ActivityEditor: Learning to Synthesize Physically Valid Human Mobility

Anonymous ACL submission

Abstract

Human mobility modeling is indispensable for diverse urban applications. However, existing data-driven methods often suffer from data scarcity, limiting their applicability in regions where historical trajectories are unavailable or restricted. To bridge this gap, we propose **ActivityEditor**, a novel dual-LLM-agent framework designed for zero-shot cross-regional trajectory generation. Our framework decomposes the complex synthesis task into two collaborative stages. Specifically, an intention-based agent, which leverages demographic-driven priors to generate structured human intentions and coarse activity chains to ensure high-level socio-semantic coherence. These outputs are then refined by editor agent to obtain mobility trajectories through iteratively revisions that enforces human mobility law. This capability is acquired through reinforcement learning with multiple rewards grounded in real-world physical constraints, allowing the agent to internalize mobility regularities and ensure high-fidelity trajectory generation. Extensive experiments demonstrate that **ActivityEditor** achieves superior zero-shot performance when transferred across diverse urban contexts. It maintains high statistical fidelity and physical validity, providing a robust and highly generalizable solution for mobility simulation in data-scarce scenarios. Our code is available at: `https://anonymous.4open.science/r/ActivityEditor-066B`.

1 Introduction

Human mobility modeling is a cornerstone of modern urban intelligence, providing critical insights for a wide spectrum of real-world applications, including map services (Ye et al., 2011), logistics dispatching (Wen et al., 2024), urban planning (Zheng et al., 2023), and public safety monitoring (Gao et al., 2017). Despite its significance, the efficacy of these applications is heavily predicated on the availability of high-fidelity trajectory data. In practice, obtaining such data is increasingly challenging due to data scarcity arising from factors such as distribution shifts in evolving urban environments(Feng et al., 2020) and the "cold-start" problem in newly planned areas where historical records have yet to be generated(He et al., 2020). Coupled with stringent privacy regulations(Gong et al., 2025), this vacuum of mobility data has become a significant bottleneck, hindering the deployment of intelligent urban services.

In recent years, deep learning-based models (Long et al., 2023; Wang et al., 2023a; Yuan et al., 2022) have been widely applied to mobility synthesis, achieving promising results by capturing high-order transition dynamics and mining shared mobility patterns among users. However, these data-driven approaches suffer from several key drawbacks. First, their success heavily relies on the collection of large-scale private mobility data, which is often inaccessible due to stringent privacy regulations. Second, such models are inherently data-hungry and struggle to generalize in zero-shot settings, where historical trajectories for a target region are entirely absent. Finally, the prediction accuracy remains limited by the constrained sequential modeling capabilities of traditional architectures and a conspicuous lack of deep understanding regarding human daily commonsense.

While recent advancements in Large Language Models (LLMs) have shown promise in preliminary mobility prediction(Wang et al., 2023b; Feng et al., 2025b) and synthesis(Shao et al., 2024; Du et al., 2025), they continue to face critical bottlenecks. Specifically, most existing LLM-based methods still require partial local trajectories to perceive regional patterns, hindering their application in strict zero-shot scenarios. Furthermore, these methods typically rely on single-pass generation architectures that lack self-correction or auditing mechanisms, often leading to spatio-temporal inconsistencies and violations of physical laws in real-world

environments.

We propose ActivityEditor, a dual-agent framework designed for high-fidelity, zero-shot mobility synthesis. Our approach is grounded in the insight that while urban functional indicators and population-level metrics vary across cities, the underlying mapping between socio-demographic characteristics and human intentions remains invariant(Evidence can be seen in Appendix A.1). ActivityEditor takes individual profiles as input and decomposes the synthesis process into two collaborative stages. In the Intention-driven Generation stage, an agent leverages demographic priors to derive structured intentions and construct a coarse activity skeleton. This is followed by Rule-constrained Refinement, where an Editor agent iteratively enforces human mobility laws via "Add, Delete, Shift, and Replace" operations to establish logical consistency and rectify spatio-temporal ungrounding. To ensure the reliability of this refinement, we utilize Group Relative Policy Optimization (GRPO) to align the agent with real-world physical regularities. This enables the model to autonomously internalize physical laws and correct latent cognitive biases regarding human movement, allowing ActivityEditor to perform robust zero-shot generation in entirely unseen target cities by learning a universal logic of trajectory refinement.

Our contributions are summarized as follows:

- We define and address the challenge of zero-shot human mobility synthesis, enabling the generation of realistic trajectories starting solely from socio-demographic profiles. This approach effectively bypasses the dependency on private historical data and provides a scalable solution for the “cold-start” problem in newly planned urban areas.

- We propose ActivityEditor, a novel dual-LLM-agent framework that treats mobility synthesis as an iterative “Generate-then-Edit” task. By decomposing the process into intention-driven generation and rule-constrained refinement, and further employing GRPO for physical-law alignment, we establish a self-correcting mechanism that ensures both semantic coherence and spatio-temporal validity.

- Extensive experiments across diverse urban scenarios demonstrate that ActivityEditor significantly outperforms state-of-the-art baselines. Our results confirm the model’s superior zero-shot generalizability, achieving remarkable improvements in physical consistency and statistical fidelity without requiring any local trajectory data for the target regions.

2 Related Work

2.1 Mobility Modeling

Existing research can be categorized into mobility prediction and mobility synthesis. Early methods in both fields primarily utilized deep learning architectures, such as RNNs and GANs, to capture spatiotemporal dynamics (Gao et al., 2019; Feng et al., 2018; Song et al., 2016; Feng et al., 2020). However, these statistical approaches often struggle to incorporate high-level semantic knowledge, leading to suboptimal performance in complex urban scenarios. To bridge this gap, recent studies have integrated Large Language Models (LLMs) to enhance predictive capabilities (Li et al., 2024a; Wang et al., 2023b; Zhong et al., 2025; Feng et al., 2024) and generative fidelity (Wang et al., 2024b; Shao et al., 2024).Meanwhile, emerging LLM-based mobility generation frameworks (Wang et al., 2024a; Liu et al., 2024, 2026; Wu et al., 2024) have further explored agent-centric, profile-conditioned, and city-aware simulation paradigms, improving the realism and personalization of synthetic travel diaries and activity chains (Amin et al., 2025; Li et al., 2024b).

Despite their semantic depth, these methods exhibit significant limitations in zero-shot generalization and physical consistency. Most LLM-based models still exhibit sensitivity to local data distributions or require few-shot samples to perceive regional patterns (Feng et al., 2025b; Han et al., 2025; Li and Lim, 2025). Crucially, current paradigms are predominantly single-pass (forward-only) architectures that lack self-correction or auditing mechanisms. This often leads to spatio-temporal inconsistencies, such as violations of travel velocity constraints or logical gaps in activity chains. Unlike these approaches, ActivityEditor treats mobility modeling as an iterative "generate-then-edit" task, aligning synthetic trajectories with physical laws and human commonsense through a dual-agent refinement process.

2.2 LLM and Agentic Reinforcement Learning

Post-training alignment has evolved from Reinforcement Learning from Human Feedback

(RLHF) (Ouyang et al., 2022) to more efficient paradigms like Direct Preference Optimization (DPO) (Rafailov et al., 2023). Recently, Group Relative Policy Optimization (GRPO) (Guo et al., 2025) and its variants, such as DAPO (Yu et al., 2025), have emerged to enhance complex reasoning by normalizing rewards within sample groups without explicit value functions,and have further inspired multi-turn agent reinforcement learning with finer-grained turn-level optimization and reward design (Zeng et al., 2025; Wei et al., 2025). Transitioning toward Agentic RL, research frames LLMs as autonomous decision-makers that optimize long-horizon trajectories through environment feedback (Zhang et al., 2025; Zhou et al., 2025; Qi et al., 2024). Unlike standard alignment, agentic approaches emphasize iterative self-correction and tool-use reasoning(Feng et al., 2025a; Wang et al., 2025; Dong et al., 2025). ActivityEditor advances this by integrating GRPO-based alignment into a dual-agent "generate-then-edit" framework, enabling the autonomous internalization of physical regularities for zero-shot mobility synthesis.

3 Preliminary

We define the zero-shot mobility synthesis task and related concepts for use in the following section.

