Generating Realistic Human Mobility Data with Hybrid Large Language Model Agent

Chenyang Shao[1], Bingbing Fan[1], Jingtao Ding[1], Yuan Yuan[1], Meng Wang[2], Fengli Xu[1,*]
[1]Department of Electronic Engineering, BNRist, Tsinghua University
[2]School of Computer Science and Information Engineering, Hefei University of Technology
[*]Corresponding author, fenglixu@tsinghua.edu.cn

Abstract

Mobility data is crucial for understanding human dynamics and facilitating many applications, but collecting them is often expensive and privacy sensitive. Previous efforts on mobiliy data generation focus on fitting the statistical distribution and often overlook the underlying intentions behind mobility behavior. Recent breakthrough in large language models (LLMs) shows emergent capabilities of human-like reasoning, inspiring us to harness their power for intention-aware mobility generation. However, this problem is non-trivial since LLMs are not originally optimized for inferring human mobility behavior. To bridge this gap, we propose a novel data generator called Chain-of-Planned-Behavior (CoPB) that consists of 3 key components: (1) **a behavior science-inspired agentic workflow** that adapts LLMs for mobility intention reasoning. We draw on the *Theory of Planned Behavior* to integrate the factors of *attitude*, *subjective norms*, and *perceived behavioral control* to jointly decide the next mobility intention. (2) **a conditional generative diffusion model** that grounds the abstract intents into physical space by generating behavior intention-conditioned mobility trajectory. (3) **a lightweight data generator solution** that implements a cost-effective alternative data generator. Specifically, we use knowledge distillation to transfer the intent reasoning capability of LLMs to smaller language models (SLMs). We also explore the potential of substituting diffusion model with physics-informed mechanistic model. Experiments on three real-world datasets show that our generated data exhibits a strong alignment with real data, substantially reducing the error rate of intent reasoning by an average of 59.8%, and improving the average spatial metrics by 18.5%. Moreover, the lightweight data generator allow us to remove the API costs and reduces the generation time by 28.8% without significant performance loss. Both the generator and the accompanying constructed dataset are publicly available at https://github.com/PLUTO-SCY/CoPB.

Keywords

Mobility Data Generation, LLM Agent, Diffusion Model

ACM Reference Format:
Chenyang Shao[1], Bingbing Fan[1], Jingtao Ding[1], Yuan Yuan[1], Meng Wang[2], Fengli Xu[1,*]. 2018. Generating Realistic Human Mobility Data with Hybrid Large Language Model Agent. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation emai (Conference acronym 'XX).* ACM, New York, NY, USA, 17 pages. https://doi.org/XXXXXXX.XXXXXXX

Conference acronym 'XX, June 03–05, 2018, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/18/06
https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

Human mobility data is an important aspect of human activities, often revealing the nuanced details of human's spatio-temporal interaction with the physical environment [10]. Acquisition of mobility data plays a crucial role in numerous applications, such as urban planning [27], epidemic control [6] and business site selection [14]. Traditionally, large-scale human mobility data is collected through household travel survey [22] or ubiquitous personal sensors like smartphone [35], which can be hugely expensive [11] and are associated with serious privacy risks [7, 28]. Therefore, it has been a long-standing research quest to design low-cost yet high-quality generators for mobility data. [8, 21].

Existing work can be categorized into mechanistic models and deep generative models. Mechanistic models employ the stochastic processes to describe the location transitions and use few parameters to model decision strategies [5, 21]. But they are often too simplistic to accurately capture the features of data. With the advancement of deep generative networks, GANs [23, 33, 34], VAEs [15], and Diffusion models [37, 38] have been adapted to handle trajectory data structures for mobility generation [16]. However, these methods primarily focus on fitting statistical distribution and often overlook the underlying coherent intentions behind mobility behavior, leading to the generated data lacking realism, which makes them difficult to adapt flexibly to diverse application scenarios.

Recent advances in large language models (LLMs) have heralded fresh prospects for the domain of mobility generation. Scaled-up versions of LLMs such as GPT [9] and PaLM [3] have shown emergent abilities for commonsense reasoning [25]. Research shows that guiding LLMs with well-designed prompts enhances their reasoning power [26, 31], and they also excel in role-playing, simulating individuals with specific profiles [18]. These advancements and explorations seem to foreshadow the immense potential of deploying LLMs in mobility behavior generation, yet they are accompanied by three challenges. **First**, LLMs are trained on extensive text corpora to acquire a broad knowledge base and general reasoning abilities. But they have limited understanding of the cognitive mechanisms behind human behaviors, thus cannot directly generate satisfactory reasoning results. **Second**, LLMs have a restricted grasp of geographic information and struggle with spatial proximity, even when given POI locations. Their context window and attention capacity are also not enough to handle the vast amount of POIs in a city. **Finally**, using LLMs is extremely costly, making large-scale applications impractical. Calculations show that generating one mobility behavior trajectory with GPT-4-turbo costs about $0.80, which is too expensive for scenarios which require generating thousands of trajectories.

To comprehensively address these challenges, we design a novel mobility data generator, **Chain-of-Planned-Behaviours (CoPB)**. To tackle the first one, we draw insights from the *Theory of Planned Behaviour* [2] and use three intertwined factors, *attitudes*, *subjective norms*, and *perceived behavioral control*, to construct **a behaviour science-inspired agentic workflow**. We decompose the complete mobility behaviour simulation into a reasoning chain, applying the theory at each reasoning step to recursively generate the next intention, with the factors dynamically updated at each step. To address the second challenge, we propose **a conditional generative diffusion model** that uses intention sequences generated by an LLM as conditions for denoising, generating physical trajectories that align with the intentions, and then mapping them to the nearest POI based on the intention type. To tackle the third challenge, we develop **a lightweight data generator solution** that implements a cost-effective alternative generator within the existing framework. Specifically, we use knowledge distillation to transfer the intent reasoning capability of LLMs to smaller language models (SLMs), significantly reducing the cost of LLM inference. We also explore the potential of substituting the diffusion model with a physics-informed mechanistic model which yields lower generation costs.

Experiments on three real-world datasets demonstrate that our data generator can generate intention-aware mobility data with high fidelity. The design of agentic workflow significantly enhances LLM's ability to perform human-like behaviour reasoning, reducing the error rate of intent reasoning by an average of 59.8%. Conditioned on the intent sequences from the LLM, trajectories generated by diffusion model align very well with the real data, improving the spatial metrics by an average of 18.5%. After knowledge distilling, SLM can achieve performance comparable to LLM on mobility reasoning, while local deployment reduces API costs to zero and decreases time costs by 25.2%. The application of the mechanistic model also results in a 40.5% increase in the efficiency of physical trajectory generation. Moreover, our experiments demonstrate that the generated data not only provides valuable utility in real-world applications but also effectively mitigates potential privacy risks. Leveraging CoPB, we construct **a simulation dataset** comprising 200,000 trajectories and a total of 794,820 points enriched with mobility intention, which has been open-sourced on our GitHub repository. Our contributions can be summarized as follows:

- We propose an intention-aware mobility data generator, which includes a behavior science-inspired agentic workflow for generating intention sequences and a conditional generative diffusion model for generating physical trajectories. The lightweight solution further extends the applicability of our generator in cost-constrained scenarios.
- Experiments conducted on three real-world datasets show that the generated trajectories align well with the real trajectories. Based on CoPB, we further construct and open-source a simulation dataset.
- Further experiments demonstrate that the generated data holds practical application value and is free from privacy risks, thereby laying a solid foundation for the widespread application of our data generator.

2 Related Work

Mobility Data Generation with Deep Generative Model. Deep generative models have been widely applied to human trajectory generation problems. Feng et al. [8] incorporate an attention-based region network into the generator of an adversarial network to introduce prior information about urban structures. Long et al. [15] design two VAEs to model the locations of users' homes and workplaces as well as their temporal patterns. Zhu et al. [38] embed extra pattern information in the reverse denoising process in the diffusion model to improve the fidelity of trajectories. Similarly using diffusion, SynMob [37] employs a classifier-free diffusion guidance method to generate diverse synthetic simulation data and, based on this, proposes a mobility trajectory benchmark.

However, these methods often overlook the coherent behavioral intentions of humans behind the physical trajectories. While they are precise in data distribution, the generated trajectories lack semantics and appear unrealistic, thus limiting their applicability.

LLM for Behaviour Simulation. LLMs have increasingly shown their potential in the domain of behavioural generation, marking an innovative leap in how behavioural data can be acquired. Recent works [1, 4, 18–20] demonstrate the ability of LLMs to generate complex, nuanced survey questions and scenarios, potentially replacing traditional methods of survey creation and administration. At the heart of this capability is the LLMs' advanced role-play capability, which enables them to assume the perspectives of various respondents, thereby generating more dynamic and engaging survey content. This function not only allows for the creation of diverse and tailored survey materials but also opens the door to understanding intricate behavioural patterns through simulated interactions. LLMob [24] is precisely this type of approach that uses LLMs to simulate human trajectories. However, it relies heavily on past trajectories and cannot generalize to new locations that are not seen in historical trajectories.

3 Preliminaries

3.1 Problem Definition

Mobility Data Generation. A piece of mobility data consists of a sequence of spatiotemporal points, denoted by $x = \{s_1, s_2, ..., s_n\}$. Each point s_i can be expressed as (t_i, l_i), where l_i is the spatial position and t_i is the time duration. Besides, we use e_i to represent the intention of visiting s_i. Given a real-world mobility dataset, the objective of the data generator is to simulate new mobility data which exhibit similar characteristics and mobility patterns as the original ones.

3.2 LLM Reasoning

Let's use M to denote a pre-trained LLM. The LLM follows an autoregressive generation paradigm, where each token is generated sequentially based on the previously generated tokens. Specifically, at each step, the model predicts the next token $\hat{y}_t$ conditioned on the preceding tokens, such that: $p(\hat{y}_t | y_1, y_2, \ldots, y_{t-1}) = M(y_1, y_2, \ldots, y_{t-1})$, where $\hat{y}_t$ is the predicted token at time step t, and $y_1, y_2, \ldots, y_{t-1}$ are the previously generated tokens. For simplicity, we can use the formula $\alpha \sim M(\beta)$ to represent the process where the LLM reasons and generates α given the input prompts β.

