$ Feature Fusion: Use LLM to integrate key profile factors and high-order mobility characteristics in $\mathbf{c}_2$ of the similar template users. (2) Trajectory Description Generation: Using the fused features, generate trajectory descriptions by referencing $\mathbf{r}_1$ and $\mathbf{r}_2$ in compression stage. (3) $\mathbf{c}_1$ Feature Fusion: Use LLM to integrate both the unique movement patterns and universal movement patterns in $\mathbf{c}_1$ of the similar template users.

3.2 GeoGenerator: Integrating Geospatial and Collective Mobility Knowledge

As previously discussed, existing methods struggle to effectively leverage urban structural knowledge. To address this challenge, following the recent practice from community (Balsebre et al., 2024; Feng et al., 2025b), we train a urban-knowledgeable LLM as the foundation for integrating geospatial knowledge.

As shown in the middle of Figure 1, GeoGenerator features two key components: the Anchor Location Extractor and the Urban Structure Mapper. First, the Anchor Location Extractor generates critical anchor points based on user profiles, collective mobility distributions, and urban geographic knowledge. These anchor points are then

transformed into intent-composed trajectories by integrating mobility patterns extracted in the first stage. Subsequently, the Urban Structure Mapper, built upon an enhanced geospatial LLM, maps these intent-driven trajectories into realistic urban locations, aligning them with the underlying city structure. Detailed prompts of Geogenerator can refer to Appendix A.9.

3.2.1 Anchor Location Extractor

The locations of homes and workplaces serve as the most important anchor points in human mobility trajectories, significantly shaped by individual user profiles and regional characteristics. To effectively identify these critical anchor locations, we propose a two-step extraction method.

Macro-to-micro cascaded generation. We propose a macro-to-micro cascaded generation system with iterative reasoning-execution-reflection cycles(Yao et al., 2023; Du et al., 2024) to progressively refine spatial distributions. First, we transfer coordinates of all homes and workplaces into a hierarchical address representation, namely administrative area → subdistrict → street → POI. For regions in each hierarchy, we calculate user profile distributions and generate descriptive summaries. Then, from coarse (administrative area) to fine (street) spatial scales, the model hierarchically generates home/workplace assignments by propagating upper hierarchy outputs as contextual constraints for finer-grained reasoning. In reasoning stage, model consider descriptive summaries and geographical knowledge of child regions contained within each parent region's extent (upper hierarchy) and user profile characteristics. In execution stage, the model select a region that best matches user profile characteristics guided by the reasoning stage. In reflection stage, the model performs periodic distribution-aware reflection. Finally, in POI spatial scale, the model directly generates the precise location of home/workplace.

Reflection with collective distribution. We incorporate collective knowledge as feedback in reflection stage, progressively aligning generated results with distribution in real urban spaces. Upon completing execution stage of all users, we compute spatial distribution of generated locations. Then, in reflection stage, the model does comparative analysis against ground-truth distribution and adjusts generation strategies for subsequent iterations. Finally, in execution stage, model dynamically adjust individual output to minimize distributional divergence.

3.2.2 Urban Structure Mapper

To generate the remaining location points in a mobility trajectory beyond the two anchor locations (home and workplace) from the previous stage, we introduce an Urban Structure Mapper (referred to as UrbanMapper). Given the anchor points and activity sequences, this module flexibly integrates urban spatial structure information to synthesize the remaining trajectory points.

Enhancing Geospatial LLM To mitigate geographic hallucinations and improve spatial precision when generating specific location in real urban space, we augmented the knowledge embedded in LLM through fine-tuning with fine-grained urban spatial data. At a finer granularity, we posit that urban space is composed of three fundamental elements: points (POIs), lines (streets), and polygons (AOIs)(Dempsey et al., 2010). Among these, points (POIs) constitute the most basic building blocks, which also serve as the foundational components of trajectories. Therefore, we construct our training data based on POI-level granularity. To simulate human cognitive and exploratory processes in urban spaces, we generate navigation paths between population-weighted randomly sampled origin-destination (OD) POIs, recording all traversed POIs along the pathways, and subsequently identifying specified-category POIs within defined radius around each recorded waypoint. The radius is determined by the average jump distance between consecutive trajectory points and is correlated with the user's mobility pattern, while the category is related to user's intention at each time point. We construct a fine-tuning dataset comprising 10,000 question-answer pairs, encompassing all POIs of specified categories within certain radius around every sampled POI.

To activate the geospatial knowledge embedded in LLM, we enhance user profiles with address information to infer user approximate activity ranges in real urban spaces. Furthermore, we represent the geographic elements in datasets with semantically rich addresses rather than coordinates or grid-ID. We also investigate how different address representation formats impact the model's comprehension of geographical information: (1) **Hierarchical address representation**: Use structured address hierarchies (e.g., admin→subdistrict→street→POI) to guide the model in recalling location names and attributes within specific region, reducing hallu-

cinations and generating more realistic, specific locations. (2) **Human-intuitive geospatial representations**: Leverage human-intuitive geospatial representations (e.g., 100 meters from the intersection of Road B and Road C) to prompt the model to associate nearby locations and their attributes.

3.3 TrajEnhancer: Enhancing Mobility Trajectory with Mobility Style Alignment

After processing by MobExtractor and GeoGenerator, we obtain a preliminary location sequence that integrates user activity patterns and urban structural knowledge. However, such sequences may still lack fine-grained spatiotemporal coherence and individualized movement styles observed in real-world mobility. To bridge this gap, we introduce TrajEnhancer, which refines the generated trajectories through implicit alignment with real human mobility patterns using the Direct Preference Optimization (DPO) framework. This module ensures that synthetic trajectories not only follow high-level intents but also exhibit realistic temporal dynamics and spatial regularities.

As shown in the right part of Figure 1, TrajEnhancer performs integrated reasoning by synthesizing urban geospatial knowledge (generated in Section 3.2) and mobility patterns (extracted in Section 3.1). It first constructs daily activity plans for target users based on their profiles and historical patterns, consisting of semantic intentions (e.g., "dining out", "commuting") and temporal constraints (e.g., duration). Then, it generates realistic movement trajectories by jointly reasoning over user profiles, activity plans, anchor points, and contextual urban knowledge—ensuring spatial plausibility and temporal consistency.