Socio-demographic Attributes: For each agent i, we define their background as a set of K attributes, denoted by $D_i = \{d_{1,i}, d_{2,i}, \ldots, d_{K,i}\}$. These attributes encompass both individual-level demographics (e.g., age, gender, employment) and household-level descriptors (e.g., income, household size, vehicle ownership), serving as the initial prompt seeds for mobility generation.

Human Intention: Distinct from prior works, we introduce the concept of latent intention to bridge demographics and behaviors. Let $I_{n,i} \in \mathcal{V}_I$ be the underlying motivation (e.g., *sustenance*, *education*, *recreation*) for the n-th activity of agent i. An intention chain is defined as a time-ordered sequence: $\mathcal{I}_i = \{I_{1,i}, I_{2,i}, \ldots, I_{n,i}\}$.

Activity and Activity Chain: An individual activity $A_{n,i}$ is defined as a tuple $[T_{n,i}, S_{n,i}, E_{n,i}]$, where $T_{n,i}$ denotes the activity type (semantic category), while $S_{n,i}$ and $E_{n,i}$ represent the start and end times, respectively. An activity chain $A_i = \{A_{1,i}, A_{2,i}, \ldots, A_{n,i}\}$ represents the complete sequence of events for agent i over a 24-hour period.

Mobility Trajectory: The final synthetic trajectory $Traj_i$ is expressed as a sequence of activity-location pairs: $Traj_i = \{(A_{1,i}, Z_{1,i}), \ldots, (A_{n,i}, Z_{n,i})\}$, where $Z_{n,i}$ denotes the zone-level location (e.g., Transportation Analysis Zone, TAZ) where the activity occurs.

Zero-shot Mobility Synthesis: Given a set of socio-demographic attributes D_i from an unseen target region $\mathcal{R}_{test}$, the objective is to learn a generative mapping $\mathcal{M} : D_i \rightarrow Traj_i$. The model $\mathcal{M}$ must achieve high statistical fidelity and physical validity without having observed any historical trajectory samples from $\mathcal{R}_{test}$ during the training phase.

4 Methods

4.1 Overview

As illustrated in Figure 1, ACTIVITYEDITOR comprises two core modules: the Intention-Driven Agent and the Editor Agent. Serving as the semantic architect, the Intention-Driven Agent is designed to induce the Socio-Semantic Prior by mapping demographic profiles into a activity draft. To rectify inherent Mobility Biases and instill an explicit awareness of spatio-temporal regularities, the Editor Agent functions as a refinement module that performs Add, Delete, Shift, and Replace operations. While the Intention-Driven Agent operates in a zero-shot manner to preserve universal behavioral priors, the Editor Agent is specifically optimized through a two-stage training pipeline involving SFT and GRPO to ensure the final synthesis achieves both socio-semantic coherence and strict physical validity. Detailed prompt templates and task-specific instructions for both agents are provided in Appendix A.3.4.

4.2 Intention-driven Agent

The Intention-driven Agent induces the Socio-Semantic Prior, defined as the cross-regionally similar mapping between demographics and behavioral intentions. While urban functional indicators vary, the underlying logic linking user profiles to intentions remains consistent. By leveraging a frozen LLM as a world model, the agent infers latent intentions directly from user profiles to generate foundational human mobility patterns.

Input Normalization. The agent receives a structured user profile D_i, comprising K diverse attributes such as `age_range`, `employment_status`, `relationship`, and `primary_activity`.

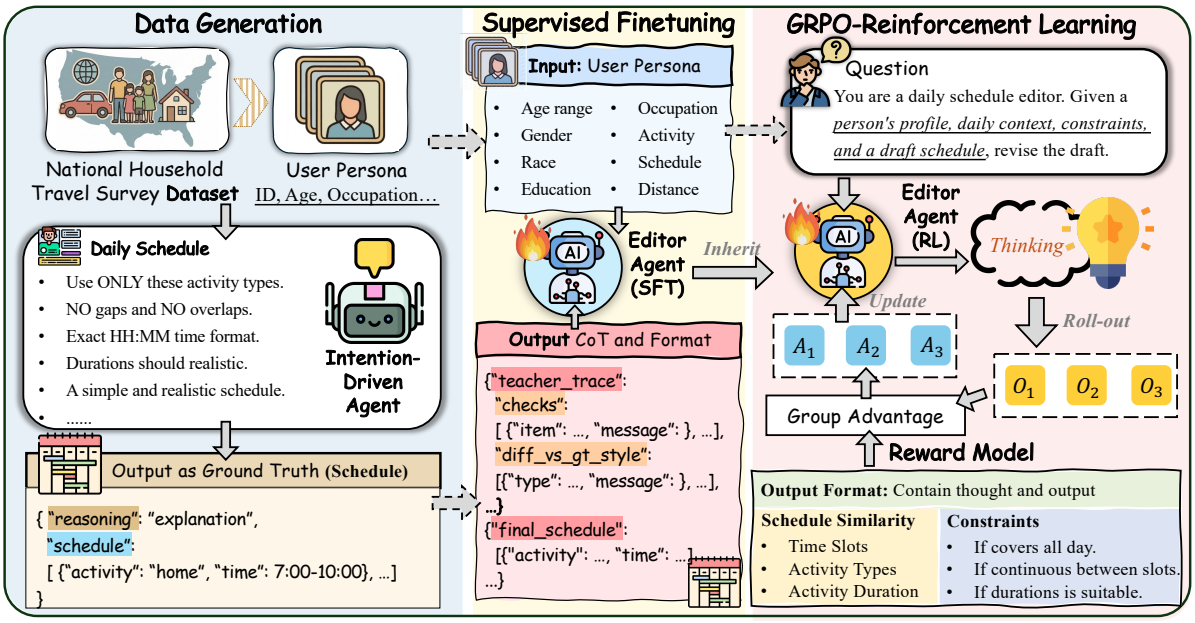


Figure 1: Overview of the ACTIVITYEDITOR framework.

Skeleton Generation. Upon receiving D_i, the agent initiates a reasoning trace within the thought space to infer the user's latent intention, denoted as V_I. By leveraging its parametric knowledge of population clustering and behavioral patterns, the agent articulates the rationale behind the user's daily demands. For example, if a user profile indicates `primary_activity = "Retired"`, the agent may explicitly reason that "the user is retired and therefore does not require a daily work commute."Guided by these inferred intentions, the agent constructs a preliminary draft schedule, denoted as S_{draft}. This draft establishes the chronological order of the activity chain and allocates rough temporal intervals.

4.3 Editor Agent

The Editor Agent acts as a rule-based optimizer to bridge the gap between abstract intentions and physical reality. While the Intention-driven Agent ensures semantic logic, the resulting S_{draft} may fail to satisfy the underlying spatio-temporal constraints of human mobility. The Editor Agent specifically injects explicit mobility regularities to rectify the spatio-temporal ungrounding inherent in LLMs. By functioning as a rule-aware auditor, it transforms these coarse drafts into strictly valid trajectories by enforcing rigorous physical and logical boundaries.

Constraint Checking. The agent first acts as a rigorous "critic" within the `<Thought>` space, comprehensively auditing the draft to identify any violations of predefined rules. As illustrated in the "Reward Model" module of Figure 1, these constraints are formalized into two major categories:

1) Hard Constraints. These are strictly enforced rules that ensure the fundamental mathematical and structural validity of the trajectory:

- *Physical Constraints:* Absolute spatiotemporal boundaries. For example, the activity chain seamlessly cover a complete 24-hour period.
- *Logical Constraints:* Inherent structural and state-transition mandates. Consecutive segments in the schedule must represent distinct states; transitioning directly from one activity to the exact same activity type is structurally invalid and must be merged into a single continuous segment.

2) Soft Constraints. These act as heuristic guidelines to evaluate the behavioral realism, socioeconomic alignment, and profile consistency of the sequence:

- *Commonsense Constraints:* Alignment with socio-demographic profiles. For example, a user explicitly profiled as *retired* should not be assigned regular, prolonged *work* activities.
- *Temporal Constraints:* Plausible duration boundaries for specific activities. For example, while sleeping or working can naturally take several hours, a *drop-off/pick-up* or *service* event should be relatively brief and cannot unrealistically span half a day.