3.3 Diffusion Model Training and Sampling

Training Process In the training process of a diffusion model, the goal is to minimize a loss function that enables the model to recover the original data from noise. Let the data distribution be p_{data}, and the forward diffusion process be $q(x_t|x_{t-1})$, where x_t represents the data at time step t after t steps of noise diffusion, and x_0 is the original data. The objective of training is to optimize the following loss function:

$$\mathcal{L}_{\text{train}} = \mathbb{E}_{q(x_0),q(\mathbf{z})}\left[\|\epsilon_\theta(x_t,t)-\epsilon\|^2\right] \tag{1}$$

where $\epsilon_\theta(x_t,t)$ is the model's estimate of the noise at time step t, ϵ is the true noise, and θ represents the model parameters. This loss function measures the discrepancy between the model's predicted noise and the actual noise.

Sampling Process During the sampling process, we begin with noise x_T and iteratively reverse the diffusion process to recover the original data. The reverse process is modeled by $p_\theta(x_{t-1}|x_t)$ and is performed according to the following formula:

$$x_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(x_t,t)\right) + \sigma_t\mathbf{z} \tag{2}$$

where α_t and β_t are parameters controlling the noise diffusion at each timestep, $\bar{\alpha}_t$ is the accumulated diffusion coefficient, and $\mathbf{z}$ is standard Gaussian noise.

4 Methodology

This section introduces the framework of our proposed mobility data generator. We will first introduce **the behavior science-driven agentic workflow** that incorporates the *Theory of Planned Behaviour* into behavioral reasoning to generate intention sequences. (Figure 1 Part ①) Next, we present **the conditional generative diffusion model** that generates physical trajectories conditioned on the intention sequences. (Figure 1 Part ②) Finally, we will provide a detailed explanation of **the lightweight data generator solution** that offers cost-effective alternatives for both the LLM and the diffusion model. (Figure 1 Part ③)

4.1 Behavior Science-Driven Agentic Workflow

The *Theory of Planned Behaviour* [2] was developed by Icek Ajzen as a framework to comprehend and forecast behaviour across a wide range of behaviour categories in behavioural science, suggesting that actions are directly influenced by behavioural intentions and the intentions are shaped by three elements: *attitudes toward behaviours, subjective norms, and perceived behavioural control*:

- ***Attitudes toward behaviours*** refer to an individual's positive or negative feelings toward intentions, which encompass numerous subjective perspectives, intricately intertwined with the individual's preferences and are closely linked to character attributes, denoted as: $\alpha = \langle Preference \rangle : [preference_1, preference_2, ...]$
- ***Subjective norms*** involve individuals' perceptions of whether they should engage in a particular behaviour, influenced by societal moral standards or the viewpoints of significant others such as family, which may pressure individuals to follow expected norms or long-term planning criteria. In our problem, they are set as routines, which can be represented as: $\beta = \langle Routine \rangle : [routine_1, routine_2, ...]$
- ***Perceived behavioural control*** refers to the extent to which an individual believes they can complete a specific behaviour successfully, influenced by current conditions and obstacles encountered, e.g., the distance to the destination and the current time, which can be denoted as: $\gamma = \langle Perceived\ likelihood \rangle : [intention_1 : likelihood_1, intention_2 : likelihood_2, ...]$

Prompts for the agentic workflow include two parts: the demographic profile and a few manually annotated intent sequences as few-shot examples. We use $\mathbf{u}$ to represent these two collectively. We break down the mobility intention inference into step-by-step reasoning. At each step, based on the input $\mathbf{u}$ and the preceding intention sequence $\mathbf{v_i} = \{(t_0, e_0), ..., (t_i, e_i)\}$, we let the LLM infer $e_{i+1} \sim M(\mathbf{v_i}, \mathbf{u})$, which is then utilized to obtain $t_{i+1} \sim M(\mathbf{v_i}, \mathbf{u}, e_{i+1})$, iterating in this manner until generation completes.

As shown in Figure 1 Part ①, for each reasoning step $e_{i+1} \sim M(\mathbf{v_i}, \mathbf{u})$, we integrate the *Theory of Planned Behaviour* into generation. First, based on the profile (p), we ask about daily preferences (α). Then, also based on the profile, we ask LLM about routines in life (β). Finally, combining profile, generated preferences, routines with the preceding generation results, we ask LLM to dynamically assess the perceived likelihood (γ_i) of the next intention, which needs to be updated at each generation step. Finally, we combine $p, \alpha, \beta, \gamma_i$ as inputs to generate e_{i+1}. Prompts design is denoted as `prompt` and the whole reasoning process can be represented as:

$$\begin{gathered}\alpha \sim M(\texttt{prompt}_\alpha(p)) \quad \beta \sim M(\texttt{prompt}_\beta(p)) \\ \gamma_i \sim M(\texttt{prompt}_\gamma(p, \mathbf{v_i}, \alpha, \beta))\end{gathered} \tag{3}$$

$$e_{i+1} \sim M(\texttt{prompt}_\text{e}(\mathbf{v_i}, \alpha, \beta, \gamma_i)) \tag{4}$$

4.2 Conditional Generative Diffusion Model

LLMs struggle with mapping abstract intentions to real-world locations due to their limited understanding of urban space. While including POI information in prompts can help, LLMs still have difficulty grasping distances and spatial relationships, and the constantly changed surrounding POIs as individuals move further complicate intention grounding. To address this limitation, we propose a conditional generative diffusion model to ground the intents into physical space.

Diffusion models effectively capture complex patterns and high-frequency details in data distributions by gradually introducing noise and learning the reverse denoising process. This characteristic makes them particularly well-suited for trajectory generation tasks, as trajectory data often exhibits high non-linearity and complex spatial dependencies. Through a multi-step denoising process, diffusion models can progressively refine the generated results, enabling fine-grained modeling of trajectories and accurately capturing spatial details such as turning points, stop areas, and local variations in paths. Moreover, diffusion models possess strong conditional generation capabilities, allowing the generation process to be flexibly guided by conditional inputs (such as mobility intentions), ensuring that the generated trajectories not only conform to geographic constraints but also reflect specific behavioral patterns.

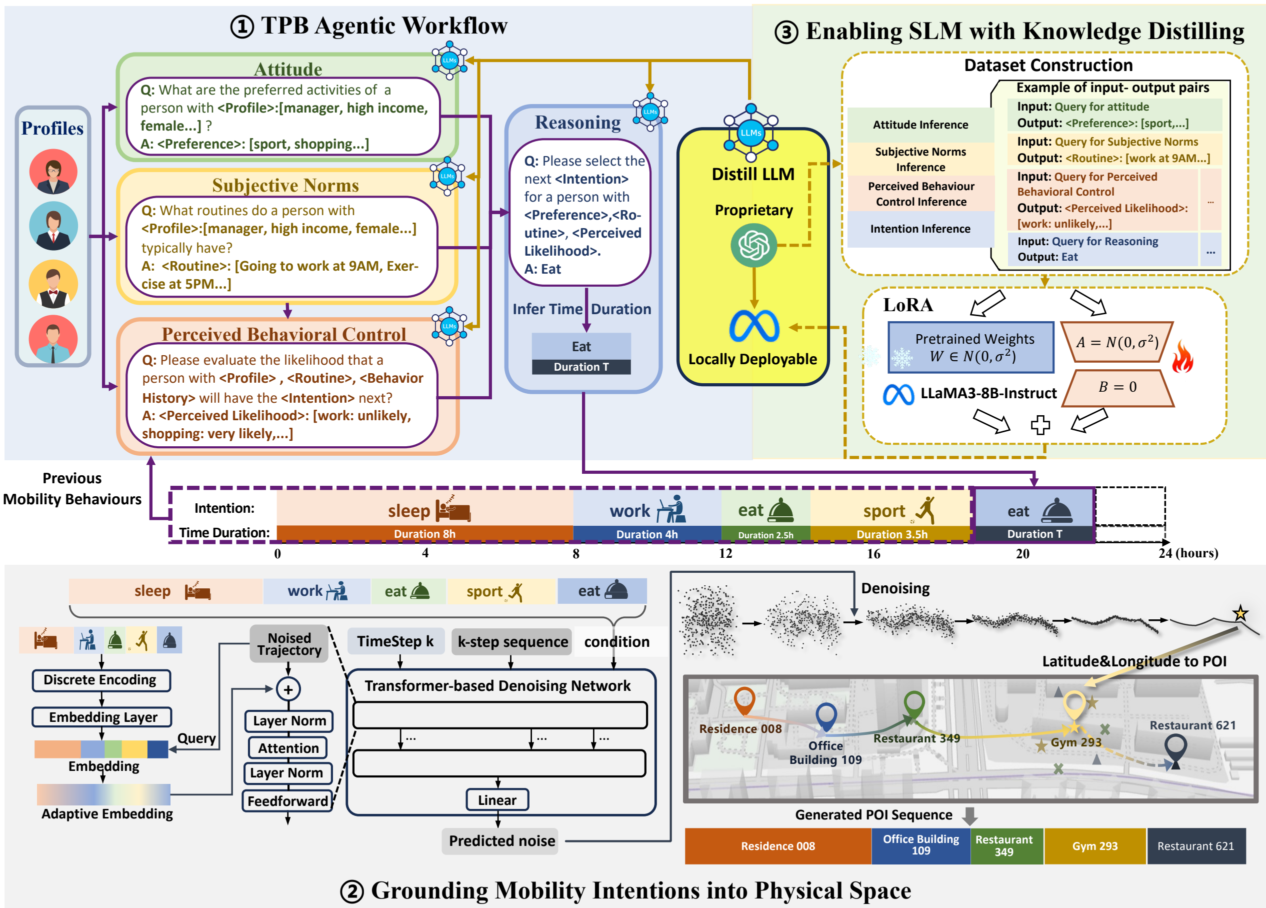


Figure 1: Overview of the proposed mobility data generator.