To further enhance the spatiotemporal continuity and personalization of the generated trajectories, we employ an iterated DPO training pipeline. The training data is constructed from pairs of model-generated trajectories (from current version **GAMS**) and corresponding real user trajectories. We adapt DPO to the mobility generation setting by treating real human trajectories as preferred responses and synthetic ones as dispreferred. The loss implicitly learns a reward function that captures spatiotemporal regularities and individual behavioral patterns, guiding the model to generate more realistic and personalized trajectories without explicit reward modeling. The formula is as follows,

$$\mathcal{L}(\theta) = -\,\mathbb{E}_{(u,\tau_w,\tau_l)\sim\mathcal{D}} \left[\log\sigma\left(\beta\log\frac{\pi_\theta(\tau_w \mid u,\mathcal{K}_{\text{geo}},\mathcal{P}_{\text{mob}})}{\pi_{\text{ref}}(\tau_w \mid u,\mathcal{K}_{\text{geo}},\mathcal{P}_{\text{mob}})} - \beta\log\frac{\pi_\theta(\tau_l \mid u,\mathcal{K}_{\text{geo}},\mathcal{P}_{\text{mob}})}{\pi_{\text{ref}}(\tau_l \mid u,\mathcal{K}_{\text{geo}},\mathcal{P}_{\text{mob}})}\right)\right],$$

where u denotes the user profile, τ_w and τ_l are the preferred (e.g., real) and less preferred (e.g., synthetic) trajectories, respectively, $\mathcal{D}$ is the mixed trajectory dataset, π_θ is the trainable policy model, π_{ref} is the fixed reference policy, $\mathcal{K}_{\text{geo}}$ and $\mathcal{P}_{\text{mob}}$ represent urban geospatial knowledge and extracted mobility patterns, respectively, and β is the temperature coefficient controlling the regularization strength.

Based on the above formula, we implement a closed-loop workflow: training $\rightarrow$ deployment $\rightarrow$ testing $\rightarrow$ data collection $\rightarrow$ retraining, executed over multiple cycles. Through this progressive, multi-phase refinement, we continuously enrich the model's urban geographic understanding and strengthen its ability to capture individual mobility characteristics. Ultimately, this process enhances framework's spatiotemporal reasoning capabilities, enabling more realistic and personalized human mobility simulation.

4 Experiments

4.1 Experimental Setup

Datasets. We carry out experiment using two real-world mobility datasets, ChinaMobile and Tencent. The basic information of the datasets is shown in Table 2. To test **GAMS** 's performance on public datasets, we employ open street map's road network data and AOI data along with global POI data from Foursquare to jointly represent urban spaces. This does not compromise the overall experimental results. This confirms the transferability of **GAMS** across different datasets, and can achieve reasonably good performance even on smaller, lower-quality datasets.

Metrics. Following previous work (Feng et al., 2020; Shao et al., 2024), we evaluate the quality of generated mobility data from three dimensions, including statistical evaluation, aggregation evaluation and semantics evaluation. We also use Toponym Valid Ratio (TVR) to measure geographic knowledge hallucination, and Composite Mean Reciprocal Rank (CMRR) to measure overall performance across all metrics. Detailed metrics are described in appendix.

Table 1: Performance comparison of mobility simulation methods across datasets. Best and second-best results are highlighted in **bold** and underline, respectively.

Dataset	Model	Generation Metrics								CMRR↑
		FVLoc↓	ActProb↓	Distance↓	Radius↓	SI↓	SD↓	DARD↓	STVD↓	
Tencent	TimeGeo	0.3671	0.5915	0.3053	0.3267	0.2312	0.3414	0.6972	0.6899	0.1798
	ActSTD	0.3518	0.6972	0.2815	0.1007	0.6880	0.0907	0.6946	0.6371	0.2574
	DSTPP	0.3115	0.5236	**0.1599**	0.1589	0.6880	0.0308	0.6863	0.6742	0.4458
	MoveSim	0.3650	**0.2647**	0.3876	0.3192	0.2309	0.1716	0.4946	**0.5271**	0.4354
	CoPB	**0.2744**	0.3907	0.1997	0.2394	0.4131	0.1915	0.4089	0.5660	0.4146
	LLMob	0.5044	0.3723	0.5266	0.1723	**0.1618**	0.0752	0.4400	0.6639	0.3420
	GAMS	0.3213	0.2769	0.2596	**0.0517**	0.3362	**0.0457**	**0.3832**	0.6540	**0.5208**
ChinaMobile	TimeGeo	0.3825	0.6793	0.4009	0.5466	0.5139	0.3065	0.5762	0.6907	0.1940
	ActSTD	0.4025	0.6900	0.4577	0.3400	0.5749	**0.0452**	0.6891	0.6947	0.2673
	DSTPP	0.3895	0.6794	0.3991	0.2553	0.5320	0.0989	0.6774	0.6613	0.2229
	MoveSim	0.3776	0.2413	0.5117	0.3810	0.3720	0.1063	0.5264	**0.5775**	0.3479
	CoPB	**0.2724**	0.2097	0.2690	0.1948	0.4794	0.2282	0.4997	0.5817	0.4479
	LLMob	0.5130	0.4198	0.5387	0.1690	**0.1207**	0.0839	0.3976	0.6585	0.4003
	GAMS	0.2994	**0.1244**	**0.1867**	**0.0507**	0.2524	0.0473	**0.3544**	0.6776	**0.7125**

Table 2: Basic information of the trajectory datasets

Datasets	**Tencent**	**ChinaMobile**
Duration	2019.10.1 – 2019.12.31	2017.7.1 – 2017.8.31
City	Beijing	Beijing
#Users	100,000	1,246
#Trajectory Points	297,363,263	4,163,651

Methods. We evaluated our model in comparison with several state-of-the-art approaches, categorized into the following three groups: mechanistic models (TimeGeo (Jiang et al., 2016)), deep learning based methods (MoveSim (Feng et al., 2020), ActSTD (Yuan et al., 2022), DSTPP (Yuan et al., 2023)), LLM based methods (CoPB (Shao et al., 2024), LLMob (Wang et al., 2024a)).

4.2 Main Results

We want to validate the model's ability to generate geospatially accurate trajectories in real-world urban space without external geographic knowledge and with only limited user profile information. To ensure fair comparison, for LLM-based models, we employ llama3.1-8b as LLM core, while removing all specific location names (except anchor points) and manually extracted user-specific trajectory features from prompts; for deep-learning-based methods, we reduce the training-to-test ratio to 3:7 (which may slightly inflate their results because it is not feasible to completely remove the dependency on geographic knowledge from their frameworks).

The experiment results in Table 1 demonstrate that **GAMS** achieves better performance on 11 out of 18 metrics, with a particularly significant 41.81% improvement over the best baseline in key spatial metrics (Distance, Radius, SD), which can be attributed to its effective utilization of built-in urban spatial knowledge. Notably, **GAMS** achieves the highest CMRR-temporal (encompassing ActProb, STVD, DARD) and CMRR-spatial (encompassing SD, Distance, and Radius) on both datasets, outperforming all baselines. While some models excel on individual metrics, **GAMS** demonstrates a superior balance across all dimensions of mobility realism.