- *Coherence Constraints:* Macroscopic sequence rationality. This prevents excessive temporal fragmentation, such as alternating between home and shopping every 10 minutes, thereby ensuring smooth and logical activity transitions.

Action and Iterative Refinement. Upon detecting any of the aforementioned violations, the agent triggers an internal ReAct loop(Yao et al., 2022). It selects specific editing operations from a predefined action space to locally modify the draft. These operations include ADD (inserting missing activities), DELETE (removing irrational events), SHIFT (adjusting start/end times to resolve overlaps), REPLACE (swapping contextually inappropriate activities), and SPLIT (dividing excessively long activities proportionally). After each execution, the agent re-evaluates the updated schedule until all conflicts are resolved. Once it confirms that the schedule satisfies all spatiotemporal and physical rules, the agent outputs the finalized, high-fidelity JSON-structured trajectory.

4.4 Training Strategy

Our training pipeline, comprising SFT and GRPO, is designed to systematically eliminate the latent Mobility Bias and hallucinations of pre-trained LLMs. This process replaces noisy, sparse internal representations with verified domain knowledge, allowing the Editor Agent to internalize the fundamental laws of human movement. Through this two-stage alignment, the model shifts from generic reasoning to a precise, rule-governed cognitive model of mobility.

Data Generation. Real-world human mobility datasets, such as travel surveys, exclusively record the final executed trajectories. They inherently lack the intermediate cognitive processes of planning, constraint checking, and conflict resolution. To bridge this gap and construct a training corpus for the editing mechanism, we employ a powerful teacher model, GPT-5.2, to perform a reverse-engineering task. Specifically, we supply the teacher model with the user demographic profile, the flawed preliminary draft generated by the Intention-driven Agent, and the corresponding ground-truth schedule. Tasked with mapping the draft to the ground truth, the teacher model infers a highly detailed Chain-of-Thought trace. This trace explicitly articulates the step-by-step logic required to identify temporal overlaps or commonsense violations in the draft, along with the precise refinement operations needed to correct them. Consequently, we successfully synthesize the missing cognitive traces, creating high-quality paired data of reasoning processes and structural outputs.

Supervised Fine-Tuning. Utilizing the synthesized reasoning dataset, we perform supervised fine-tuning to initialize the student model as the Editor Agent. This phase is fundamentally designed to solve the cold-start problem inherent in complex generation tasks. By training strictly on the paired thought traces and the final JSON formats, the student model acquires foundational domain knowledge regarding human mobility patterns. More importantly, it learns the specific structural paradigm of self-critique. The model masters how to properly open a thought space, sequentially propose editing actions such as adding, deleting, or shifting activities, and ultimately output a rigidly formatted JSON trajectory.

GRPO-Reinforcement Learning. To transcend mere data imitation and deeply internalize physical laws, we introduce GRPO in the second phase. Given a user profile, the policy model parallelly performs roll-outs to generate a group of candidate schedules. Subsequently, a programmatic, rule-based reward model evaluates these outputs without human-in-the-loop intervention. The composite reward is formulated through three explicit dimensions:

Formatting Reward (R_{fmt}): This ensures adherence to the "Generate-then-Edit" paradigm. It assigns scores based on the presence of the reasoning process (`[THOUGHT]`) and the structured output (`[JSON]`):

$$R_{\text{fmt}} = \begin{cases} 1.0, & \text{if both } \texttt{[THOUGHT]} \text{ and } \texttt{[JSON]} \text{ exist} \\ 0.5, & \text{if only one tag exists} \\ 0.0, & \text{otherwise} \end{cases} \tag{1}$$

Constraint Reward (R_{con}): This validates the physical feasibility of the trajectory. Let $\mathcal{S} = \{(s_1, e_1), (s_2, e_2), \ldots, (s_N, e_N)\}$ denote the generated schedule containing N segments, where s_i and e_i are the start and end times in minutes. R_{con} is the average of three Boolean checks:

$$R_{\text{con}} = \frac{1}{3}\left(\mathbb{I}_{\text{full}} + \mathbb{I}_{\text{cont}} + \mathbb{I}_{\text{dur}}\right) \tag{2}$$

where $\mathbb{I}_{\text{full}}$ ensures full-day coverage; $\mathbb{I}_{\text{cont}}$ enforces temporal continuity without gaps or overlaps; and

$\mathbb{I}_{\text{dur}}$ constrains each activity duration to a realistic physiological range.

Fidelity Reward (R_{sim}): This evaluates the statistical realism of the generated sequence compared to standard mobility patterns. It combines the token-level accuracy (Acc), Macro-F1 score, and the Jensen-Shannon Divergence (JSD) of activity type distributions (p_{act}) and interval length distributions (p_{int}):

$$\begin{aligned} R_{\text{sim}} = 0.40 \cdot \text{Acc} + 0.10 \cdot \text{F1}_{\text{macro}} \\ + 0.25 \cdot \left(1 - \text{JSD}(p_{\text{gen}}^{\text{act}} \parallel p_{\text{gt}}^{\text{act}})\right) \\ + 0.25 \cdot \left(1 - \text{JSD}(p_{\text{gen}}^{\text{int}} \parallel p_{\text{gt}}^{\text{int}})\right) \end{aligned} \quad (3)$$

where p_{gen} and p_{gt} denote the probability distributions of the generated schedule and the ground truth, respectively.

Based on the sum of these rewards $R = R_{\text{fmt}} + R_{\text{con}} + R_{\text{sim}}$, the group advantage is calculated to update the policy model. Through this group-relative comparison, model autonomously learns to increase generation probability of high-quality CoTs, thereby thoroughly internalizing spatiotemporal constraints and self-editing capabilities.

5 Experiment

5.1 Experimental Setup

Datasets. We conduct experiments using the 2017 National Household Travel Survey (NHTS), a nationwide travel survey covering approximately 129,696 households and 264,234 persons (McGuckin and Fucci, 2018). To evaluate geographical generalization, we extract subsets from four states: California, Arizona, Georgia, and Oklahoma. Each user persona contains 12 sociodemographic features. All raw travel records are standardized into 10 semantic activity categories. Further details regarding the specific definitions of these metrics are provided in Appendix A.1

Baselines Setup. We benchmark our approach against three categories of baselines. The first category comprises traditional deep learning models, including DeepMove (Feng et al., 2018) and LSTPM (Sun et al., 2020), which rely heavily on historical sequential modeling mechanisms. The second category includes prompt-based and retrieval-augmented generation frameworks, where we compare with CoPB (Shao et al., 2024) and RAGHome (Liu et al., 2024). These methods leverage advanced prompt engineering and retrieval-augmented generation to synthesize schedules. The third category evaluates zero-shot large language models. We assess representative proprietary and open-source models, including ChatGPT-5.2, Qwen3-8b, and LLaMA3.1-8b, under both single-pass generation and dual-agent architectural modes. Further details regarding the specific definitions of these metrics are provided in Appendix A.2.

Implementation Details. We employ GPT-5.2 as the teacher model to generate Chain-of-Thought (CoT) reasoning data. Based on data, we initialize the editor agent by Supervised Fine-Tuning (SFT) on Qwen3-8B with LoRA ($r = 32, \alpha = 64$) for 3 epochs using a learning rate of 5×10^{-5}. We further optimize the SFT model with Group Relative Policy Optimization (GRPO), while keeping the same LoRA configuration. For GRPO, we train the model for 1000 steps with a learning rate of 1×10^{-5}, using 8 roll-out samples per prompt and a sampling temperature of 0.7. Further details regarding the specific definitions of these metrics are provided in Appendix A.4.

Evaluation Metrics To evaluate generated activity chains, we use a three-level framework of 12 metrics, summarized in Table 3. Sequence Similarity Metrics (e.g., accuracy, f1_score, and edit_dist) measure structural fidelity and sequential consistency with respect to reference diaries. Temporal Alignment Metrics (e.g., macro_int and micro_hour) assess whether generated schedules match reference timing and duration patterns at a 15-minute resolution. Distributional Consistency Metrics (e.g., data_jsd and act_type) use Jensen-Shannon Divergence to evaluate whether synthesized populations recover real-world aggregate regularities and semantic distributions. Detailed metric definitions are provided in Appendix A.5.