We employ the transformer as the backbone of the diffusion model due to its ability to effectively capture temporal and spatial relationships within trajectory sequences through the attention mechanism. The intent sequence generated by the LLM for a single day is divided into sequences of length 144 with 10-minute intervals, and the direct goal of the denoising process is to generate latitude-longitude trajectories of the same length that correspond to the intents. The Part ② in Figure 1 illustrates the detailed architecture of our denoising network, which incorporates an adaptive pre-conditioning denoising mechanism.

- **Pre-conditioning Generation:** We aim for the intention sequences generated by the LLM to provide guidance and constraints for the trajectory generation of the diffusion model. To this end, we have designed an effective pre-conditioning generation mechanism. Initially, we construct paired intention sequences and trajectory data. During training, the intention sequences are first encoded into discrete numerical sequences and then mapped to continuous conditions via an embedding module. Subsequently, we *integrate* the conditions with the trajectory data before passing them through stacked transformer layers. The term "pre" signifies that the intention conditions are integrated into the trajectory sequences prior to their input into the transformer layers, thereby eliminating the need for further modifications to the transformer architecture.
- **Adaptive Mechanism:** The *integration* of conditions with trajectory data is not merely a simple addition. We develop a query mechanism to achieve "adaptive" condition acquisition. Using the trajectory data as the key, we query the intention sequences to obtain scores analogous to attention weights. These scores are then used to weight the condition embeddings before they are added to the trajectory sequences. This approach dynamically determines the extent to which conditions should be incorporated into specific trajectory sequences, enabling the model to adaptively acquire key conditional guidance information and achieve a balance between accuracy and diversity.

Denote the denoised trajectory sequence at step k as $\mathbf{x}_k$ and the noisy trajectory sequence at step $k-1$ as $\mathbf{x}_{k-1}$. The intention condition embeddings can be represented by $\mathbf{c}_i$, where i ranges from 1 to L, the length of intention sequences. The functions $\mathbf{Q}(\cdot)$ and $\mathbf{K}(\cdot)$ correspond to the query, key transformations applied to the trajectory sequence and intention condition embeddings, respectively. The dimensionality of the query and key vectors is denoted by d. $\mathcal{D}(\cdot)$ represents the stacked transformer layers. Then the denoising process of step k can be represented as:

$$\mathbf{x}_k = \mathcal{D}\left(\mathbf{x}_{k-1} + \sum_{i=1}^{N} \text{softmax}\left(\frac{\mathbf{Q}(\mathbf{x}_{k-1}) \cdot \mathbf{K}(\mathbf{c}_i)^T}{\sqrt{d}}\right) \cdot \mathbf{c}_i\right) \quad (5)$$

After generating the latitude-longitude sequences with equal time intervals, we map the coordinates to POIs. Specifically, for each latitude-longitude coordinate and associated intent, we search the surrounding environment for POIs that match the intent type. From these, we then select the POI closest to the generated latitude-longitude coordinate.

4.3 Learning a Lightweight Model

Generation cost is a critical factor that data generators must consider. Specifically, using closed-source commercial models for mobility intention simulation incurs significant expenses, while diffusion models also introduce substantial costs in terms of training and sampling. To mitigate this issue, we offer a lightweight data generator solution that significantly reduces generation costs while maintaining the generation quality. Our approach consists of two components: enabling SLMs to replace LLMs with knowledge distillation, and grounding behavior intentions with a simple mechanistic model.

Enabling SLMs with Knowledge Distillation. We aim to enable local SLM to achieve LLM-comparable performance in mobility intention reasoning. Fine-tuning is a widely adopted approach for capability enhancement, making the construction of a dedicated dataset essential. However, fully relying on manual annotation would be prohibitively expensive. To overcome this challenge, we harness the powerful in-context learning ability of LLM, using a small number of human-annotated mobility intention sequences as examples to prompt the LLM to generate a large volume of high-quality reasoning data.

As shown in Part ③ in Figure 1, we use GPT-4-turbo to reason through our agentic workflow and record the dialogues to build a dataset. Each dialogue includes four key reasoning steps: a) preference reasoning for attitude, b) daily routine reasoning for subjective norms, c) feasibility assessment of the next intent for perceived behavioral control, and d) final intent decision, as elaborated in formulas 3 and 4. We collect 1,000 multi-turn dialogues per dataset. The constructed dataset encapsulates the mobility reasoning capabilities of LLM, which we then fine-tune on the local SLM (LLaMA3-8B-Instruct) using LoRA, realizing knowledge distillation. Finally, the fine-tuned LLaMA3-8B-Instruct replaces the LLM for task execution, eliminating API call costs and significantly reducing time overhead. The results of our method shown in Tables 1,2,3 are all generated by the fine-tuned LLaMA3-8B-Instruct model.

Grounding Behavior Intention with Simple Mechanistic Model. We leverage a specialized mechanistic model as an alternative to the diffusion model. Considering both the model scale and performance, we choose the simple but efficient *gravity model*, which highlights the distance in human migration and simplifies the migration traffic between two regions to this form: $P_{i,j} \propto \frac{P_i P_j}{r_{ij}}$, where P_i and P_j generally denote the population of region i and region j, respectively. r_{ij} denotes the distance between two regions. Taking the nonlinear factors into account, it can be further expressed as: $P_{i,j} = K m_i m_j f(r_{ij})$, where m_i is the function of P_i, as is m_j. $f(r_{ij})$ represents a decreasing function of distance and K is a normalization constant. In our implementation, we replace the population with POI density, based on the common understanding that higher POI density makes these locations more attractive. In the grounding of intentions, the current location (region i) serves as the center of the search area, and the candidate regions (region j) are defined as concentric annular regions uniformly divided by radius. For all POIs within each annulus, m_j is set to the POI density in that annulus (number per unit area). The specific form of $f(r_{ij})$ is fitted from real data.

When a new mobility intention is generated, the location is updated as follows: The geographic space is partitioned into discrete regions centered around the current location, and the density is calculated for POIs that align with the new intention. Using the gravity model, the sampling probability of each candidate POI is computed by factoring in both distance and density. Finally, next POI corresponding to the new intention is sampled.

Compared to diffusion model, the mechanistic model has fewer parameters, lower complexity and incurs less generation costs. However, precisely because of its simplicity, mechanistic model is less accurate in modeling spatiotemporal patterns compared to the diffusion model. The choice between these two models can be made flexibly depending on the scenario.

5 Experiments

For the proposed data generator, we primarily focus on the following aspects: **RQ1:** Compared to baselines, how is the quality of the generated data, and how consistent is it with real-world trajectories? **RQ2:** Does the generated mobility data hold value in downstream practical applications? **RQ3:** Does the model training and generation process involve any privacy risks? **RQ4:** What is the efficiency of data generation by our generator? In addition, we also provide heatmap visualization of trajectories in Appendix B.1, an analysis of the impact of the number of in-context examples in Appendix B.2, a diversity evaluation in Appendix B.3, and a brief introduction to our constructed simulation dataset in Appendix B.4.

5.1 Generation Setup

*5.1.1 **Original Datasets.*** We conducte experiments on three real-world trajectory datasets, learning from real data to generate semantically rich, high-quality trajectory data.

- **FourSquare [29, 30]**: FourSquare is a publicly available, well-known dataset. Specifically, we selected Tokyo, with the dataset containing 573,703 check-ins from 1,939 users collected between April 12, 2012 and February 16, 2013. Each check-in is associated with its timestamp, GPS coordinates, and semantic meaning.
- **Tencent**: Tencent dataset includes trajectory data from 100,000 users collected between October 1, 2019, and December 31, 2019, comprising 297,363,263 trajectory points. These points are associated with latitude and longitude coordinates as well as the nearest POIs. Each user is characterized by five profile attributes: income, gender, education, age, and occupation.
- **ChinaMobile**: ChinaMobile dataset contains information on the base stations connected by 1,246 users during their movements between July 1, 2017 and August 31, 2017, totaling 4,163,651 trajectory points. It also includes user profile information, such as

FourSquare											
Model	**Category**	**Statistical**						**Semantic**		**Aggregated**	
		distance	radius	duration	dayloc	itdNum	g-rank	itdErr	itdType	locfreq	odSim
TimeGeo	Mechanistic	0.027	0.09	0.141	0.176	0.069	**0.089**	0.611	0.179	0.546	8.55
MoveSim	Deep Learning	0.052	0.086	0.143	0.191	0.143	0.353	0.760	0.258	0.533	6.25
Volunteer		0.105	0.099	<u>0.118</u>	0.154	0.132	0.411	0.775	0.230	0.406	5.83
DiffTraj		<u>0.006</u>	<u>0.060</u>	0.127	0.185	0.047	0.273	0.473	0.158	0.368	4.99
Act2Loc		0.016	0.063	0.131	<u>0.123</u>	<u>0.040</u>	0.315	<u>0.355</u>	0.116	0.387	4.76
SynMob		0.008	0.077	0.122	0.128	0.044	0.105	0.399	0.143	<u>0.365</u>	4.53
Vanilla	LLM Workflow	0.195	0.145	0.207	0.513	0.384	0.698	0.521	0.149	0.548	4.54
CoT		0.176	0.116	0.193	0.341	0.128	0.334	0.436	0.115	0.498	4.58
ToT		0.154	0.112	0.147	0.338	0.097	0.301	0.379	<u>0.099</u>	0.432	<u>4.41</u>
CoPB	Hybrid	**0.002**	**0.031**	**0.107**	**0.096**	**0.038**	<u>0.098</u>	**0.253**	**0.067**	**0.312**	**3.64**

Table 1: Performance of our data generator and baselines on *FourSquare* dataset. All the metrics are better when smaller. Bold denotes the best results and <u>underline</u> denotes the second-best results. The order of magnitude for the odSim value is *E-05*.