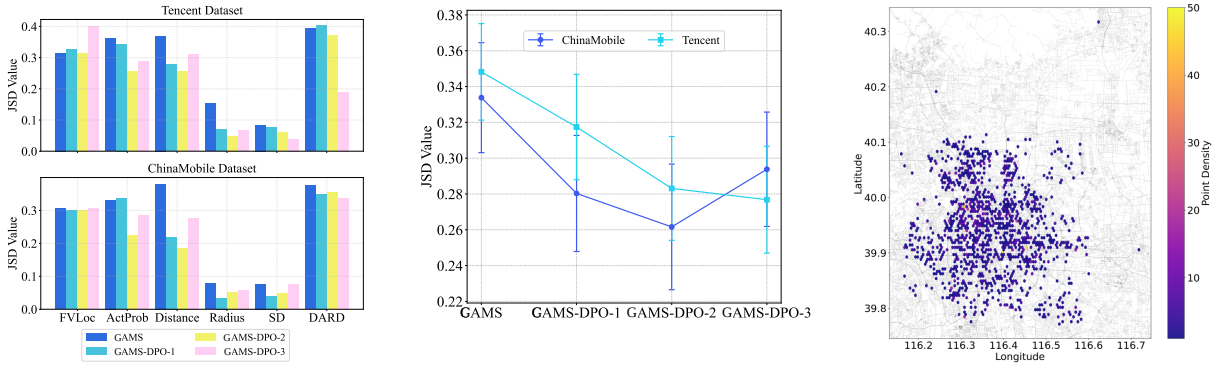
4.3 Ablation Studies

Impact of collective knowledge and individual knowledge. As we introduce in section 3.1 and section 3.2, we integrate individual knowledge through extracting mobility patterns in MobExtractor, while incorporating collective knowledge via introducing feedback in reflection stage of Anchor Location Extractor. By analysing experimental results in Table 8 and generation results in Figure 3, we find that the original implementation significantly outperforms both the individual-knowledge-ablated (w/o Individual) and collective-knowledge-ablated variants (w/o Collective), confirming that by integrating individual and collective knowledge, the model can more accurately understand the relationship between mobility patterns, specific user profiles and real-world urban spatial patterns, consequently generating trajectories that better align with user profile and actual urban mobility distributions.

Impact of Trajectory Style Alignment. We evaluate overall performance of trajectory enhancement module in Table 8. As visually confirmed in Figure 2b, there is an overall reduction in JSDs across successive DPO iterations, indicating that TrajEnhancer progressively enhances the spatiotemporal continuity of generated trajectories to approximate

Table 3: Performance comparison of different LLMs within the **GAMS** framework. Best and second-best results are highlighted in **bold** and underline, respectively.

Dataset	Base Model	Generation Metrics									CMRR↑
		FVLoc↓	ActProb↓	Distance↓	Radius↓	SI↓	SD↓	DARD↓	STVD↓	TVR↑	
Tencent	LLaMA3.1-8B	0.4315	0.4649	0.3109	0.0920	0.2751	0.0883	0.3810	0.6672	0.9570	0.1630
	LLaMA3-70B	0.4342	0.3102	0.2985	0.0529	0.2053	0.0460	**0.2896**	0.6573	0.9560	0.4260
	Qwen2-72B	0.4119	0.3421	0.4384	0.1028	0.2128	0.1072	0.3203	0.6601	0.8420	0.1829
	Qwen3-235B	0.4089	0.2738	0.3582	0.1015	0.2899	0.1725	0.3569	0.6416	0.8247	0.2639
	GPT-4o-mini	0.4119	**0.2499**	0.3753	0.1143	0.1874	0.1218	0.3046	0.6624	0.8672	0.3402
	Gemma3-27B	0.3994	0.3903	0.4160	0.0695	**0.1717**	0.0506	0.3265	0.6570	0.8252	0.3583
	Mistral-7Bv3	0.4089	0.3547	0.3064	0.0811	0.3101	0.0980	0.3574	0.6677	0.8854	0.2036
	CityGPT	0.4342	0.3849	0.3108	0.1062	0.3076	0.0574	0.3166	**0.6356**	0.9196	0.2961
	GAMS-LLM	**0.3213**	0.2769	**0.2596**	**0.0517**	0.3362	**0.0457**	0.3832	0.6540	**1.0000**	**0.6111**
ChinaMobile	LLaMA3.1-8B	0.3992	0.4434	0.4045	0.0640	0.2521	0.0731	0.3987	0.6788	0.9684	0.2088
	LLaMA3-70B	0.4059	0.4120	0.3452	0.0638	0.2031	0.0517	0.3372	0.6792	0.9650	0.3125
	Qwen2-72B	0.4027	0.4414	0.3770	0.0618	0.2164	0.0479	0.3525	0.6780	0.8535	0.2958
	Qwen3-235B	0.3992	0.5140	0.4254	0.0937	0.2976	0.0949	0.3791	0.6586	0.8228	0.2118
	GPT-4o-mini	0.4027	0.4493	0.4894	0.0714	0.2802	0.0630	0.3689	0.6644	0.7836	0.1736
	Gemma3-27B	0.3994	0.3903	0.4160	0.0695	**0.1717**	0.0506	**0.3265**	**0.6570**	0.8252	0.5470
	Mistral-7Bv3	0.4059	0.4344	0.3806	0.0913	0.2035	0.0822	0.3608	0.6821	0.9007	0.1859
	CityGPT	0.3992	0.4816	0.3153	0.0879	0.2448	0.0832	0.3646	0.6686	0.8725	0.2512
	GAMS-LLM	**0.2994**	**0.1244**	**0.1867**	**0.0507**	0.2524	**0.0473**	0.3544	0.6776	**1.0000**	**0.6991**



(a) Detailed results for various metrics. (b) Aggregated performance. (c) Trajectory visualization

Figure 2: Results of individual mobility regularity alignment results for TrajEnhancer across various metrics and DPO iterations.

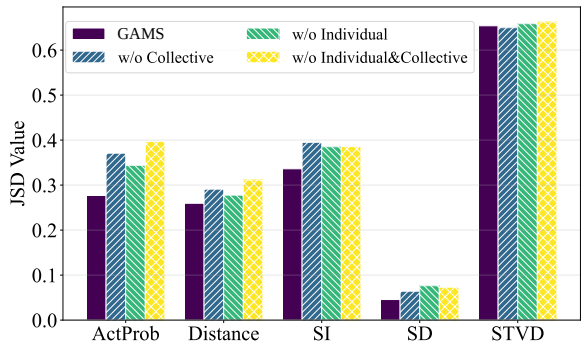


Figure 3: Ablation study on model designs (generation phase), where lower values indicate better performance.

real-world mobility patterns. Variations of each metric are visualized in Figure 2a.

Performance comparison between the enhanced geospaital LLM. We test the performance of multiple open-source and closed-source LLMs in experimental scenarios. The results in Table 3 demonstrate that CityGPT, a representative geospatial knowledge enhanced LLM, can provide more authentic and fine-grained urban geospatial knowledge compared to other larger-parameter models. Additionally, GAMS-LLM which used as the geospatial foundation model in our framework, achieves the highest CMRR, indicating its superior ability to capture the connections between user profiles, mobility patterns and geospatial knowledge.