Table 1 presents the comprehensive performance comparison of all models across the four states. Beyond the absolute numerical superiority, the results reveal fundamental differences in how various architectures handle the mobility synthesis task.

5.2 Main Results

Traditional deep learning baselines, such as DeepMove and LSTPM, degrade sharply in the zero-shot setting because they rely heavily on historical trajectory sequences. Without past mobility records, they struggle to map static demographic attributes to dynamic behaviors. Prompt-based and retrieval-augmented methods, including CoPB and RAGHome, partially alleviate this issue by exploiting the language understanding of large language

Method	California			Arizona			Georgia			Oklahoma		
	Acc ↑	*Mint* ↓	*Atype* ↓	*Acc* ↑	*Mint* ↓	*Atype* ↓	*Acc* ↑	*Mint* ↓	*Atype* ↓	*Acc* ↑	*Mint* ↓	*Atype* ↓
Traditional Baselines												
◇ DeepMove	0.554	0.660	0.229	0.521	0.666	0.252	0.448	0.657	0.264	0.472	0.676	0.281
◇ LSTPM	0.402	0.705	0.290	0.420	0.691	0.300	0.379	0.693	0.288	0.376	0.697	0.287
General LLM Baselines												
◇ ChatGPT	0.660	0.425	0.199	0.664	0.399	0.195	0.583	0.403	0.201	0.616	0.404	0.158
◇ Qwen	0.553	0.535	0.228	0.551	0.598	0.194	0.511	0.583	0.238	0.542	0.570	0.167
◇ LLaMA	0.526	0.576	0.291	0.524	0.643	0.248	0.486	0.624	0.305	0.515	0.611	0.213
Specific LLM												
◇ CoPB	0.703	0.577	0.141	0.674	0.579	0.157	0.602	0.603	**0.152**	0.627	0.567	0.140
◇ RAGHome	0.679	0.556	0.183	0.658	0.548	0.198	0.593	0.574	0.167	0.614	0.528	0.142
Fine-tuned LLMs												
◇ **GRPO_LLaMA**	0.712	0.281	0.141	0.742	0.322	0.169	**0.661**	**0.311**	0.165	**0.689**	**0.252**	0.132
◇ **GRPO_Qwen**	**0.734**	**0.258**	**0.118**	**0.780**	**0.251**	**0.089**	0.655	0.319	0.171	0.659	0.265	**0.099**

Table 1: Performance comparison across four states under different models and settings.

Model	CA → CA			CA → AZ			CA → GA			CA → OK			AZ → CA		
	Acc ↑	*Mint* ↓	*Atype* ↓	*Acc* ↑	*Mint* ↓	*Atype* ↓	*Acc* ↑	*Mint* ↓	*Atype* ↓	*Acc* ↑	*Mint* ↓	*Atype* ↓	*Acc* ↑	*Mint* ↓	*Atype* ↓
SFT_CoPB	0.743	0.289	0.154	0.682	0.420	0.127	0.642	0.303	0.153	0.711	0.325	**0.114**	0.654	0.371	0.109
SFT_RAGHome	0.691	0.349	0.176	0.671	0.322	0.183	0.643	0.324	0.134	0.672	0.246	0.120	0.636	0.287	0.151
SFT_LLaMA	0.724	0.214	0.110	0.689	0.256	0.158	0.632	0.253	0.195	0.650	0.227	0.165	0.653	0.228	0.141
SFT_Qwen	0.753	0.231	0.104	0.745	0.239	0.113	0.708	0.248	0.130	0.714	0.241	0.127	0.699	0.214	0.097
GRPO_LLaMA	0.732	0.248	0.122	0.720	0.260	0.131	0.681	0.276	0.147	0.690	0.268	0.142	0.671	0.233	0.118
GRPO_Qwen	**0.761**	**0.202**	**0.094**	**0.768**	**0.215**	**0.101**	**0.736**	**0.229**	**0.119**	**0.743**	**0.223**	0.116	**0.721**	**0.191**	**0.086**

Table 2: Cross-state transfer performance. Models (SFT_LLaMA, SFT_Qwen, GRPO_LLaMA, GRPO_Qwen) are implemented on the multi-agent architecture. CA, AZ, GA, and OK denote California, Arizona, Georgia, and Oklahoma, respectively.

Category	Metrics
Sequence Similarity	accuracy (*Acc*), f1_score, edit_dist, bleu_score
Temporal Alignment	macro_hour, micro_hour, micro_int, macro_int (*Mint*)
Distributional Consistency	data_jsd, act_type (*Atype*), uni_act_type, traj_len

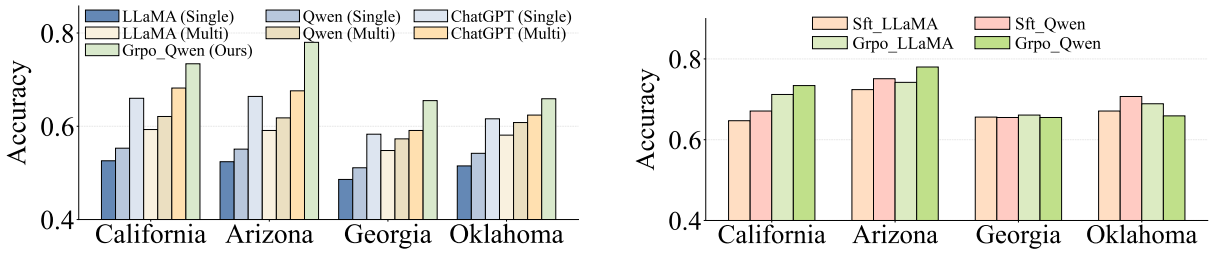
Table 3: Evaluation metrics.

models, but their performance remains limited. Although prompts can steer the general structure of daily routines, they lack an explicit mechanism to enforce physical consistency, leading to spatiotemporal conflicts such as overlapping activities or incomplete 24-hour schedules.

Our fine-tuned models overcome these limitations by moving from constrained generation to active editing. In Table 1, Grpo_Qwen and Grpo_LLaMA achieve state-of-the-art performance across all regions. In particular, Grpo_Qwen not only delivers the highest temporal accuracy but also minimizes the Jensen-Shannon Divergence metrics (macro_int and act_type). These results show that our framework goes beyond generating plausible activities, enabling the synthesis of personalized, statistically accurate, and physically consistent daily schedules.

5.3 Zero-shot Transferability Analysis

Cold-start remains a fundamental challenge in urban mobility modeling, as unseen regions often lack sufficient historical data. To test whether our framework learns transferable mobility patterns rather than state-specific shortcuts, we conduct cross-state zero-shot transfer experiments. Models trained on California are directly test on California, Arizona, Georgia, and Oklahoma without domain adaptation, and we consider the reverse setting of Arizona → California. As shown in Table 2, our method outperforms task-specific baselines such



(a) Single vs. Multi performance

(b) SFT vs. GRPO performance

Figure 2: Ablation studies on the proposed ActivityEditor.

as CoPB (Shao et al., 2024) and RAGHome (Liu et al., 2024), while maintaining stable macro-level behavior statistics across target states.

These results show that our rule-guided framework captures transferable mobility logic instead of memorizing localized geographic patterns. With California as the source domain, performance drops on other states remain limited, and `Multi_agent_grpo_Qwen` delivers the best overall results in most settings. Arizona → California performs worse than California → Arizona, likely because Arizona offers fewer training samples for learning transferable decision patterns. Even so, the relative ranking of model variants remains largely unchanged, demonstrating the robustness of our framework under data-scarce source domains.

5.4 Ablation Study on Model Designs

To disentangle the contributions of our specific architectural choices and training pipelines, we perform an in-depth ablation analysis based on the results in Table 2.

Architecture Ablation on Generation Modes. We evaluated general zero-shot large language models under both single-agent and multi-agent configurations. Across ChatGPT, Qwen, and LLaMA, the multi-agent mode consistently and noticeably outperforms the single-pass generation in Figure 2a. This consistent performance gap highlights a crucial structural insight: daily mobility scheduling is essentially a constrained optimization problem. A single-pass autoregressive generation accumulates logical errors over a long horizon, leading to hallucinations. Introducing a multi-agent auditing mechanism provides the necessary reflection space to iteratively resolve these constraints, validating our core motivation for designing a Generate-then-Edit pipeline.