Tencent											
Model	**Category**	**Statistical**						**Semantic**		**Aggregated**	
		distance	radius	duration	dayloc	itdNum	g-rank	itdErr	itdType	locfreq	odSim
TimeGeo	Mechanistic	0.02	0.048	0.146	0.251	0.204	0.018	0.536	0.155	0.693	7.38
MoveSim	Deep Learning	0.072	0.026	0.081	0.052	0.101	0.024	0.904	0.178	0.638	5.93
Volunteer		0.078	0.061	<u>0.023</u>	0.056	0.330	0.021	0.804	0.162	0.296	6.00
DiffTraj		<u>0.008</u>	0.028	0.030	0.293	0.253	0.029	0.597	0.080	<u>0.291</u>	6.38
Act2Loc		0.105	<u>0.024</u>	0.038	<u>0.035</u>	0.102	**0.010**	0.391	0.054	0.294	5.45
SynMob		0.011	0.026	0.027	0.036	0.132	0.015	0.621	0.147	0.301	<u>5.42</u>
Vanilla	LLM Workflow	0.152	0.113	0.185	0.465	0.634	0.647	0.578	0.187	0.603	6.80
COT		0.142	0.104	0.171	0.093	0.080	0.593	0.374	0.078	0.536	5.92
TOT		0.173	0.105	0.156	0.081	<u>0.079</u>	0.573	<u>0.351</u>	<u>0.073</u>	0.517	5.85
CoPB	Hybrid	**0.007**	**0.020**	**0.021**	**0.023**	**0.070**	<u>0.014</u>	**0.194**	**0.034**	**0.287**	**5.40**

Table 2: Performance of our data generator and baselines on *Tencent* dataset.

ChinaMobile											
Model	**Category**	**Statistical**						**Semantic**		**Aggregated**	
		distance	radius	duration	dayloc	itdNum	g-rank	itdErr	itdType	locfreq	odSim
TimeGeo	Mechanistic	0.023	0.262	0.149	0.276	0.220	0.018	0.544	0.155	0.639	8.40
MoveSim	Deep Learning	0.192	0.102	0.117	0.284	0.252	0.044	0.879	0.278	0.454	6.59
Volunteer		0.052	0.047	<u>0.028</u>	0.251	0.569	0.027	0.863	0.220	0.436	7.00
DiffTraj		<u>0.007</u>	**0.025**	0.029	0.194	0.693	0.015	0.554	0.060	<u>0.294</u>	5.58
Act2Loc		0.018	0.058	0.044	<u>0.143</u>	0.158	0.019	0.416	<u>0.057</u>	0.305	5.75
SynMob		0.017	<u>0.041</u>	0.073	0.214	0.436	<u>0.017</u>	0.771	0.192	0.440	<u>5.39</u>
Vanilla	LLM Workflow	0.029	0.048	0.135	0.209	0.138	0.057	0.642	0.198	0.436	6.59
COT		0.024	0.047	0.101	0.185	<u>0.122</u>	0.053	0.436	0.064	0.429	6.49
TOT		0.018	0.048	0.065	0.180	0.126	0.046	<u>0.397</u>	0.083	0.411	6.48
CoPB	Hybrid	**0.004**	**0.025**	**0.020**	**0.058**	**0.061**	**0.012**	**0.247**	**0.047**	**0.253**	<u>5.34</u>

Table 3: Performance of our data generator and baselines on *ChinaMobile* dataset.

income, gender, education, and age. Both Tencent and ChinaMobile are intended to be open-sourced in the near future.

Additionally, we obtain a large amount of POI information for Beijing from Tencent Maps, while for Tokyo, we directly source the POI data from the Foursquare trajectory dataset. For more detailed information about the datasets, please refer to Appendix A.1.

5.1.2 ***Training Setup.*** For each benchmark, we use 70% of the data for training and the remaining 30% for evaluation. The details of the implementation, the training parameters for the diffusion model and LLaMA3-8B-Instruct are all included in the Appendix A.7. All training is conducted on one Nvidia 4090 GPU.

5.1.3 ***Baselines.*** We consider the following three types of advanced baselines.

- **Mechanistic model.** ***TimeGeo*** [11] is a classical mechanistic model leveraging physics model and decision trees.
- **LLM reasoning approaches.** ***Vanilla*** is a straightforward baseline that uses in context learning to directly ask LLM to generate travel schedule and location choices. **Chain-of-Thought (*CoT*)** [26] prompts LLM to generate a day's intention sequence step-by-step, and choose visited location based on generated intentions. **Tree-of-Thought (*ToT*)** [31] generates intention sequences in a tree search manner, and then

make location choice similar to CoT. Moreover, we do not choose LLMob [24] as a baseline here because it requires each individual's history movement as input, and hence it will not be a fair comparison.
- **Deep generative models.** ***MoveSim*** [8], ***Volunteer*** [15], ***DiffTraj*** [38] ***Act2Loc*** [13], and ***SynMob*** [37] are five state-of-the-art deep generative models for mobility data. ***SynMob*** is specifically designed for generating trajectory dataset.

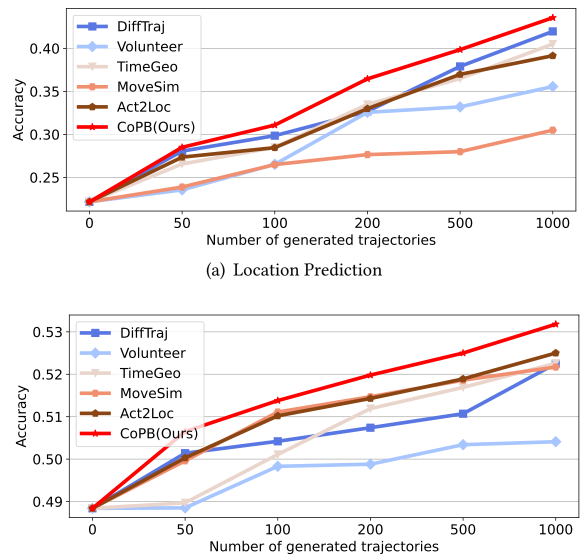
Moreover, all LLM reasoning baselines use GPT-4-turbo for reasoning, while our framework uses fine-tuned LLaMA3-8B-Instruct. We provide detailed information of baselines in Appendix A.3.

5.1.4 ***Evaluation Methods.*** We evaluate the quality of the generated mobility data from three aspects: statistical evaluation, semantic evaluation and aggregation evaluation. Here, we will briefly outline the content of each dimension, with details provided in the Appendix A.4 A.5 A.6. (1) **Statistical Evaluation.** We calculate the distance of the distribution of *distance*, *radius*, *duration*, *dayloc*, *itdNum*, *g-rank* between the simulated data and the real one with Jensen–Shannon divergence (JSD). (2) **Semantic Evaluation.** In previous work, assessments have been limited to the statistical level, while here we add evaluation on the semantic dimension by means of two metrics: intention error (*itdErr*) and the distribution of intention categories (*itdType*). *itdErr* measures the discrepancy between the generated intention sequence for a given profile and the actual intention sequence of this profile's population. While *itdType* measures the proportion of time each intention category occupies. Categories of intentions considered in our experiments are detailed in Appendix A.1. (3) **Aggregation Evaluation.** It evaluates the authenticity of the spatial distribution of data from an aggregated perspective through *locfreq* and *odSim*. *locfreq* divides the map into grid points and calculates the visit frequency for each grid point. *odSim* calculates origin-destination matrices based on trajectories and examines the generation error.

5.2 Generated Dataset Overall Analysis (RQ1)

Tables 1,2,3 shows the full results of our evaluation of the three dimensions on three datasets. Compared with the sub-optimal baseline results, our generation approach has achieved an average relative performance improvement of 21.93% across three dimensions. Notably, the improvement at the semantic level is particularly remarkable, reaching a performance boost of 35.8%. In addition to metric comparisons, we also provide heatmap visualizations in Appendix B.1.

Compared with the LLM reasoning baselines, we have achieved an average 38.1% improvement in semantic accuracy, and also achieved an average 51% improvement in the statistic and aggregated dimensions. The vanilla LLM method exhibited the weakest performance, demonstrating that LLMs with simple and direct prompts fail to simulate satisfactory mobility behavior. Even with POI location information (latitude-longitude and planar mapping) as inputs, the poor metrics reveal the LLM's low spatial sensitivity and difficulty aligning with real trajectory distributions. The introduction of CoT or ToT improves semantic reasoning, confirming that additional reasoning steps enhance the rationality of results, but their impact on other dimensions remains minimal.



(a) Location Prediction

(b) Intention Prediction

Figure 2: Performance boost in downstream mobility prediction task when different sizes of generated mobility data is provided as data augmentation.

Compared to deep generative models, our generator achieves significant improvements at the semantic level, with a 45.5% increase in intention reasoning accuracy and a 23.3% improvement in intention distribution consistency. Performance of deep generative models in statistical and aggregation dimensions is tolerable, but their results on the semantic metrics are quite poor. Even Act2Loc, which has a specialized transformer for intent generation, fail to meet the accuracy and diversity requirements. But Act2Loc's performance in numerical and statistical evaluations is close to ours, demonstrating the effectiveness of the divide-and-conquer approach.

5.3 Downstream Application (RQ2)

We further investigate the practical value of the generated dataset for downstream applications. Specifically, we assess its effectiveness in the general *mobility prediction task*, where the generated data is integrated with scarce real-world data to enhance the performance of training mobility prediction models. This approach is a commonly adopted method for evaluating the empirical utility of generated data [23, 38]. The experiment is conducted on the Tencent dataset and the task is divided into geospatial prediction and semantic-level activity type prediction. We fix the number of real mobility trajectory as 100 and vary the number of generated mobility trajectory across 0/50/100/200/500/1000. As shown in Figure 2, the increasing trend shows that the performance of trained mobility prediction model consistently improves as the amount of generated data increases. Compared to baselines, our generated data leads to a significantly larger accuracy improvement, demonstrating the high practical value of CoPB.

5.4 Privacy Risk Analysis (RQ3)

As mentioned in the introduction, privacy risks are one of the reasons hindering the current large-scale sharing and application of mobility data. Therefore, here we conduct two experiments on the three datasets to investigate whether our proposed generator poses any risk of privacy leakage.