5 Conclusion

In this paper, we propose **GAMS**, an agentic framework for generating realistic human mobility behavior trajectories. **GAMS** utilizes geospatial foundation model's inherent geospatial knowledge while incorporating advanced commonsense reasoning techniques to capture underlying movement patterns. Extensive experiments on real-world trajectory datasets demonstrate the framework's capability to directly generate realistic trajectories from new user profiles.

6 Limitations

First, despite the use of a geospatial foundation model for knowledge enhancement, the inherent hallucination tendency of LLMs may introduce spurious geographic facts; we mitigate this by combining exact matching with semantic similarity to handle long-tail cases.

Second, our evaluation relies on established real-world mobility datasets from prior literature, which inevitably constrain the scale and geographic diversity of simulated users.

Third, the integration of a large language model as a geospatial reasoning component incurs substantial computational overhead, as training requires approximately tens of hours on 4 A100 GPUs, limiting accessibility.

References

Pasquale Balsebre, Weiming Huang, and Gao Cong. 2024. Lamp: A language model on the map. *arXiv preprint arXiv:2403.09059*.

Ciro Beneduce, Bruno Lepri, and Massimiliano Luca. 2025. Large language models are zero-shot next location predictors. *IEEE Access*.

Prabin Bhandari, Antonios Anastasopoulos, and Dieter Pfoser. 2023. Are large language models geospatially knowledgeable? In *Proceedings of the 31st ACM International Conference on Advances in Geographic Information Systems*, pages 1–4.

Dirk Brockmann, Lars Hufnagel, and Theo Geisel. 2006. The scaling laws of human travel. *Nature*, 439(7075):462–465.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33:1877–1901.

Nicola Dempsey, Caroline Brown, Shibu Raman, Sergio Porta, Mike Jenks, Colin Jones, and Glen Bramley. 2010. Elements of urban form. *Dimensions of the sustainable city*, pages 21–51.

Jingtao Ding, Yunke Zhang, Yu Shang, Yuheng Zhang, Zefang Zong, Jie Feng, Yuan Yuan, Hongyuan Su, Nian Li, Nicholas Sukiennik, and 1 others. 2024. Understanding world or predicting future? a comprehensive survey of world models. *arXiv preprint arXiv:2411.14499*.

Yuwei Du, Jie Feng, Jie Zhao, and Yong Li. 2024. Trajagent: An agent framework for unified trajectory modelling. *arXiv preprint arXiv:2410.20445*.

Jie Feng, Yuwei Du, Jie Zhao, and Yong Li. 2025a. Agentmove: A large language model based agentic framework for zero-shot next location prediction. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1322–1338.

Jie Feng, Tianhui Liu, Yuwei Du, Siqi Guo, Yuming Lin, and Yong Li. 2025b. Citygpt: Empowering urban spatial cognition of large language models. *Proceedings of the 31th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Jie Feng, Zeyu Yang, Fengli Xu, Haisu Yu, Mudan Wang, and Yong Li. 2020. Learning to simulate human mobility. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 3426–3433.

Chen Gao, Xiaochong Lan, Nian Li, Yuan Yuan, Jingtao Ding, Zhilun Zhou, Fengli Xu, and Yong Li. 2024. Large language models empowered agent-based modeling and simulation: A survey and perspectives. *Humanities and Social Sciences Communications*, 11(1):1–24.

Chen Gao, Xiaochong Lan, Zhihong Lu, Jinzhu Mao, Jinghua Piao, Huandong Wang, Depeng Jin, and Yong Li. 2023. S3: Social-network simulation system with large language model-empowered agents. *arXiv preprint arXiv:2307.14984*.

Letian Gong, Yan Lin, Yiwen Lu, Xuedi Han, Yichen Liu, Shengnan Guo, Youfang Lin, Huaiyu Wan, and 1 others. 2024. Mobility-llm: Learning visiting intentions and travel preference from human mobility data with large language models. *Advances in Neural Information Processing Systems*, 37:36185–36217.

Wes Gurnee and Max Tegmark. 2024. Language models represent space and time. *ICLR*.

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):E5370–E5378.

Song Jiang, Da JU, Andrew Cohen, Sasha Mitts, Aaron Foss, Justine T Kao, Xian Li, and Yuandong Tian. 2024. Towards full delegation: Designing ideal agentic behaviors for travel planning. *arXiv preprint arXiv:2411.13904*.

Xuchuan Li, Fei Huang, Jianrong Lv, Zhixiong Xiao, Guolong Li, and Yang Yue. 2024. Be more real: Travel diary generation using llm agents and individual profiles. *arXiv preprint arXiv:2407.18932*.

Xiao Liang, Xudong Zheng, Weifeng Lv, Tongyu Zhu, and Ke Xu. 2012. The scaling of human mobility by taxis is exponential. *Physica A: Statistical Mechanics and its Applications*, 391(5):2135–2144.

Kang Liu, Xin Jin, Shifen Cheng, Song Gao, Ling Yin, and Feng Lu. 2024. Act2loc: a synthetic trajectory generation method by combining machine learning and mechanistic models. *International Journal of Geographical Information Science*, 38(3):407–431.

Qingyue Long, Huandong Wang, Tong Li, Lisi Huang, Kun Wang, Qiong Wu, Guangyu Li, Yanping Liang, Li Yu, and Yong Li. 2023. Practical synthetic human trajectories generation based on variational point processes. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 4561–4571.

Gengchen Mai, Weiming Huang, Jin Sun, Suhang Song, Deepak Mishra, Ninghao Liu, Song Gao, Tianming Liu, Gao Cong, Yingjie Hu, and 1 others. 2023. On the opportunities and challenges of foundation models for geospatial artificial intelligence. *arXiv preprint arXiv:2304.06798*.

Rohin Manvi, Samar Khanna, Marshall Burke, David Lobell, and Stefano Ermon. 2024a. Large language models are geographically biased. *arXiv preprint arXiv:2402.02680*.

Rohin Manvi, Samar Khanna, Gengchen Mai, Marshall Burke, David Lobell, and Stefano Ermon. 2024b. Geollm: Extracting geospatial knowledge from large language models. *ICLR*.

OpenAI. 2022. Introducing chatgpt. https://openai.com/blog/chatgpt/.

Luca Pappalardo, Ed Manley, Vedran Sekara, and Laura Alessandretti. 2023. Future directions in human mobility science. *Nature Computational Science*, 3(7):588–600.

Luca Pappalardo and Filippo Simini. 2018. Data-driven generation of spatio-temporal routines in human mobility. *Data Mining and Knowledge Discovery*, 32(3):787–829.