Training Strategy Ablation on SFT and GRPO. Figure 2b and Table 1 show that SFT and GRPO play complementary roles in capability acquisition. SFT mainly provides statistical alignment, allowing the model to learn the mapping between demographic attributes and activity distributions and achieve competitive divergence scores. However, imitation alone is insufficient to reliably enforce hard physical constraints, leaving spatiotemporal conflicts unresolved. GRPO addresses this limitation by introducing rule-based rewards, which drive the model to internalize physical constraints and improve logical consistency during reasoning. This shift from SFT to GRPO reflects the transition from statistical fitting to constraint-aware reasoning, yielding the best overall accuracy. For the latter two states, however, the gap between SFT and GRPO becomes much smaller. This is mainly due to the extremely limited training data in these states, which makes GRPO more prone to overfitting and reduces its advantage over SFT. This result further validates the necessity and effectiveness of our Zero-shot Transferability Analysis, especially in low-resource settings.

6 Conclusion

We presented ActivityEditor, a dual-agent framework designed for high-fidelity, zero-shot mobility synthesis in cold-start and cross-state scenarios. By integrating intention-driven generation with rule-constrained refinement, our method captures universal mobility logic rather than merely memorizing localized patterns. Experimental results demonstrate that ActivityEditor consistently outperforms state-of-the-art baselines, achieving superior statistical fidelity and physical validity across diverse urban contexts. Future work will focus on developing a unified mobility foundation framework that integrates agentic reasoning with human mobility laws to support trajectory generation, completion, and prediction within a single modeling paradigm.

Limitations

Despite its promising performance, several limitations of ACTIVITYEDITOR warrant further investigation:

- **Spatial Granularity:** Currently, the framework focuses on generating zone-level activity chains. It lacks the capability to synthesize precise POI-level coordinates for micro-level urban analysis. Future research will explore the integration of external POI-retrieval tools and urban perception modules to assist the LLM in achieving more fine-grained spatial generation.

- **Computational Efficiency:** The multi-turn iterative refinement between the Intention-driven and Editor agents incurs significant computational overhead, which may limit its scalability for massive population simulations. Knowledge distillation into smaller language models or the optimization of reinforcement learning efficiency represents a viable direction to reduce inference latency.

- **Robustness and Hallucination:** As an LLM-based framework, the output of ACTIVITYEDITOR is not fully controllable. While we utilize a structured parser to extract the expected context, it may occasionally fail. Furthermore, due to the inherent potential for hallucination, the agent may generate "phantom" addresses that do not exist in the real world. While verifying this against a valid location list is feasible in controlled experiments, ensuring absolute robustness in open-world urban applications remains a significant challenge.

Ethical Considerations

All human mobility data used in the experiments come from publicly available open-source datasets(McGuckin and Fucci, 2018). We do not attempt to extract any personal information from these datasets.

AI Usage

We state that AI-assisted technologies were used strictly for the purpose of language polishing and grammar correction to enhance the clarity of our writing. The authors take full responsibility for the integrity of the work, including the validity of all experimental results and the accuracy of all references.

References

Sepehr Golrokh Amin, Devin Rhoads, Fatemeh Fakhrmoosavi, Nicholas E Lownes, and John N Ivan. 2025. Generating individual travel diaries using large language models informed by census and land-use data. *arXiv preprint arXiv:2509.09710*.

Guanting Dong, Yifei Chen, Xiaoxi Li, Jiajie Jin, Hongjin Qian, Yutao Zhu, Hangyu Mao, Guorui Zhou, Zhicheng Dou, and Ji-Rong Wen. 2025. Tool-star: Empowering llm-brained multi-tool reasoner via reinforcement learning. *arXiv preprint arXiv:2505.16410*.

Yuwei Du, Jie Feng, Jian Yuan, and Yong Li. 2025. Cams: A citygpt-powered agentic framework for urban human mobility simulation. *arXiv preprint arXiv:2506.13599*.

Jiazhan Feng, Shijue Huang, Xingwei Qu, Ge Zhang, Yujia Qin, Baoquan Zhong, Chengquan Jiang, Jinxin Chi, and Wanjun Zhong. 2025a. Retool: Reinforcement learning for strategic tool use in llms. *arXiv preprint arXiv:2504.11536*.

Jie Feng, Yuwei Du, Jie Zhao, and Yong Li. 2025b. 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, Yong Li, Chao Zhang, Funing Sun, Fanchao Meng, Ang Guo, and Depeng Jin. 2018. Deepmove: Predicting human mobility with attentional recurrent networks. In *Proceedings of the 2018 world wide web conference*, pages 1459–1468.

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.

Shanshan Feng, Haoming Lyu, Fan Li, Zhu Sun, and Caishun Chen. 2024. Where to move next: Zero-shot generalization of llms for next poi recommendation. In *2024 ieee conference on artificial intelligence (cai)*, pages 1530–1535. IEEE.

Qiang Gao, Fan Zhou, Goce Trajcevski, Kunpeng Zhang, Ting Zhong, and Fengli Zhang. 2019. Predicting human mobility via variational attention. In *The world wide web conference*, pages 2750–2756.

Qiang Gao, Fan Zhou, Kunpeng Zhang, Goce Trajcevski, Xucheng Luo, and Fengli Zhang. 2017. Identifying human mobility via trajectory embeddings. In *IJCAI*, volume 17, pages 1689–1695.

Letian Gong, Shengnan Guo, Yan Lin, Yichen Liu, Erwen Zheng, Yiwei Shuang, Youfang Lin, Jilin Hu, and Huaiyu Wan. 2025. Stcdm: Spatio-temporal contrastive diffusion model for check-in sequence

generation. *IEEE Transactions on Knowledge and Data Engineering*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Chonghua Han, Yuan Yuan, Yukun Liu, Jingtao Ding, Jie Feng, and Yong Li. 2025. Unimove: A unified model for multi-city human mobility prediction. *arXiv preprint arXiv:2508.06986*.

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

Kunrong Li and Kwan Hui Lim. 2025. Rallm-poi: Retrieval-augmented llm for zero-shot next poi recommendation with geographical reranking. In *Pacific Rim International Conference on Artificial Intelligence*, pages 210–218. Springer.

Peibo Li, Maarten de Rijke, Hao Xue, Shuang Ao, Yang Song, and Flora D Salim. 2024a. Large language models for next point-of-interest recommendation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1463–1472.

Siyu Li, Toan Tran, Haowen Lin, John Krumm, Cyrus Shahabi, Lingyi Zhao, Khurram Shafique, and Li Xiong. 2024b. Geo-llama: Leveraging llms for human mobility trajectory generation with spatiotemporal constraints. *arXiv preprint arXiv:2408.13918*.

Qi Liu, Can Li, and Wanjing Ma. 2026. Gatsim: Urban mobility simulation with generative agents. *Transportation Research Part C: Emerging Technologies*, 186:105576.

Yifan Liu, Xishun Liao, Haoxuan Ma, Brian Yueshuai He, Chris Stanford, and Jiaqi Ma. 2024. Human mobility modeling with household coordination activities under limited information via retrieval-augmented llms. *arXiv preprint arXiv:2409.17495*.

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.

Nancy McGuckin and Anthony Fucci. 2018. Summary of travel trends: 2017 national household travel survey.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Zehan Qi, Xiao Liu, Iat Long Iong, Hanyu Lai, Xueqiao Sun, Wenyi Zhao, Yu Yang, Xinyue Yang, Jiadai Sun, Shuntian Yao, and 1 others. 2024. Webrl: Training llm web agents via self-evolving online curriculum reinforcement learning. *arXiv preprint arXiv:2411.02337*.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741.

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 preprint arXiv:2402.09836*.

Xuan Song, Hiroshi Kanasugi, and Ryosuke Shibasaki. 2016. Deeptransport: Prediction and simulation of human mobility and transportation mode at a city-wide level. In *Proceedings of the twenty-fifth international joint conference on artificial intelligence*, pages 2618–2624.

Ke Sun, Tieyun Qian, Tong Chen, Yile Liang, Quoc Viet Hung Nguyen, and Hongzhi Yin. 2020. Where to go next: Modeling long-and short-term user preferences for point-of-interest recommendation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 214–221.