Uniqueness Testing. Define the overlap ratio as the proportion of identical slices to the total length. We randomly sample 1,000 trajectories from the generated data of each dataset and compare them pairwise with 1,000 length-matched trajectories from the real dataset in a slice-by-slice way, then calculate the average overlap ratio. We first measure the similarity at the intention level. The overlap ratio is 19.62% for Foursquare, 20.31% for Tencent and 22.61% for ChinaMobile. Given that work and sleep periods are relatively fixed, overlap rate around 20% can not be considered high. Furthermore, when considering the accuracy of physical locations, the average overlap rate is only 2.05‱, and the average maximum overlapping rate (find the most similar real trajectory for each generated one) is only 5.86‰, demonstrating that our generator is indeed able to generate brand-new and unique trajectories rather than simply copying.

Membership Inference Attacks: Membership Inference Attacks arose from concerns about privacy breaches in training process. An "attack" refers to analyzing the model's output to judge whether a data point is in the model's training set. A correct judgment is considered a successful attack. The lower the success rate, the stronger the model's privacy safeguards. Our experimental setup follows the configurations in [32] and [12]. To control the impact of classification methods, we used three classic and commonly employed classifiers: Random Forest (RF), Logistic Regression (LR), and Support Vector Machine (SVM) to perform the attacks. The positive samples are choosed from the trajectories used to fit our generator, while the negative samples are from the remaining dataset. The feature used is the overlap ratios from multiple runs. The metric for evaluation is the success rate, defined as the percentage of successful attempts in determining whether a sample is part of the training set. As shown in Figure 3, the attack success rates for all three methods remain below 52% across all three datasets, which is a similar privacy standard as in previous works [12, 32]. Given that naive random guessing would yield an attack success rate of 50%, our results indicate that the attacker can hardly infer whether individuals are in the training data or in the generated data. Therefore, it can be concluded that our data generator provides excellent privacy safeguards and poses minimal privacy risks.

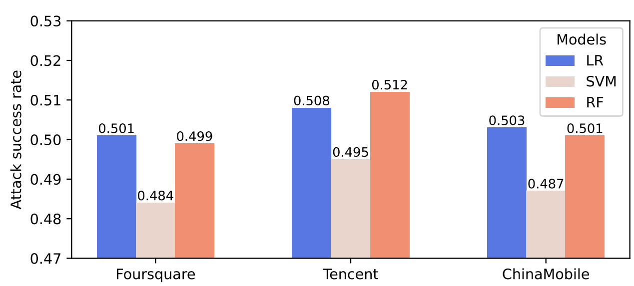


Figure 3: Success rate of membership inference attacks.

5.5 Generation Efficiency Evaluation (RQ4)

To enhance the applicability of our data generator in cost-sensitive scenarios, we propose a lightweight data generator solution for both LLM and diffusion model. We conduct comparative experiments on Tencent dataset to investigate how it improves efficiency.

LLMs	itdErr	itdType	time cost (s)	API cost ($)
GPT-3.5-turbo	0.233	0.038	170.0	22.02
GPT-4	0.196	0.035	263.0	94.72
GPT-4-turbo	0.194	0.033	123.0	52.36
LLaMA3-8B-Instruct	0.315	0.065	92.0	0.0
Fine-tuned LLaMA3-8B-Instruct	0.194	0.034	92.0	0.0

Table 4: Lightweight data generator: LLM→SLM.

Methods	distance	radius	duration	dayloc	time cost (s)
Best Baseline	0.008	0.024	0.023	0.035	43.1
Ours (Diffusion)	0.007	0.020	0.021	0.023	37.5
Ours (Gravity)	0.010	0.019	0.023	0.029	22.3

Table 5: Lightweight data generator: Diffusion→Gravity.

We first test several proprietary GPT models on the proposed agentic workflow and compare the quality of their generated intention sequences with that of the original LLaMA3-8B-Instruct model and its fine-tuned version. We show the performance of two semantic metrics and the inference time and API cost in Table 4, which indicates original LLaMA3-8B-Instruct performed much worse than GPT-4 or GPT-4-turbo, but after fine-tuning, it can achieve performance similar to GPT-4. The API cost and inference time cost refer to the expenses for generating 1,000 intention sequences. We use official OpenAI APIs for all GPT models and deploy LLaMA3-8B-Instruct on a local Nvidia 4090 GPU for inference. All experiments are carried out under the same network condition. Results reveal that the finetuned LLaMA3-8B-Instruct can decrease the simulation time by 28.24% and reduce the cost of the api to zero.

Moerover, we compare the mechanistic model with the diffusion model, using exactly the same 1,000 intent sequences as conditions to generate 1,000 trajectories. The comparison is conducted across four spatial metrics and the generation time, and we also visualize the best baseline results. As shown in the results in Table 5, while the mechanism model lags behind the diffusion model in terms of quality, it still outperforms the best baseline, and the generation time is only 60% of that of the diffusion model. Both experiments collectively demonstrate that our lightweight data generator solution is highly effective.

6 Conclusion

In this paper, we propose our Chain-of-Planned-Behaviour data generator for realistic human mobility data generation. We design a behavior science-driven LLM agentic workflow to generate coherent behavioral intentions. Conditioned on these intentions, we develop a diffusion model that grounds abstract intentions into physical trajectories. We also introduce a lightweight data generation solution to enhance the accessibility and practicality of mobility data generation. Based on this generator, we further construct and release an open-source dataset containing 200,000 synthetic mobility trajectories. We hope that our approach contributes to advancements in mobility data mining.

References

[1] Gati V Aher, Rosa I Arriaga, and Adam Tauman Kalai. 2023. Using large language models to simulate multiple humans and replicate human subject studies. In *International Conference on Machine Learning*. PMLR, 337–371.

[2] Icek Ajzen. 1991. The theory of planned behavior. *Organizational behavior and human decision processes* 50, 2 (1991), 179–211.

[3] Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. 2023. Palm 2 technical report. *arXiv preprint arXiv:2305.10403* (2023).

[4] Lisa P Argyle, Ethan C Busby, Nancy Fulda, Joshua R Gubler, Christopher Rytting, and David Wingate. 2023. Out of one, many: Using language models to simulate human samples. *Political Analysis* 31, 3 (2023), 337–351.

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

[6] Lin Chen, Fengli Xu, Zhenyu Han, Kun Tang, Pan Hui, James Evans, and Yong Li. 2022. Strategic COVID-19 vaccine distribution can simultaneously elevate social utility and equity. *Nature Human Behaviour* 6, 11 (2022), 1503–1514.

[7] Yves-Alexandre De Montjoye, Laura Radaelli, Vivek Kumar Singh, and Alex "Sandy" Pentland. 2015. Unique in the shopping mall: On the reidentifiability of credit card metadata. *Science* 347, 6221 (2015), 536–539.

[8] 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*. 3426–3433.

[9] Luciano Floridi and Massimo Chiriatti. 2020. GPT-3: Its nature, scope, limits, and consequences. *Minds and Machines* 30 (2020), 681–694.

[10] Marta C Gonzalez, Cesar A Hidalgo, and Albert-Laszlo Barabasi. 2008. Understanding individual human mobility patterns. *nature* 453, 7196 (2008), 779–782.

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

[12] 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.

[13] 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 (2024), 407–431.

[14] Yanchi Liu, Chuanren Liu, Xinjiang Lu, Mingfei Teng, Hengshu Zhu, and Hui Xiong. 2017. Point-of-interest demand modeling with human mobility patterns. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*. 947–955.

[15] 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 *KDD 2023*.

[16] 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.

[17] Lucidrains. 2021. denoising-diffusion-pytorch. https://github.com/lucidrains/denoising-diffusion-pytorch.

[18] Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. 2023. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*. 1–22.

[19] Chen Qian, Xin Cong, Cheng Yang, Weize Chen, Yusheng Su, Juyuan Xu, Zhiyuan Liu, and Maosong Sun. 2023. Communicative agents for software development. *arXiv preprint arXiv:2307.07924* (2023).

[20] Murray Shanahan, Kyle McDonell, and Laria Reynolds. 2023. Role play with large language models. *Nature* 623, 7987 (2023), 493–498.

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

[22] Peter R Stopher and Stephen P Greaves. 2007. Household travel surveys: Where are we going? *Transportation Research Part A: Policy and Practice* 41, 5 (2007), 367–381.

[23] Huandong Wang, Changzheng Gao, Yuchen Wu, Depeng Jin, Lina Yao, and Yong Li. 2023. PateGail: a privacy-preserving mobility trajectory generator with imitation learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 37. 14539–14547.

[24] Jiawei Wang, Renhe Jiang, Chuang Yang, Zengqing Wu, Makoto Onizuka, Ryosuke Shibasaki, Noboru Koshizuka, and Chuan Xiao. 2024. Large language models as urban residents: An llm agent framework for personal mobility generation. *arXiv preprint arXiv:2402.14744* (2024).

[25] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. 2022. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682* (2022).

[26] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems* 35 (2022), 24824–24837.

[27] Fengli Xu, Yong Li, Depeng Jin, Jianhua Lu, and Chaoming Song. 2021. Emergence of urban growth patterns from human mobility behavior. *Nature Computational Science* 1, 12 (2021), 791–800.

[28] 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*. 1241–1250.

[29] Dingqi Yang. 2014. Foursquare Check-in Dataset. https://sites.google.com/site/yangdingqi/home/foursquare-dataset. Accessed: 2025-07-26.

[30] Dingqi Yang, Daqing Zhang, Vincent W Zheng, and Zhiyong Yu. 2014. Modeling user activity preference by leveraging user spatial temporal characteristics in LBSNs. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 45, 1 (2014), 129–142.

[31] Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. *arXiv preprint arXiv:2305.10601* (2023).

[32] Yuan Yuan, Jingtao Ding, Huandong Wang, and Depeng Jin. 2024. Generating daily activities with need dynamics. *ACM Transactions on Intelligent Systems and Technology* 15, 2 (2024), 1–28.

[33] 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.

[34] Yuan Yuan, Huandong Wang, Jingtao Ding, Depeng Jin, and Yong Li. 2023. Learning to Simulate Daily Activities via Modeling Dynamic Human Needs. In *Proceedings of the ACM Web Conference 2023*. 906–916.