Jinghua Piao, Yuwei Yan, Jun Zhang, Nian Li, Junbo Yan, Xiaochong Lan, Zhihong Lu, Zhiheng Zheng, Jing Yi Wang, Di Zhou, and 1 others. 2025. Agentsociety: Large-scale simulation of llm-driven generative agents advances understanding of human behaviors and society. *arXiv preprint arXiv:2502.08691*.

Jonathan Roberts, Timo Lüddecke, Sowmen Das, Kai Han, and Samuel Albanie. 2023. Gpt4geo: How a language model sees the world's geography. *arXiv preprint arXiv:2306.00020*.

Camille Roth, Soong Moon Kang, Michael Batty, and Marc Barthélemy. 2011. Structure of urban movements: polycentric activity and entangled hierarchical flows. *PLOS ONE*, 6(1):e15923.

Chenyang Shao, Fengli Xu, Bingbing Fan, Jingtao Ding, Yuan Yuan, Meng Wang, and Yong Li. 2024. Chain-of-planned-behaviour workflow elicits few-shot mobility generation in llms. *arXiv e-prints*, pages arXiv–2402.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and 1 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Jianying Wang, Lei Dong, Ximeng Cheng, Weijun Yang, and Yu Liu. 2019. An extended exploration and preferential return model for human mobility simulation at individual and collective levels. *Physica A: Statistical Mechanics and Its Applications*, 534:121921.

Jiawei Wang, Renhe Jiang, Chuang Yang, Zengqing Wu, Ryosuke Shibasaki, Noboru Koshizuka, Chuan Xiao, and 1 others. 2024a. Large language models as urban residents: An llm agent framework for personal mobility generation. *Advances in Neural Information Processing Systems*, 37:124547–124574.

Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, and 1 others. 2024b. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6):186345.

Xinglei Wang, Meng Fang, Zichao Zeng, and Tao Cheng. 2023. Where would i go next? large language models as human mobility predictors. *arXiv preprint arXiv:2308.15197*.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, and 1 others. 2022. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*.

Fengli Xu, Qianyue Hao, Zefang Zong, Jingwei Wang, Yunke Zhang, Jingyi Wang, Xiaochong Lan, Jiahui Gong, Tianjian Ouyang, Fanjin Meng, and 1 others. 2025. Towards large reasoning models: A survey of reinforced reasoning with large language models. *arXiv preprint arXiv:2501.09686*.

Yuwei Yan, Qingbin Zeng, Zhiheng Zheng, Jingzhe Yuan, Jie Feng, Jun Zhang, Fengli Xu, and Yong Li. 2024. Opencity: A scalable platform to simulate urban activities with massive llm agents. *arXiv preprint arXiv:2410.21286*.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*.

Xiaoyan Yu, Tongxu Luo, Yifan Wei, Fangyu Lei, Yiming Huang, Hao Peng, and Liehuang Zhu. 2024. Neeko: Leveraging dynamic lora for efficient multi-character role-playing agent. *2024 Conference on Empirical Methods in Natural Language Processing*.

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*, pages 3173–3184.

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*, pages 4752–4762.

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):748–762.

A Appendices

A.1 Ethical considerations

We employ a LLM as an assistive tool to correct typographical errors and refine phrasing. All resulting modifications have been carefully reviewed and verified by the authors.

A.2 Comparison of embedding-based vs. language-based in MobExtractor

A comparison of the overall experimental results between the language-based and embedding-based methods is presented in Table 4. The results show that embedding-based method performed comparably to the language-based approach, demonstrating that embeddings effectively capture critical user profile details. Moreover, the embedding-based method is more cost-efficient and easier to generalize for large-scale experiments. Therefore, we employ the embedding-based method as our primary approach in the experiments.

A.3 Comparison of different methodologies in Urban Structure Mapper.

By comparing results of geoknowledge enhanced LLM, **GAMS** with map tools (**GAMS** -Map) and **GAMS** with social networks (**GAMS** -Social) in Figure 4, we find that enhanced CityGPT based **GAMS** outperforms other methods with visibly lower JSDs. This suggests that implicitly incorporating geographic knowledge in trajectory generation tasks is reasonable, and enhanced CityGPT offers greater advantages over traditional GIS tools and social relationships.

A.4 Detailed Metrics

- **Individual evaluation.** We calculate Jensen–Shannon Divergence (JSDs) on the following metrics of per user: Distance, Radius, Step Interval (SI), Step Distance (SD) and Spatial-temporal Visits Distribution (STVD).
- **Collective evaluation.** We evaluate the quality of all generated data from a collective perspective, calculating JSDs on following metric of all users: Frequently visited locations (FVLoc), which is defined as the overall distributions of top 40 most frequently visited locations across all users.
- **Semantics evaluation.** To evaluate the plausibility of generated mobility data, we map venue categories to user intents and compute JSDs on following category-related metrics at both individual and collective levels: Daily Activity Routine Distribution (DARD) and Activity Probability (ActProb).
- **Hallucination evaluation.** We define Toponym Valid Ratio (TVR), which is the ratio of valid generated toponyms to total generated toponyms, to assess the degree of hallucination in the model's candidate geospatial knowledge generation. A higher TVR signifies lower hallucination frequency and better generation quality.
- **Comprehensive evaluation.** To holistically assess model performance, we propose the Composite Mean Reciprocal Rank (CMRR) metric, computed through a two-stage process: (1) calculating the reciprocal rank of each metric relative to all comparable models, then (2) computing the arithmetic mean of these reciprocal ranks across all metrics. A higher CMRR indicates more consistently superior performance across all evaluation dimensions.

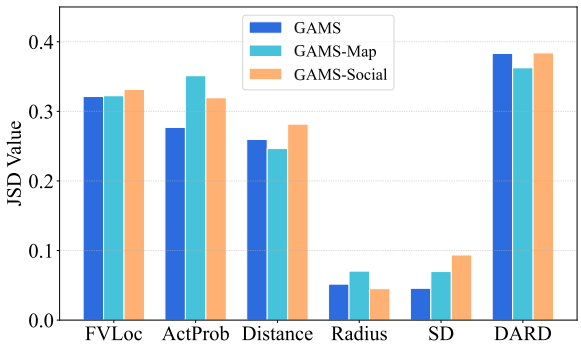


Figure 4: Methodological comparisons in GeoGenerator, where lower values indicate better performance. Here, '**GAMS**-Map' refers to the CityGPT with map tools, and '**GAMS**-Social' refers to the CityGPT with social graph integration.

A.5 Results on Mobility Recovery

To verify MobExtractor's efficacy in high-level mobility patterns extraction, we examine if template users can reconstruct their mobility patterns using extracted profile-specific semantic mobility pattern

Table 4: Performance comparison of embedding-based vs. language-based similarity methods in MobExtractor for trajectory generation. Best results for each metric are highlighted in **bold**.