Dongsheng Wang, Shen Gao, Chengrui Huang, Yuxi Huang, Ruixiang Feng, and Shuo Shang. 2025. Tool4poi: A tool-augmented llm framework for next poi recommendation. *arXiv preprint arXiv:2511.06405*.

Huandong Wang, Qizhong Zhang, Yuchen Wu, Depeng Jin, Xing Wang, Lin Zhu, and Li Yu. 2023a. Synthesizing human trajectories based on variational point processes. *IEEE Transactions on Knowledge and Data Engineering*, 36(4):1785–1799.

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

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

Yu Wang, Tongya Zheng, Yuxuan Liang, Shunyu Liu, and Mingli Song. 2024b. Cola: Cross-city mobility transformer for human trajectory simulation. In *Proceedings of the ACM Web Conference 2024*, pages 3509–3520.

Quan Wei, Siliang Zeng, Chenliang Li, William Brown, Oana Frunza, Wei Deng, Anderson Schneider, Yuriy Nevmyvaka, Yang Katie Zhao, Alfredo Garcia, and 1 others. 2025. Reinforcing multi-turn reasoning in llm agents via turn-level reward design. *arXiv preprint arXiv:2505.11821*.

Haomin Wen, Youfang Lin, Lixia Wu, Xiaowei Mao, Tianyue Cai, Yunfeng Hou, Shengnan Guo, Yuxuan Liang, Guangyin Jin, Yiji Zhao, and 1 others. 2024. A survey on service route and time prediction in instant delivery: Taxonomy, progress, and prospects. *IEEE Transactions on Knowledge and Data Engineering*.

Ziqing Wu, Zhu Sun, Dongxia Wang, Lu Zhang, Jie Zhang, and Yew Soon Ong. 2024. Mrp-llm: Multitask reflective large language models for privacy-preserving next poi recommendation. *arXiv preprint arXiv:2412.07796*.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. In *The eleventh international conference on learning representations*.

Mao Ye, Peifeng Yin, Wang-Chien Lee, and Dik-Lun Lee. 2011. Exploiting geographical influence for collaborative point-of-interest recommendation. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 325–334.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, and 1 others. 2025. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*.

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.

Siliang Zeng, Quan Wei, William Brown, Oana Frunza, Yuriy Nevmyvaka, Yang Katie Zhao, and Mingyi Hong. 2025. Reinforcing multi-turn reasoning in llm agents via turn-level credit assignment. In *ICML 2025 Workshop on Computer Use Agents*.

Guibin Zhang, Hejia Geng, Xiaohang Yu, Zhenfei Yin, Zaibin Zhang, Zelin Tan, Heng Zhou, Zhongzhi Li, Xiangyuan Xue, Yijiang Li, and 1 others. 2025. The landscape of agentic reinforcement learning for llms: A survey. *arXiv preprint arXiv:2509.02547*.

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.

Lin Zhong, Lingzhi Wang, Xu Yang, and Qing Liao. 2025. Comapoi: A collaborative multi-agent framework for next poi prediction bridging the gap between trajectory and language. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1768–1778.

Yifei Zhou, Song Jiang, Yuandong Tian, Jason Weston, Sergey Levine, Sainbayar Sukhbaatar, and Xian Li. 2025. Sweet-rl: Training multi-turn llm agents on collaborative reasoning tasks. *arXiv preprint arXiv:2503.15478*.

A Appendix

A.1 Dataset Description and Analysis

To comprehensively evaluate our proposed framework, we utilize the 2017 National Household Travel Survey (NHTS) dataset. This dataset provides rich socio-demographic profile information paired with highly detailed 24-hour activity and travel diaries for residents across the United States. We specifically extract subsets from four representative states: California, Arizona, Georgia, and Oklahoma. These states encompass high diversity in urban structures, economic development levels, and population densities. As illustrated in Figure 3, the volume of available trajectory data varies significantly across different regions. California possesses over 56,000 home activity records, followed by Georgia with over 25,000, Arizona with approximately 6,900, and Oklahoma with roughly 3,100. This severe data imbalance highlights a critical real-world challenge: many newly planned or smaller regions lack sufficient historical data to support the training of reliable predictive models. This discrepancy directly motivates our zero-shot cross-regional generation task, which aims to transfer mobility knowledge from data-rich source domains to data-scarce target domains.

Despite the vast differences in data scale and geospatial layouts, the macroscopic statistical features of the dataset reveal that fundamental human mobility patterns remain highly consistent, which strongly validates the theoretical feasibility of our transfer learning approach. Figure 4 presents the stacked percentage distribution of activity types across the four states. The proportion of human activities remains universally stable across different regions, with home, work, and shopping consistently dominating daily routines. Similarly, Figure 6 demonstrates that the distribution of daily activity chain lengths follows a nearly identical right-skewed pattern across all states, typically peaking between three to five activities per person. This cross-regional consistency provides empirical support for our core assumption: human mobility intentions are fundamentally driven by socio-demographic attributes rather than determined by local spatial geometric features. It is exactly this ubiquitous similarity that enables our intention-driven agent to successfully generalize to unseen cities in a zero-shot manner.

Furthermore, the dataset clearly reflects the strict physical boundaries and physiological constraints governing human behavior. Figure 5 visualizes the average duration of various activities across different states. The temporal patterns strictly adhere to human commonsense: rigid activities such as sleeping at home or working typically span hundreds of minutes, whereas transitional or service activities, such as dropping off children, generally remain stable under forty minutes. These universal temporal regularities do not fluctuate across state borders. This observation fundamentally underpins our design of the rule-based reward model in the Group Relative Policy Optimization phase. By transforming these ubiquitous physiological and temporal constraints into programmatic reward signals, the editor agent is able to autonomously internalize the authentic logic of mobility behavior, thereby eliminating logical hallucinations without requiring any target-domain training data.

A.2 Detailed Description of Baselines

To comprehensively evaluate the performance of the proposed framework, four representative baseline methods are selected for comparison. These methods are divided into two categories: traditional deep learning baselines and large language model baselines.

DeepMove (Feng et al., 2018): DeepMove is a classic human mobility prediction model. It employs an attentional recurrent neural network to model sequential transition patterns and multi-level periodic behaviors in users' historical trajectories, thereby enabling the prediction of future mobility behavior.

LSTPM (Sun et al., 2020): LSTPM is designed to capture both long-term and short-term user mobility preferences. By combining a context-aware non-local network with a geo-dilated recurrent neural network, it effectively models users' long-term preferences and short-term dynamic changes in point-of-interest visits.

CoPB (Shao et al., 2024): CoPB is grounded in the psychological Theory of Planned Behavior and uses a structured prompt engineering workflow to guide large language models to first reason about users' subjective beliefs and behavioral intentions, and then generate daily activity sequences, thereby improving the logical consistency and plausibility of mobility generation.

RAGHome (Liu et al., 2024): RAGHome introduces a retrieval-augmented generation mechanism that retrieves historical mobility records of individuals with characteristics similar to the target user