[35] Yu Zheng, Xing Xie, Wei-Ying Ma, et al. 2010. GeoLife: A collaborative social networking service among user, location and trajectory. *IEEE Data Eng. Bull.* 33, 2 (2010), 32–39.

[36] Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyan Luo, Zhangchi Feng, and Yongqiang Ma. 2024. LlamaFactory: Unified Efficient Fine-Tuning of 100+ Language Models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*. Association for Computational Linguistics, Bangkok, Thailand. http://arxiv.org/abs/2403.13372

[37] Yuanshao Zhu, Yongchao Ye, Ying Wu, Xiangyu Zhao, and James Yu. 2024. SynMob: creating high-fidelity synthetic GPS trajectory dataset for urban mobility analysis. *Advances in Neural Information Processing Systems* 36 (2024).

[38] Yuanshao Zhu, Yongchao Ye, Shiyao Zhang, Xiangyu Zhao, and James Yu. 2023. DiffTraj: Generating GPS Trajectory with Diffusion Probabilistic Model. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Appendix

A Details of the Experimental Setup

A.1 Datasets

FourSquare Dataset [30]. This is a publicly available and widely used dataset. The dataset comprises 573,703 check-in records from 2,293 users collected in Tokyo, Japan, over a 10-month period from April 12, 2012, to February 16, 2013. Each check-in entry includes a specific timestamp, GPS coordinates, and semantic information about the visited venue, represented by fine-grained venue categories. The Foursquare dataset offers three geographic scales: NYC, Tokyo, and Global-scale. For our study, we selected the representative Tokyo dataset. The dataset contains a total of 276 fine-grained POI types. To align with the number of intents, we aggregated them into 13 broader categories: 1. Arts&Entertainment, 2. Athletics&Sports, 3. Clothing&Store, 4. College&University&School, 5. Food&Drink, 6. Medical&Center, 7. Movie&Theater, 8. NightlifeSpot, 9. Outdoors&Recreation, 10. Travel&Others, 11. Shop&Service, 12. Spiritual&Religious, 13. Hotel&Accommodation.

Tencent Dataset. This dataset is collected from a social network platform and records users' mobility trajectories. Additionally, users' profiles, such as income level, gender, occupation, education level and age, are collected through digital surveys. We randomly select 200 7-day trajectories from the whole dataset to fine-calibrate the POI matching and manually annotate specific intention types.

Mobile Operator Dataset. This dataset is collected from a local mobile operator and records the time and location of users' connections to nearby cellular base stations. Similarly, the user profiles are collected through surveys. For each record in this dataset, we match the location of the visit to the nearest POI based on latitude and longitude coordinates, and then characterize the intention of this visit by the category attributes of the POI.

Datasets	Foursquare	Tencent	Mobile Operator
Duration	April 12, 2012 - February 16, 2013	October 1, 2019 - December 31, 2019	July 1, 2017 - August 31, 2017
City	Tokyo	Beijing	Beijing
#Users	1939	100000	1246
#Trajectory Points	573,703	297363263	4163651

Table 6: Basic information about the trajectories of the two datasets

The basic information about the trajectory data for these datasets is shown in table 6. The sampling time for all datasets is unfixed.

From the micro level, there is a close relationship between people's identity information and people's mobility behaviour. Income level, gender, education level, age, job, etc. all affect people's travel patterns. For example, the mobility pattern of an IT professional is likely to be a simple "point-to-point" commute between home and the office while a delivery boy's day may involve commuting between numerous locations in the city. In our experiment on Tencent dataset, the character profile includes 4 kinds of attributes: gender, occupation, education level, and income level. In the experiments on ChinaMobile dataset, the profile includes 3 kinds of attributes: gender, education level, and income level. Additionally, due to privacy risks, we did not directly utilize real profile information in the experiment. Instead, based on the locations of homes in trajectories, we statistically derived the demographic distribution of each area. During generation, we initially sampled an area where a home is located and then sampled a specific profile from the demographic distribution of that area. Due to the lack of real user profile data in the Foursquare dataset, we obtain basic demographic information, such as gender ratio and occupational distribution, from Japanese population statistics and used this data for sampling. The distribution of character attributes for Tencent and Chinamobile datasets is shown in tables 7 and 8.

User Profile	Category
Income	Low (11.44%), Slightly Low (18.12%), Medium (28.67%),Slightly High (18.19%), High (3.36%), Uncertain (20.22%)
Gender	Male (54.67%), Female (44.68%), Uncertain (0.65%)
Education	Bachelor's degree (25.90%), High school diploma (21.70%), Master's degree (9.94%), Elementary school diploma (13.30%), Junior high school diploma (13.05%), Associate degree (8.02%), Doctoral degree (5.27%)
Age	0-30 (44.50%), 30-40 (30.40%), 40-60 (23.55%), 60-99 (1.55%)
Job(top8)	IT Engineer (11.57%), Online Sales (8.37%), Training Instructor (7.34%), Investment (7.31%), Finance (6.97%), Copywriting (6.91%), Front Desk Reception (6.15%), Technical Worker (6.05%)

Table 7: Distribution of profiles in Tencent dataset

User Profile	Category
Income	Low (23.27%), Relatively Low (20.30%) Medium (36.44%), Relatively High (15.97%) High (4.01%)
Gender	Male (63.56%), Female (36.44%)
Education	Bachelor degree (58.43%), High school degree (21.03%) Master's degree (11.32%), Junior high school degree (9.23%)
Age	0-30 (22.71%), 30-40 (28.01%), 40-60 (40.85%), 60-99 (8.43%)

Table 8: Distribution of profiles in Mobile Operator dataset

POI Dataset. Our POI data for Beijing comes from the Tencent Map, developed by Tencent, China. It includes a total of 933,576 POIs covering the area within the sixth ring road of Beijing. Each POI has information such as ID, name, three-level category classification, specific location, and the corresponding AOI. An example of a POI is as follows:

```
{ 'id': 700831650, 'name': 'Building 3, Jintai North
Street Community', 'category': '281010', 'position': {
'x': 454577.8408222591, 'y': 4419119.168818601 }, 'aoi_id':
500006223, 'external': { 'tencent_poi_id': '162749807
28221167175' }, 'shapely_xy': <POINT (454577.841
4419119.169)>, 'shapely_lnglat': <POINT (116.468 39.921)>
}
```

The classification number `281010` represents a three-level classification. Here, `28` denotes a residential community, the first `10` indicates a residential area, and the second `10` specifies a residential neighborhood. The category meanings corresponding to the ID are also provided in the repository.

However, for the POI data in Tokyo, we used the check-in locations covered by Foursquare, without incorporating any external POI datasets.

A.2 Categories of Intentions Considered

Our experiment considers a total of 10 types of intentions, which are: 1. go to work, 2. go home, 3. eat, 4. do shopping, 5. do sports, 6. excursion, 7. leisure or entertainment, 8. sleep, 9. medical treatment, 10. handle the trivialities of life.

A.3 Baselines

- **TimeGeo**. This work models human mobility behaviour as a chain of probabilistic choices. It defines the weekly home-based tour number, dwell rate and burst rate to model the temporal choices and employs EPR to model the spatial choices.
- **MoveSim**. This model utilizes a generative adversarial framework, wherein the discriminator is continually reinforced to supervise the generator in producing high-quality trajectory data. To introduce prior knowledge about urban spatial structures, an attention-based region network is designed within the generator. Furthermore, leveraging the spatial continuity and temporal periodicity migration patterns, both the generator and discriminator are pretrained to accelerate the learning process.
- **Volunteer**. This work uses two vae for joint modeling, the first user VAE is used to learn the distribution of an individual's residence and workplace, and the second VAE decouples travel time and dwell time to accurately model an individual's movement trajectory.
- **DiffTraj**. This work proposes a spatio-temporal diffusion probabilistic model for trajectory generation, the core idea of which is to reconstruct and synthesize geographic trajectories from white noise through a reverse trajectory denoising process. UNet is used to embed prior information regarding the starting and ending positions, as well as the movement speed, and accurately estimate the noise level in the inverse process.
- **Act2Loc**. This model integrates deep learning models with mechanistic models. It uses a transformer to continue writing new trajectories based on real activity sequences, while also employing the *Universal Opportunity (UO)* model to select geographic grid points based on population distribution. Despite leveraging a divide-and-conquer approach, it still adheres to the paradigm of fitting before sampling in deep generative models. The use of prediction with transformers for generation also limits the model's freedom in generating.
- **SynMob**. This model uses a diffusion network to learn mobility patterns from real data and generate trajectory data. They use Unet as the denoising backbone and employ a classifier-free diffusion guidance method for training. Additionally, they incorporate the non-Markov diffusion process method to improve efficiency. Based on this model, they have built simulated trajectory datasets for Xi'an and Chengdu.

A.4 Statistical Metrics

We use JSD (Jensen-Shannon Divergence) to calculate the distance between the distributions of metrics for generated data and real data, which is defined as $JSD(p;q) = h((p+q)/2) - (h(p)+h(q))/2$, where h represents the Shannon information, p and q are two distributions. The lower the JSD, the closer the two distributions are, and the better the quality of generated data is. Detailed information about the four statistics is shown below:

- **distance**. distance traveled each time, which reflects the habit of traveling longer or shorter distances.
- **radius**. radius of gyration, which represents the spatial range of the user's daily activities.
- **duration**. time spent on each move, which reflects the habit of traveling for longer or shorter durations.
- **dayloc**. daily visited locations, which is calculated as number of different locations visited per day for each of the user.
- **itdNum**. number of intentions per day, which differs from dayloc in that there is no de-duplication of locations.
- **g-rank**. number of visits to different locations, which is calculated as the top-100 visiting frequency among all the locations.

A.5 Semantic Metrics

One preparatory work before the semantic evaluation is needed: matching intention categories for trajectory records. For the real data, we get the intention category information of each location visit record by manual labeling or POI matching. For the results we generate, they naturally contain the category distinction. For the baseline model without the concept of intention category, after we identify the stay at home and working from the trajectories, other intentions are randomly assigned.