Dataset	Similarity Method	Generation Metrics							
		FVLoc↓	Prob↓	Distance↓	Radius↓	SI↓	SD↓	DARD↓	STVD↓
Tencent	Embedding-based	0.3148	0.3611	**0.3687**	**0.1533**	**0.4566**	0.0845	**0.3931**	**0.6533**
	Language-based	**0.3027**	**0.3474**	0.3950	0.1674	0.4679	**0.0650**	0.3936	0.6583
ChinaMobile	Embedding-based	0.3118	**0.3872**	**0.2942**	**0.0691**	0.5054	0.0619	**0.3935**	**0.6400**
	Language-based	**0.2992**	0.4792	0.3701	0.0794	**0.4766**	**0.0616**	0.4156	0.6546

descriptions. We employ GeoGenerator to map mobility patterns to real trajectories in urban spaces. We compare the experimental results of our method with other trajectory generation methods. It should be noted that **GAMS** relies solely on user profiles as external input, whereas comparative methods explicitly incorporate additional external information such as geographical data. The results of evaluating performance are detailed in Table 5. The results demonstrate that **GAMS** achieves superior performance on 7 of 16 metrics in both data sets, with particularly outstanding advantages in metrics evaluating individual mobility capability (Radius) and behavioral habits (DARD). **GAMS** also exhibits commendable performance in terms of spatial continuity within trajectories (Distance and SD). This superior performance of **GAMS** can be attributed to its comprehensive consideration of the alignment between urban geographical knowledge and user mobility patterns.

A.6 Downstream Application: Epidemiological Simulation

To evaluate the practical realism and utility of the trajectories generated by our framework, we deploy them in a representative downstream task: simulation of infectious disease transmission. Specifically, we feed the synthetic mobility data produced by **GAMS** and a baseline method (ActSTD) into an epidemic propagation model that tracks the daily counts of Susceptible (S), Infectious (I), and Recovered (R) individuals. From multiple simulation runs, we compute the mean absolute percentage error (MAPE) across days to quantify fidelity relative to real-world epidemic dynamics. As shown in Table 6, **GAMS** explicitly accounts for both individual behavioral patterns and collective social influences, generating a denser and more realistic contact network. This enables the simulated epidemic curve to closely align with observed transmission patterns in real populations. In contrast, ActSTD largely neglects collective interaction effects, resulting in an extremely sparse contact network with isolated, linear transmission chains that are prone to premature termination. Consequently, its simulation outcomes exhibit excessive smoothness: daily population states (S, I, R) change very slowly and show high variance across runs, failing to capture the stochastic yet structured nature of real outbreaks. These results demonstrate that the trajectories generated by **GAMS** not only achieve higher statistical realism but also possess functional validity in complex, real-world decision-support scenarios such as public health modeling.

A.7 Data preprocessing

Trajectory data preprocessing Among the 100,000 users in the Tencent dataset, 44,313 users have both complete user profiles and accurate coordinates for their home and workplace. We conduct Collective Knowledge Extractor experiment using this subset of data. For each mobility dataset, we select the top 150 users with the highest average daily trajectory points for Geographic Knowledge Extraction, among which we randomly assign 100 users for trajectory recovery and the remaining 50 users for trajectory generation. We utilize the top 1,500 users with relatively higher average daily trajectory points for Individual Knowledge Extractor.

We aligned the private dataset with public datasets by replacing the private dataset's urban elements (including AOIs, POIs and roads) with relevant elements in OSM and Foursquare. Notably, while the aligned mobility dataset becomes sparser, this does not compromise the overall experimental results. This confirms the transferability of GAMS accross different datasets, and can achieve reasonably good performance even on smaller, lower-quality datasets.

DPO training data construction We construct the training dataset using the corpus output by GAMS. We employ Qwen2-72B, a large language model primarily trained on Chinese corpora, to extract mobility patterns while leveraging CityGPT

Table 5: Performance comparison of trajectory recovery methods across datasets.

Dataset	Model	Recovery Metrics								CMRR↑
		FVLoc↓	ActProb↓	Distance↓	Radius↓	SI↓	SD↓	DARD↓	STVD↓	
Tencent	TimeGeo	0.3555	0.6673	0.1506	0.2749	0.2010	0.3362	0.6895	0.6921	0.1890
	ActSTD	0.2007	0.6737	**0.0841**	0.0581	0.6880	0.0606	0.6910	0.6746	0.3494
	DSTPP	0.1975	0.3582	0.1346	0.2611	0.3960	0.0775	0.4824	0.4233	0.2792
	MoveSim	0.1966	0.1647	0.3176	0.3189	0.2009	0.1315	0.4456	**0.4141**	0.4045
	CoPB	0.1981	0.3592	0.1322	0.2270	0.3858	0.1885	0.3930	0.5474	0.3000
	LLMob	**0.1213**	**0.0891**	0.1822	0.1330	**0.1618**	0.0630	0.3961	0.5923	**0.5563**
	GAMS	0.3421	0.2444	0.2087	**0.0340**	0.2945	**0.0443**	**0.2681**	0.6284	0.5146
ChinaMobile	TimeGeo	0.3766	0.6862	0.3779	0.2908	0.4517	0.3057	0.6874	0.6845	0.1619
	ActSTD	0.2985	0.6862	0.4336	0.2816	0.5321	**0.0324**	0.6905	0.6907	0.2631
	DSTPP	0.2136	0.3722	0.1570	0.2705	0.2941	0.1537	0.5692	**0.4350**	0.3708
	MoveSim	0.1776	**0.1496**	0.3300	0.2619	0.4514	0.0950	0.4346	0.4617	0.4104
	CoPB	0.2677	0.2287	0.1415	0.1948	0.4735	0.1945	0.4790	0.5474	0.2917
	LLMob	**0.1560**	0.1590	0.1654	0.0763	**0.1207**	0.0718	0.3795	0.6560	0.5417
	GAMS	0.3055	0.3664	**0.1339**	**0.0717**	0.3464	0.0484	**0.3248**	0.6601	**0.5563**

Table 6: Comparison of epidemic simulation outcomes (averaged over multiple runs). Values denote the mean percentage of individuals in each compartment: Susceptible (S), Infectious (I), and Recovered (R).

Method	S (%)	I (%)	R (%)
GAMS	13.23	40.17	30.32
ActSTD	51.71	53.65	52.76

for urban geographical knowledge. Our experimental setup involves 300 training users and 1200 test users, with all textual outputs from GAMS being collected for analysis. Using Qwen2.5-72B, we evaluate the quality of mobility pattern extraction from this corpus by assigning quality scores on a 0-10 scale, where we select textual outputs scoring above 5 as negative samples while using the corresponding individuals' real trajectories as positive samples for our training data construction.

A.8 Ablation studies

The hierarchical address representation is designed to activate geographic knowledge in CityGPT by associating location names with their attributes across defined regional hierarchies, resulting in generating more accurate locations with reduced hallucinations. In comparison, the human-centric address representation directly prompt the model to recall neighborhood geographic information from the training corpus. Results in Table 9 demonstrate that the second one performs better, likely because training corpus of CityGPT has been pre-aligned with OSM data.