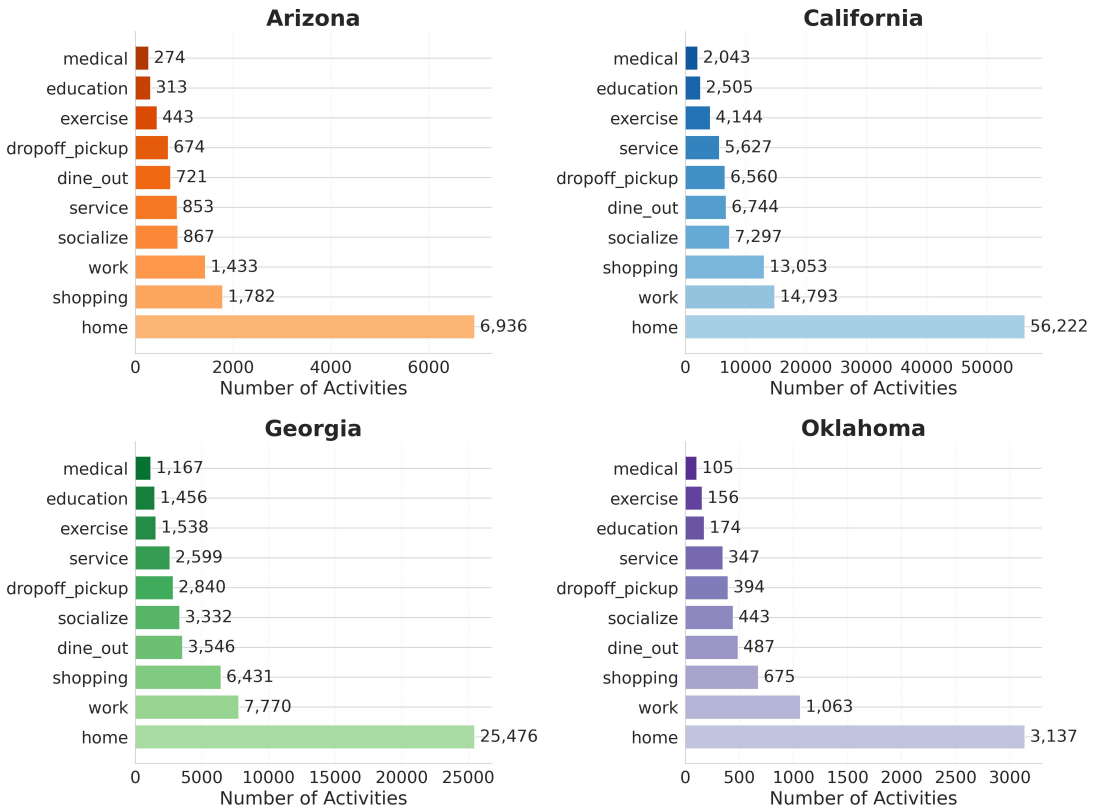


Figure 3: Activity number distribution by state.

and incorporates them as contextual examples into the large language model, thereby enhancing generation quality and realism in complex spatiotemporal tasks.

A.3 Prompt Examples

For completeness, this subsection summarizes the core prompts used throughout our trajectory generation framework.

A.3.1 Prompt of Intention-driven Agent

```
You are a daily schedule planner. Given a
    person's profile, generate a complete
    and realistic 24-hour schedule with
    exact start/end times.

RULES:
- Use ONLY these activity types: home, work,
     education, shopping, service, medical,
    dine_out, socialize, exercise,
    dropoff_pickup
- Schedule MUST cover full 24 hours: start
    at 00:00, end at 24:00, NO gaps or
    overlaps
- Must start and end with "home"
- Consecutive activities must connect (
    end_time of one = start_time of the next
    )
- Durations must be realistic based on the
    person's profile

OUTPUT FORMAT:
{
  "reasoning": "Brief explanation of the
     schedule choices",
  "schedule": [
    {"activity": "home", "start_time":
     "00:00", "end_time": "07:30"},
    {"activity": "work", "start_time":
     "07:30", "end_time": "17:00"},
    {"activity": "home", "start_time":
     "17:00", "end_time": "24:00"}
  ]
}

USER MESSAGE FORMAT:
Generate a complete 24-hour schedule for
    this person:

- Age range: {age_range}
- Gender: {gender}
- Race: {race}
- Education: {education}
- Employment status: {employment_status}
- Work schedule: {work_schedule}
- Occupation: {occupation}
- Primary activity: {primary_activity}
- Work from home: {work_from_home}
- Driver on travel day: {
    driver_on_travel_day}
- Distance to work (miles): {
    distance_to_work_miles}
- Work state: {work_state}
```

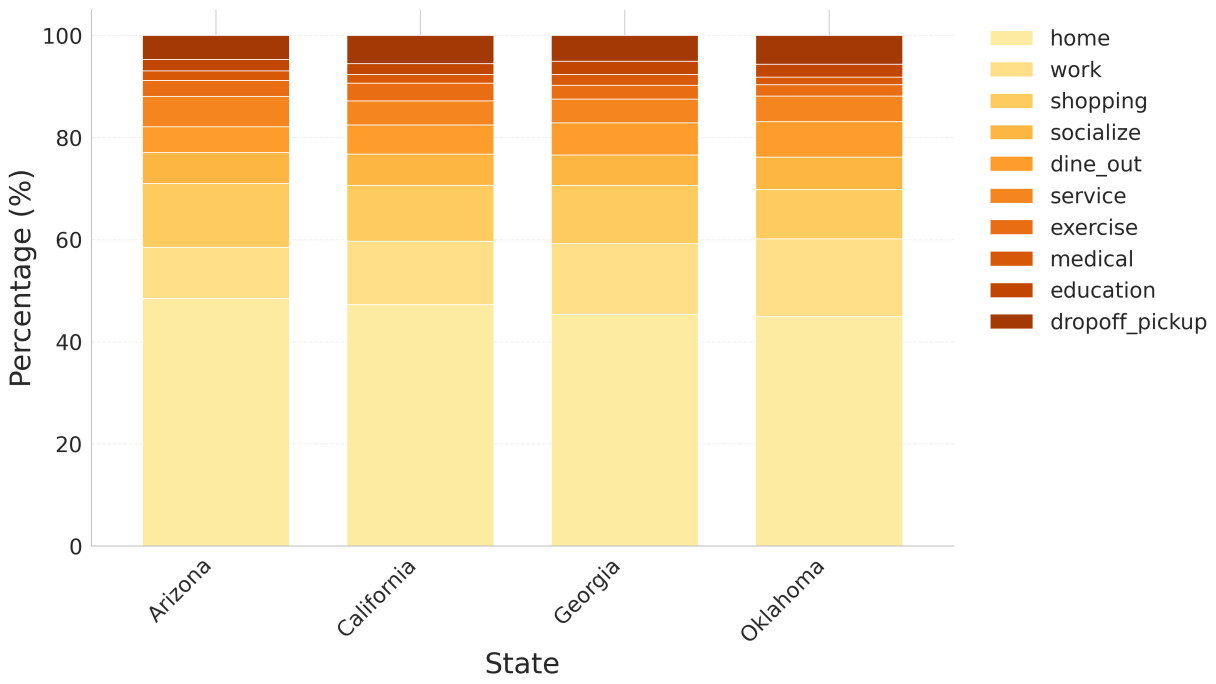


Figure 4: Activity type distribution by state.

A.3.2 Prompt of Editor Agent

```
You are an Editor Agent for daily schedule
    refinement.

Your task:
1. Check the INITIAL SCHEDULE against 5
    constraint types:
   Physical, Logical, Common Sense, Temporal,
     and Coherence
2. Identify any violations in the draft
3. Apply edit operations from the action
    space:
   ADD / DELETE / SHIFT / REPLACE / SPLIT
4. Output the final constraint-satisfying
    schedule

OUTPUT FORMAT:
[THOUGHT]
Constraint Checking:
1. Physical Constraints (Hard): overlaps, 24
    h coverage, ends at 24:00
2. Logical Constraints (Hard): starts/ends
    at home, starts at 00:00, consecutive
    identical activities must be merged
3. Common Sense Constraints (Soft):
    activities match socio-demographic
    profile
4. Temporal Constraints (Soft): realistic
    durations for each activity type
5. Coherence Constraints (Soft): logical
    transitions, not over-fragmented

Edit Operations Applied:
- ADD: activity 'Y' at time HH:MM (reason)
- DELETE: activity 'X' at index N (reason)
- SHIFT: activity 'Z' start/end time
    adjustment (reason)
- REPLACE: activity 'A' -> 'B' (reason)
- SPLIT: activity 'X' divided into two
    segments (reason)

Final Result: All constraints satisfied /
    Violations remain
[/THOUGHT]

[JSON]
[final schedule as JSON array]
[/JSON]

USER MESSAGE FORMAT:
PERSON PROFILE:
{user_profile as JSON}

INITIAL SCHEDULE:
{initial_schedule as JSON}
```

A.3.3 Prompt of Data Generation Teacher Model

```
SYSTEM PROMPT:
You are a schedule planning expert. You
    understand human behavior patterns and
    generate realistic daily schedules.

USER MESSAGE FORMAT:
You are a Critic and Editor Agent. Your job
    is to validate and refine a daily
    schedule.

PERSON PROFILE:
{user_profile as JSON}

INITIAL SCHEDULE (from unified 2-stage
    generator):
{initial_schedule as JSON}

GROUND TRUTH REFERENCE (for comparison):
```