Then, for the metric **itdErr**, we need to calculate the similarity by direct comparison between the generated intention sequences and the real ones. Since real data inevitably possesses uncertainty and randomness, we aggregate 7 days of a person's trajectories into 1 day by majority voting on the same time slice. Of course, we only focus on the level of intention type when aggregating. For example, if 5 of these 7 days are eating at 12:00, and 2 days are working at 12:00, then the intention of 12:00 in the aggregated result is eating. With the help of aggregation, the uncertainty of the trajectory is eliminated and a more stable and representative real trajectory of the specified profile is obtained. Then the generated results are compared with the aggregated trajectory on a time-slice-by-time-slice basis. The length of the time slices is standardized to half an hour. For the metric **itdType**, we calculate the time proportion of each category of intentions in the trajectories weighted by durations, and calculate the JSD of the proportion vectors.

A.6 Aggregated Metrics

To calculate the metric **locfreq**, we need to partition the map uniformly into grids, count the frequency of trajectories visiting each grid, and calculate JSD between the frequency vectors. To get **odSim**, we count the movement of individuals between grids, calculates the OD transfer probability matrix, and computes the MSE values between the matrices. In addition to the numerical measurement, we also visualize the grid-visiting frequency difference between the generated data and real data using heatmaps, such that we can intuitively judge the quality of the generated data through the visual difference between our model and baselines.

A.7 Implementation Details and Training Hyperparameters

The training hyperparameters of the diffusion model are as follows:

- modeldim: 256
- epochs: 100001
- diffusionstep: 100
- train_batch_size: 256
- train_lr: 8e-05
- save_and_sample_every: 2000
- N_block: 4

The implementation of the diffusion component in our method builds upon the open-source repository denoising-diffusion-pytorch by Lucidrains [@lucidrains2021]. Our work introduces substantial modifications, including a novel conditional generation mechanism, a redesigned attention architecture, and a tailored input-output serialization process. We gratefully acknowledge the foundational implementation provided by the repo [17].

Our fine-tuning for LLaMA3-8B-Instruct was based on the open-source project LLAMA-FACTORY [36]. Hyperparameters for training are as follows:

- finetuning_type: lora
- cutoff_len: 4096
- max_samples: 1000
- overwrite_cache: true
- preprocessing_num_workers: 16
- per_device_train_batch_size: 2
- gradient_accumulation_steps: 8
- learning_rate: 0.0001
- num_train_epochs: 2.0
- lr_scheduler_type: cosine
- warmup_steps: 0.1
- fp16: true

B Additional Experiments

B.1 Heatmap Visualization of Generated Trajectories.

In addition to using metrics for performance comparison, heatmaps are employed to contrast grid-visiting frequencies of generated and real data on Tencent dataset in a square area around Beijing's fifth ring road, allowing for a comparison of the outcomes produced by our data generator with the baselines. As shown in Figure 4, the color depth of each grid point represents the magnitude of the difference. The closer to purple, the larger the difference. Trajectories generated by CoPB's diffusion are the closest to the real ones as there are very few purple or red grid color blocks on our graph. Results of Act2Loc are visually similar to ours, except that some local areas have slightly darker colors. Results of DiffTraj and Volunteer are acceptable, indicating Diffusion and VAE indeed have good spatial fitting capabilities. But MoveSim and TimeGeo are poorer.

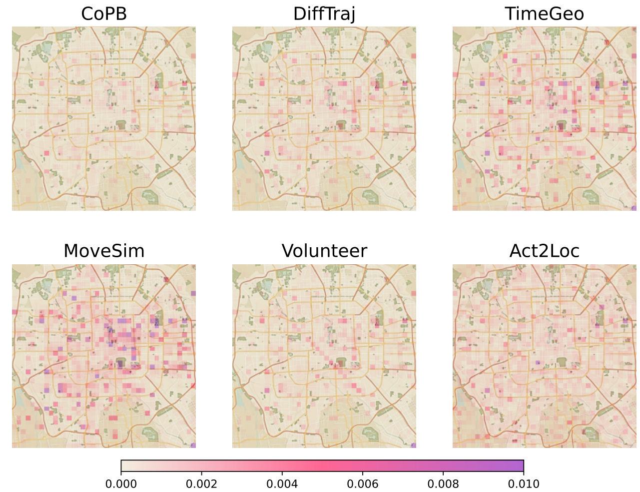


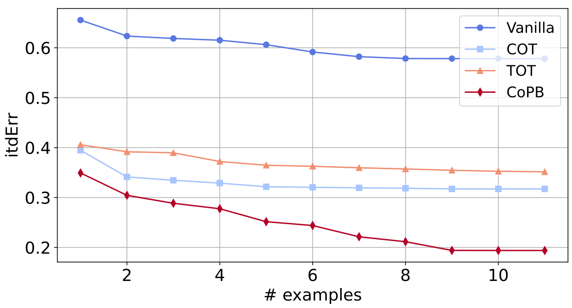
Figure 4: Visualization of grid visit frequency error after aggregating all the trajectories on Tencent dataset.

B.2 Impact of Different Numbers of In-context Examples

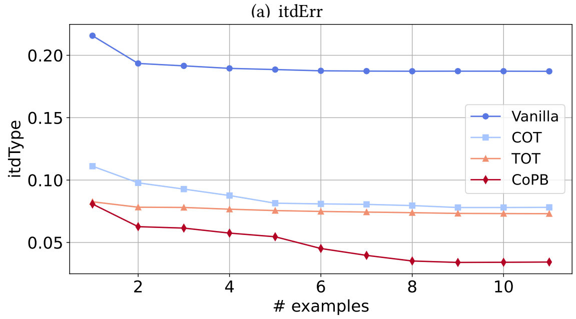
We explore how in-context examples in prompts affect LLMs' few-shot performance. Can we really enhance the behavioral imitation capability of LLMs with just a few words? We start from zero and gradually increase the number of in-context examples up to 10, comparing the changes in two semantic metrics (since prompt design directly impacts semantic quality). We test all LLM reasoning baselines and our CoPB on the Tencent dataset. The results in Figure 5 reveal that the accuracy of intent reasoning generally improves with the increase in the number of examples, with a particularly prominent enhancement in the initial stage. The curves demonstrate that the LLM baseline models not only exhibit poor overall performance but also are unable to effectively utilize more examples to continuously improve their performance. In contrast, CoPB can fully leverage the profound semantics contained in examples to achieve steady performance enhancement, and a notable performance advantage can already be obtained with just 8 examples.

B.3 Diversity Evaluation

The diversity of generated mobile trajectories is a less discussed issue. Previous works have mainly focused on the quality of the data itself, while discussions on diversifying the generated data are lacking. Considering the complexity and diversity of human behaviour, we believe that discussing the diversity of generated trajectories is essential. Moreover, the diversity of trajectories generated by LLM has always been a focal point of concern. Therefore, in this section, we will use additional experiments and visualizations to discuss the diversity issue of the proposed framework. Specifically, we generated 10 1-day results for six people. Every person's profile, home and work locations are fixed separately. The visualization of generated intention sequence is in Figure 6. We use different colored blocks to represent different intentions, with the start and end times of the intentions determining the relative position and length of each block. For example, blue represents sleep, yellow



(a) itdErr



(b) itdType

Figure 5: The impact of the number of in-context examples on few-shot performance. Tested on Tencent Dataset.

represents eating, orange represents home, shallow blue represents work, and pink represents entertainment. As seen in the Figure 6, each individual has relatively fixed work and rest patterns, but there is also significant diversity with more random variations occurring. The visualization of trajectories is in Figure 7. We use different colors to visualize these 10 different trajectories. As seen in the figure, the locations are mostly distributed around the home and workplace. However, there are also possibilities of visiting more distant POIs.

B.4 Introduction to the constructed simulation dataset

We construct and release a large-scale simulated mobility dataset based on our proposed CoPB data generator. The dataset contains a total of **200,000** user trajectories, comprising **794,820** individual trajectory points. All trajectories are situated within the geographic context of Beijing, China. Each trajectory is represented as a sequence of user activities, annotated with behavioral intent, timestamps, place descriptions, and geographic coordinates. To facilitate data processing, we provide both `.jsonl` and `.parquet` formats. The dataset is publicly available via Google Drive: https://drive.google.com/drive/folders/1V7FIitnuC4w6b4INXcBItYKgq4z97moB?usp=drive_link

More detailed usage instructions and documentation are provided in the GitHub repository: https://github.com/PLUTO-SCY/CoPB This project repository will be continuously maintained and updated, with future releases of simulated mobility datasets covering more cities.

C Prompts

All prompts used in our LLM agentic workflow are listed as follows:

C.1 Attitude Reasoning

```
What activities does a person with a character
profile of [Gender: woman; Education: Master's
degree; Consumption level: slightly high; Occupation
 or Job Content: Online Sales;] tend to prefer and
engage in? Please list 2-5 points.

Output format: 1.xxx 2.xxx 3.xxx
```

C.2 Subjective Norms Reasoning

```
Most people in life have some fixed routines or
habits that are generally non-negotiable and must be
 adhered to. For example, civil servants usually
have fixed working hours (such as from 9 am to 5 pm),
 and engineers at technology companies usually go to
 work close to noon, and may not get off work until
10 p.m.. Some people insist on going to bed before
23:00, while some people are used to staying up late
 and getting up very late too.

Now I give you a description of a person, and I hope
 you can generate 3 habits or tendencies that this
person may have.

Hint: I hope you can take into consideration the
habits that people of this kind might realistically
have in their daily life. For example, a significant
 number of people may not exercise every day; for
most people, their lives may have little aside from
work and rest with few fixed activities; some jobs
may require frequent overtime until 22:00 or even
later, while others may only require half-day work.
I don't need you to tell me how this person should
plan their life; I want you, based on this person's
attributes, to tell me what kind of life and habits
they might have in real life.

The person's basic information is as follows:

Gender: woman; Education: Master's degree;
Consumption level: slightly high; Occupation or Job
Content: Online Sales;

Please generate 2-5 anchor points for him. No
explanation is required. Try to keep it concise,
emphasizing time and key terms (Example1: You are
accustomed to waking up before 8 AM. Example2: Your
working hours are from 9 AM to 7 PM.).
```

Figure 6: Diversity in the generated intention sequence of 6 randomly selected users.