A.9 Examples

Here, we present the detailed prompt for the *Geo-generator* module.

```
###### Explore Geographical Knowledge

You are an urban resident planning your next
    point of interest (POI) to visit based on
    current <INTENT> and <LOCATION>.

<INTENT>

You are going to eat, so the next POI should
    belong to:
{POI_categories}.

<LOCATION>

Your current location is {POI_name}, and the
    next POI should be within 3 km of your
    current location.
Please list five POIs you are most likely to
    visit next.

###### Utilize Geographical Knowledge

Act as a person in an urban neighborhood. Think
    about your daily routine and generate a plan
     for a weekday (from 0:00 to 24:00),
    explaining it. Follow these principles:

1. **Retain key features**:
   Consider important features in <TRAJECTORY_
    DESCRIPTION> and retain them in your plan.
    Refer to <IMPORTANT_FEATURES>.

2. **Supplement sparse data**:
   Use general knowledge (e.g., shortest-path
    algorithms) to enrich sparse information in
    <TRAJECTORY_DESCRIPTION>. Refer to <GENERAL_
    INFORMATION>.

3. **Incorporate high-level patterns**:
   Reflect movement patterns, travel habits, and
     user motivations. Refer to <HIGH_LEVEL_
    FEATURES>.

4. **Select POIs**:
   Choose one POI per time period from <
    CANDIDATE_POIS>.

<TRAJECTORY_DESCRIPTION>

{trajectory_description}
```

Table 7: Performance comparison of different LLMs in trajectory recovery within the GAMS framework.

Dataset	Base Model	Recovery								
		FVLoc↓	ActProb↓	Distance↓	Radius↓	SI↓	SD↓	DARD↓	STVD↓	TVR↑
Tencent	LLaMA3.1-8B	0.3468	0.2776	0.1913	0.0362	0.3070	0.0794	0.3300	0.6774	0.9697
	LLaMA3-70B	0.4468	0.3180	**0.1770**	0.0529	0.1697	**0.0383**	0.2684	0.6310	0.9541
	Qwen2-72B	0.4421	0.3567	0.2177	0.0785	0.1682	0.0881	0.2848	**0.6091**	0.8224
	Qwen3-235B	0.4468	0.3152	0.1646	0.0687	0.2819	0.0895	0.3255	0.6178	0.9040
	GPT-4o-mini	0.4468	0.3682	0.2737	0.1022	0.1650	0.1038	**0.2628**	0.6170	0.7844
	Gemma3-27B	0.4468	0.3134	0.3243	0.0887	**0.1637**	0.0964	0.2789	0.5967	0.8477
	Mistral7Bv3	0.4468	0.2799	0.2273	0.0639	0.2030	0.0865	0.2886	0.6265	0.9035
	GAMS	**0.3421**	**0.2444**	0.2087	**0.0340**	0.2945	0.0443	0.2681	0.6284	**1.0000**
ChinaMobile	LLaMA3.1-8B	0.3176	0.3964	0.2930	0.0987	0.2400	0.0612	0.3760	0.6769	0.9815
	LLaMA3-70B	0.4118	0.3826	0.2746	0.0753	0.1953	0.0501	0.3333	0.6615	0.9690
	Qwen2-72B	0.4118	0.4025	0.3616	0.1050	0.2190	0.0864	0.3373	0.6535	0.8306
	Qwen3-235B	0.4118	0.4242	0.2852	0.0770	0.3615	0.0564	0.4550	0.6595	0.9310
	GPT-4o-mini	0.4118	0.4666	0.3737	0.1285	0.2175	0.0803	0.3276	**0.6468**	0.8561
	Gemma3-27B	0.4118	0.4752	0.3390	0.0871	**0.1684**	0.0883	0.3346	0.6480	0.8436
	Mistral7Bv3	0.4118	0.5327	0.2345	0.0876	0.2598	0.0665	0.3582	0.6488	0.9164
	GAMS	**0.3055**	**0.3664**	**0.1339**	**0.0717**	0.3464	**0.0484**	**0.3248**	0.6601	**1.0000**

Table 8: Performance comparison of different methodology variants in UrbanMapper.

Dataset	Variant	Recovery Metrics							
		FVLoc↓	ActProb↓	Distance↓	Radius↓	SI↓	SD↓	DARD↓	STVD↓
Tencent	GAMS	**0.3421**	0.2444	0.2087	0.0340	**0.2945**	0.0443	**0.2681**	0.6284
	GAMS w/o C	0.3761	0.2721	0.2285	0.0688	0.3431	0.0614	0.2814	0.6365
	GAMS w/o I	0.3421	0.3721	0.1995	0.0267	0.2985	0.0229	0.3221	**0.6219**
	GAMS w/o I&C	0.3421	0.3887	**0.1563**	0.0660	0.3342	0.0377	0.3166	0.6448
	GAMS-M	0.3680	0.2458	0.1838	0.0266	0.3215	0.0334	0.3066	0.6250
	GAMS-M w/o C	0.3421	0.3108	0.1828	0.0790	0.2963	0.0390	0.3271	0.6478
	GAMS-M w/o I	0.3468	0.3248	0.1735	0.0469	0.2722	**0.0330**	0.3150	0.6265
	GAMS-M w/o I&C	0.3568	0.3794	0.1746	0.0305	0.3147	0.0467	0.2937	0.6283
	GAMS-S	0.3421	**0.2136**	0.1699	**0.0195**	0.3790	0.0378	0.3161	0.6301
	GAMS-S w/o C	0.3421	0.3240	0.2058	0.0412	0.3001	0.0435	0.3178	0.6269
	GAMS-S w/o I	0.3421	0.3504	0.1655	0.0596	0.2655	0.0513	0.3115	0.6259
	GAMS-S w/o I&C	0.3468	0.3518	0.1783	0.0790	0.3202	0.0543	0.3151	0.6356
ChinaMobile	GAMS	**0.3055**	**0.3664**	**0.1339**	0.0717	0.3464	0.0484	**0.3248**	0.6601
	GAMS w/o C	0.3055	0.3784	0.2162	0.0975	0.3330	0.0592	0.3393	0.6753
	GAMS w/o I	0.3118	0.3771	0.1956	0.0676	0.3207	0.0451	0.3693	**0.6437**
	GAMS w/o I&C	0.3118	0.3742	0.2117	0.0654	0.3500	0.0490	0.3792	0.6501
	GAMS-M	0.3055	0.3780	0.1468	**0.0340**	0.3106	0.0633	0.3526	0.6550
	GAMS-M w/o C	0.3118	0.3971	0.1934	0.0704	0.3482	0.0532	0.3592	0.6594
	GAMS-M w/o I	0.3055	0.4141	0.2122	0.0454	0.3198	**0.0396**	0.3574	0.6526
	GAMS-M w/o I&C	0.3118	0.4100	0.1846	0.0874	0.3643	0.0796	0.3336	0.6546
	GAMS-S	0.3055	0.3687	0.1669	0.0601	**0.2896**	0.0578	0.3409	0.6622
	GAMS-S w/o C	0.3055	0.3869	0.1795	0.0775	0.2973	0.0567	0.3427	0.6643
	GAMS-S w/o I	0.3118	0.4006	0.1713	0.0377	0.3081	0.0643	0.3717	0.6572
	GAMS-S w/o I&C	0.3055	0.3926	0.2303	0.0871	0.3371	0.0708	0.3766	0.6524

Table 9: Performance comparison of human vs. hierarchical address types in recovery and generation tasks. Best results for each metric are highlighted in **bold**.