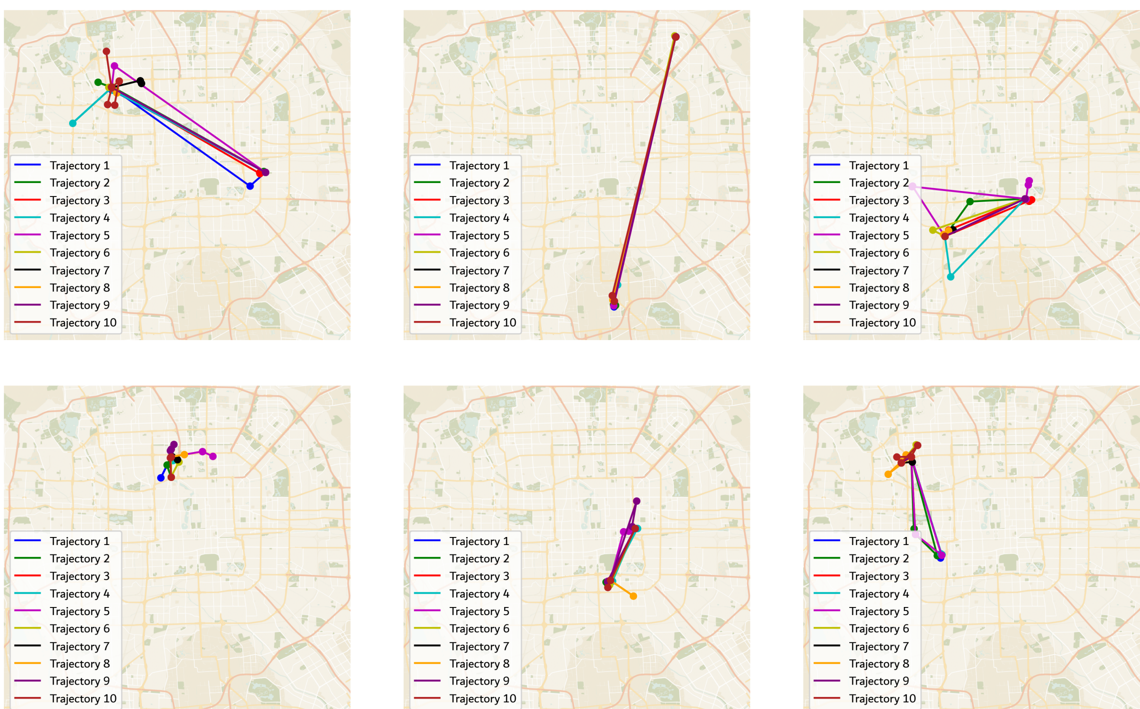


Figure 7: Diversity in physical trajectory of randomly selected users.

```
Please answer in the second person using an
affirmative tone and organize your answers in 1.xxx
2.xxx 3.xxx format.
```

C.3 Global System Information

The role of this section's prompts is system, which includes the main task description and guidelines and needs to be added at each intention reasoning step.

```
You are a person and your basic information is as
follows:
Gender: woman; Education: Master\'s degree;
Consumption level: slightly high; Occupation or Job
Content: Online Sales;

You have some preferences as follows: <Preferences>
You have some habits or tendencies as follows: <
Anchors>

Now I want you to generate your own schedule for
today.(today is Friday).

The specific requirements of the task are as follows:

1. You need to consider how your character
attributes, routines or habits relate to your
behavior decisions.
2. I want to limit your total number of events in a
day to 6. I hope you can make every decision based
on this limit.
3. I want you to answer the basis and reason behind
each intention decision.

Note that:
1. All times are in 24-hour format.
2. The generated schedule must start at 0:00 and end
 at 24:00. Don't let your schedule spill over into
the next day.
3. Must remember that events can only be choosed
from [go to work, go home, eat, do shopping, do
sports, excursion, leisure or entertainment, go to
sleep, medical treatment, handle the trivialities of
 life, banking and financial services, cultural
institutions and events].
4. I'll ask you step by step what to do, and you
just have to decide what to do next each time.

Here are some examples for reference. For each
example I will give a portrait of the corresponding
character and the reasons for each arrangement.

Example 1:
This is the schedule of a day for a coder who works
at an Internet company.
["go to sleep", "(00:00, 11:11)"], (Reason: Sleep is
 the first thing every day.)
["go to work", "(12:08, 12:24)"], (Reason: Work for
a while after sleep. This person's working hours are
 relatively free, there is no fixed start and finish
 time.)
["eat", "(12:35, 13:01)"], (Reason: It's noon after
work. Go get something to eat.)
["go to work", "(13:15, 20:07)"],   (Reason: After
lunch, the afternoon and evening are the main
working hours. And he works so little in the morning
 that he need to work more in the afternoon and
evening. So this period of work can be very long.)
["go to sleep", "(21:03, 23:59)"]  (Reason: It was
already 9pm when he got off work, and it is time to
go home and rest.)

Example 2:
This is the schedule of a day for a salesperson at a
 shopping mall.
["go to sleep", "(00:00, 08:25)"], (Reason: Of
course the first thing of the day is to go to bed.)
["go to work", "(09:01, 19:18)"], (Reason: Generally,
 the business hours of shopping malls are from 9 am
to 7 pm, so she works in the mall during this time
and will not go anywhere else.)
["go home", "(20:54, 23:59)"], (Reason: It's almost
9pm after getting off work. Just go home and rest at
 home.)

Example 3:
This is the schedule of a day for a manager who is
about to retire.
["go to sleep", "(00:00, 06:04)"], (Reason: He is
used to getting up early, so he got up around 6 o'
clock in the morning.)
["eat", "(08:11, 10:28)"], (Reason: He has the habit
 of having morning tea after getting up and enjoys
the time of slowly enjoying delicious food in the
morning.)
["go home", "(12:26, 13:06)"], (Reason: After
breakfast outside, take a walk for a while, and then
 go home at noon.)
["excursion", "(13:34, 13:53)"], (Reason: He stays
at home most of the morning, so he decides to go out
 for a while in the afternoon.)
["go to work", "(14:46, 16:19)"], (Reason: Although
life is already relatively leisurely, he still has
work to do, so he has to go to the company to work
for a while in the afternoon.)

Example 4:
```

```
This is the schedule of a day for a lawyer who
suffers a medical emergency in the morning.
["go to sleep", "(00:00, 09:36)"], (Reason: Sleep
until 9:30 in the morning. Lawyers' working hours
are generally around 10 o'clock.)
["medical treatment", "(11:44, 12:03)"], (Reason: He
 suddenly felt unwell at noon, so he went to the
hospital for treatment.)
["go to work", "(12:27, 14:56)"], (Reason: After
seeing the doctor, the doctor said there was nothing
 serious, so he continued to return to the company
to work for a while.)
["go to sleep", "(17:05, 23:59)"], (Reason: Since he
 was not feeling well, he got off work relatively
early and went home to rest at 5 p.m.)

Example 5:
This is an architect's schedule on a Sunday.
["go to sleep", "(00:00, 06:20)"], (Reason: The
first thing is of course to sleep.)
["handle the trivialities of life", "(07:18, 07:32)
"], (Reason: After getting up, he first dealt with
the trivial matters in life that had not been
resolved during the week.)
["leisure or entertainment", "(07:38, 17:27)"], (
Reason: Since today was Sunday, he didn't have to
work, so he decided to go out and have fun.)
["handle the trivialities of life", "(18:22, 19:11)
"], (Reason: After coming back in the evening, he
would take care of some chores again.)
["go to sleep", "(20:51, 23:59)"] (Reason: Since
tomorrow is Monday, go to bed early to better
prepare for the new week.)

Example 6:
This is the schedule of a day for a customer service
 specialist.
[go to work, (9:21, 16:56)], (Reason: Work dominated
 the day and was the main event of the day.)
[go home, (20:00, 23:59)], (Reason: After a day's
work and some proper relaxation, he finally got home
 at 8 o 'clock.)

Example 7:
This is the schedule of a day for a wedding event
planner.
[go to work, (11:21, 20:56)], (Reason: As a wedding
planner, her main working hours are from noon to
evening.)
[go home, (23:10, 23:30)], (Reason: After finishing
the evening's work, she went home to rest.)
[handle the trivialities of life, (23:30, 23:59)], (
Reason: Before she goes to bed, she takes care of
the trivial things in her life.)
```

```
Example 8:
This is the schedule of a day for a high school
teacher in Saturday.
[eat, (06:11, 7:28)], (Reason: He has a good habit:
have breakfast first after getting up in the morning
.)
[handle the trivialities of life, (07:48, 08:32)],
(Reason: After breakfast, he usually goes out to
deal with some life matters.)
[go home, (9:00, 11:00)], (Reason: After finishing
all the things, go home.)
[medical treatment, (13:44, 17:03)], (Reason: Today
is Saturday and he doesn't have to work, so he
decides to go to the hospital to check on his body
and some recent ailments.)
[go home, (19:00, 23:59)], (Reason: After seeing the
 doctor in the afternoon, he goes home in the
evening.)
```

C.4 Perceived Behavioral Control Reasoning

```
Now is 11:32. Based on current time, your profile
and lifestyle habits, rank your level of confidence
of next intention in completion.

Specifically, for the [go to work, go home, eat, do
shopping, do sports, excursion, leisure or
entertainment, go to sleep, medical treatment,
handle the trivialities of life, banking and
financial services, cultural institutions and events
] candidate intents, you need to give a ranking from
 most to least completion confidence.

Answer format: [intent1, intent2, intent3...]
```

C.5 Generating Next Mobility Behaviour

Generating Next Intention:

```
The arrangement of the previous n things is as
follows: <history>

What's the next arrangement today? Please output the
 event name and explain your thought process or
reasons for your choice. Answer format:["event name",
 "reasons"]
```

Generating Stay Duration:

```
How long will you spend on this arrangement?
```

```
You can consider some fragmented time, such as 3
hours plus 47 minute, and 7 hours and 13 minutes.

Please answer as a list: [x,y]. Which means x hours
and y minutes.
```