Dataset	Task	Address Type	Performance Metrics								
			Loc↓	Prob↓	Dist↓	Rad↓	SI↓	DailyLoc↓	SD↓	DARD↓	STVD↓
Tencent	Recovery	Human	**0.3421**	**0.3383**	**0.2728**	**0.1212**	**0.4479**	0.1803	**0.0578**	**0.3849**	**0.6391**
		Hierarchical	0.3420	0.3117	0.2852	0.1004	0.4639	**0.1397**	0.0779	0.4099	0.6336
	Generation	Human	**0.3148**	**0.3611**	**0.3687**	0.1533	**0.4566**	**0.1687**	**0.0845**	**0.3931**	**0.6533**
		Hierarchical	0.3089	0.3840	0.4577	**0.1283**	0.4690	0.1975	0.1114	0.4092	0.6512
ChinaMobile	Recovery	Human	0.3118	**0.3872**	0.2942	**0.0691**	**0.3054**	0.1217	0.0619	**0.3935**	**0.6400**
		Hierarchical	0.3118	0.4648	**0.2828**	0.0986	0.4945	**0.0379**	**0.0587**	0.4143	0.6425
	Generation	Human	0.3059	**0.3317**	0.3807	0.0778	**0.4624**	**0.0980**	**0.0767**	**0.3757**	0.6591
		Hierarchical	**0.2992**	0.3586	**0.3251**	**0.0767**	0.4789	0.1497	0.0909	0.3888	**0.6614**

```
<IMPORTANT_FEATURES>

{important_features_of_trajectory}

<GENERAL_INFORMATION>

{general_information_of_trajectory}

<HIGH_LEVEL_FEATURES>

{high_level_features_of_trajectory}

<CANDIDATE_POIS>

{candidate_pois}
```

A.10 Visualization of Generated Trajectory Data

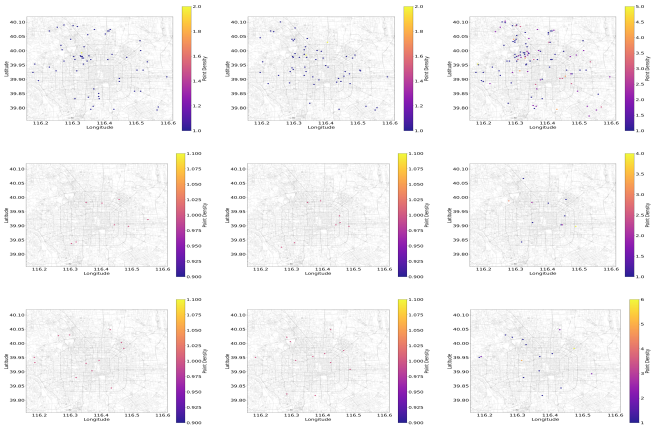
To demonstrate that **GAMS** effectively captures the relationships between user profiles, mobility patterns, and trajectories in real urban spaces, we visualized the anchor points and single-day trajectory point distributions for different user profiles generated by the model in Figure 6.

A.11 Generated mobility patterns of different user profiles

We compared the distributions of anchor points and trajectory points generated by the model for three types of user profiles in Figure 6. The first row in the left panel represents to the user profiles of 20-year-old, moderately-income male with a bachelor's degree working as an IT engineer(Profile 0). Generated work locations (Figure 5b)of this group show a higher degree of clustering in city centers compared to their residential locations(Figure 5a), and the overall trajectory point distribution is relatively dispersed(Figure 5c), suggesting that they tend to reside in suburban areas to reduce living costs, while working in IT-related companies concentrated in urban centers. The second row corresponds to young females in Finance/Accounting/Auditing/Tax/Cashier with a bachelor's degree and a moderate income(Profile 1). Generated workplaces and residences of Profile 1 cluster around major corporate hubs, suggesting that they often work in larger companies and rent apartments nearby their workplaces. The third row represents middle-aged males in Network Sales/Operations/Services with a bachelor's degree and a moderate income(Profile 2).Compared to the previous two user types, the trajectories generated for Profile 2 appear more scattered and irregular, demonstrating less regular mobility patterns, as they may travel more frequently for business.

A.12 Baselines

- **TimeGeo**(Jiang et al., 2016): It uses statistical methods to model temporal patterns while leveraging the r-EPR mechanism to model spatial patterns of user mobility data.
- **ActSTD**(Yuan et al., 2022): It adopts a GAIL framework, combining continuous spatio-temporal dynamics modeling with generative adversarial training to generate mobility data.
- **DSTPP**(Yuan et al., 2023): It employs a diffusion model to learn the joint spatio-temporal distribution, incorporating a co-attention module for modeling spatio-temporal point processes.
- **MoveSim**(Feng et al., 2020): It adopts a generative-adversarial framework in which the generator utilizes a self-attention-based sequence modeling network to capture temporal transitions and the discriminator distinguishes synthetic trajectories by incorporating key mobility patterns.
- **CoPB**(Shao et al., 2024): It leverages LLM to infer mobility-related habits and motivations from user profiles, then apply mechanistic models to map these patterns to real urban spaces.
- **LLMob**(Wang et al., 2024a): It derives mobility patterns from predefined trajectory features using LLM, then selects positive/negative training samples from historical mobility data for adversarial learning.



(a) Home (b) Workplace (c) Trajectory

Figure 6: Comparison of Mobility patterns in real urban spaces of different user profiles.

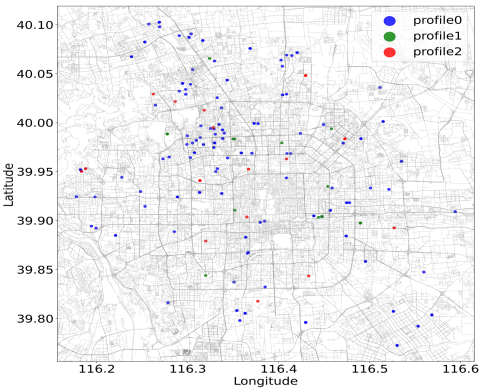


Figure 5: Comparison of daily movement patterns