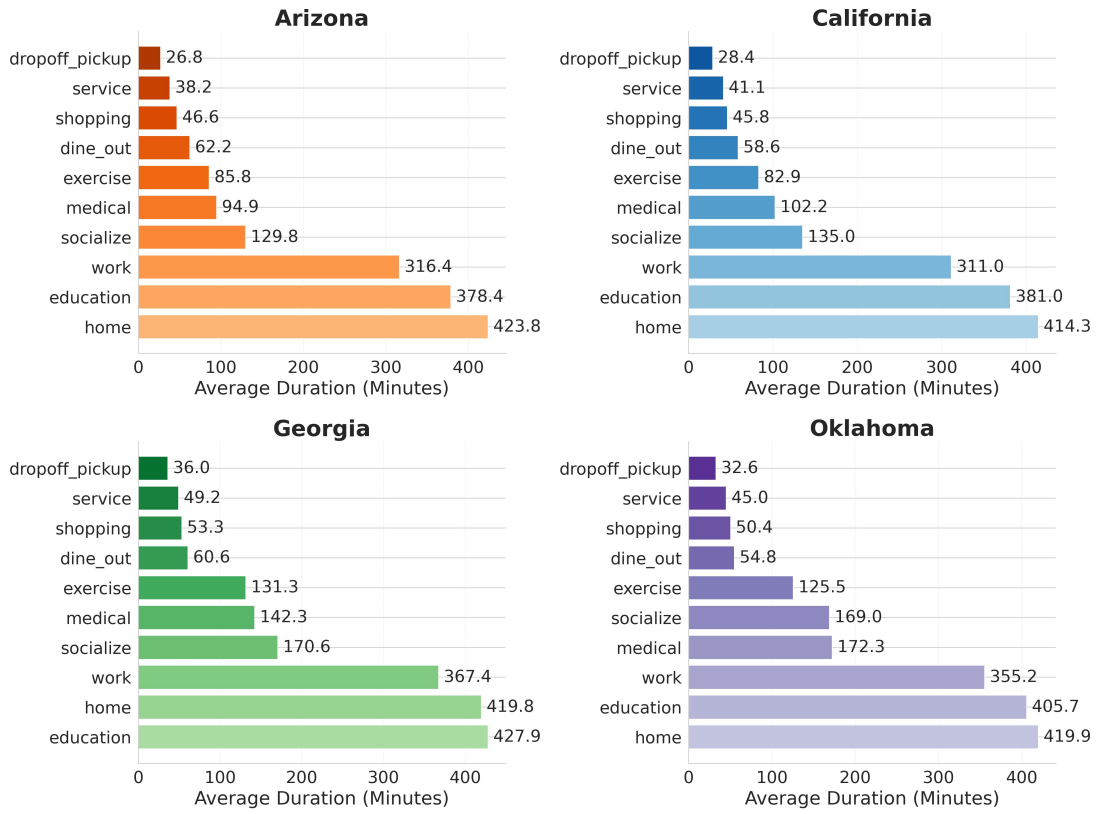


Figure 5: Activity duration by state.

```
{ground_truth_schedule as JSON}

YOUR TASK:
Act as the Editor. Check constraints on the
    INITIAL SCHEDULE, identify violations,
    and apply edits to match the GROUND
    TRUTH.

OUTPUT FORMAT:
[THOUGHT]
Constraint Checking:
1. Physical (Hard): overlaps? 24h coverage?
    ends at 24:00? -> Yes / No
2. Logical (Hard): starts/ends home? starts
    at 00:00? -> Yes / No
3. Common Sense (Soft): activities match
    profile? -> Yes / No
4. Temporal (Soft): realistic durations? ->
    Yes / No
5. Coherence (Soft): logical flow? not
    fragmented? -> Yes / No

Edits to Match Ground Truth:
If already matches: No edits needed
Otherwise list each operation:
- DELETE: 'X' at idx N (reason)
- ADD: 'Y' at time HH:MM (reason)
- SHIFT: 'Z' time adjustment (reason)
- REPLACE: 'A' -> 'B' (reason)

Final Result:
All constraints satisfied after edits? Yes /
     No
[/THOUGHT]

[JSON]
[final schedule - must match the GROUND
    TRUTH]
[/JSON]

Be thorough in checking each constraint on
    the INITIAL SCHEDULE.
Show edit operations explicitly if
    violations are found.
```

A.3.4 Prompt of SFT Student Model

```
You are a Critic and Editor Agent for daily
    schedule refinement.

Your task:
1. Check the INITIAL SCHEDULE against 5
    constraint types:
   Physical, Logical, Common Sense, Temporal,
     and Coherence
2. Identify any violations
3. Apply edit operations:
   ADD / DELETE / SHIFT / REPLACE
4. Output the refined schedule

OUTPUT FORMAT:
[THOUGHT]
Constraint Checking:
1. Physical Constraints (Hard): overlaps, 24
    h coverage, ends at 24:00
2. Logical Constraints (Hard): starts/ends
    at home, starts at 00:00
```

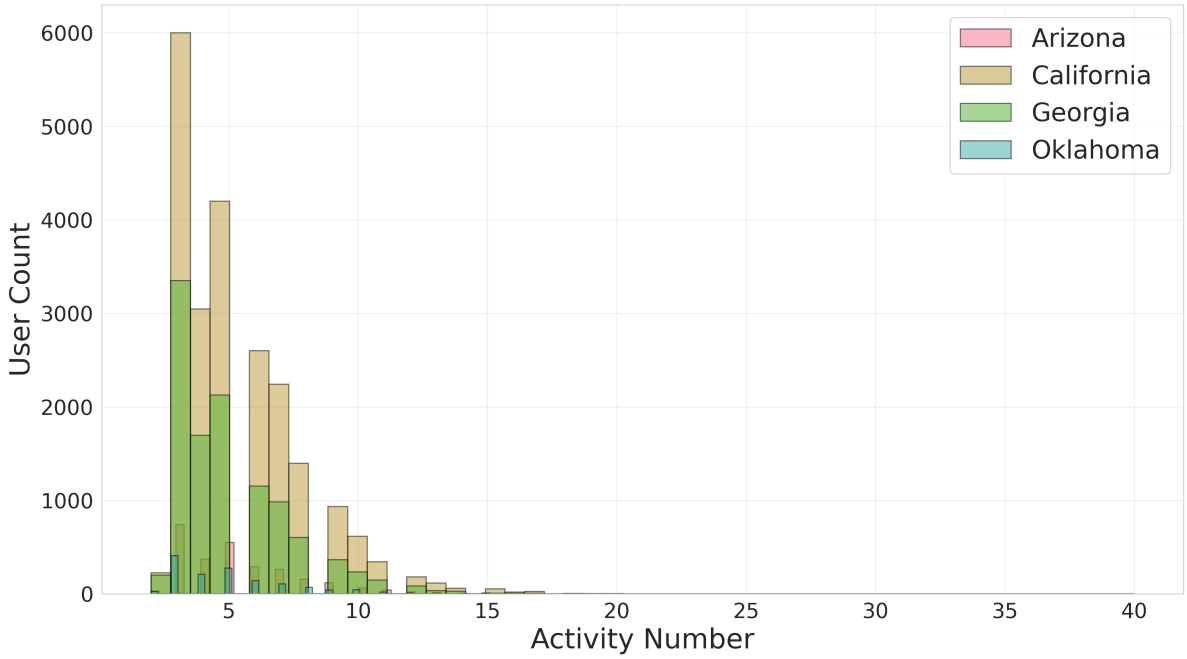


Figure 6: Chain length distribution by state.

```
3. Common Sense Constraints (Soft): age/
    employment appropriate activities
4. Temporal Constraints (Soft): realistic
    durations
5. Coherence Constraints (Soft): logical
    transitions, not over-fragmented

Applying Edit Operations:
- DELETE: activity 'X' at index N (reason)
- ADD: activity 'Y' at time HH:MM (reason)
- SHIFT: activity time adjustment (reason)
- REPLACE: activity type change (reason)

Final Result: Yes / No
[/THOUGHT]

[JSON]
[refined schedule as JSON array]
[/JSON]

USER MESSAGE FORMAT:
PERSON PROFILE:
{user_profile as JSON}

INITIAL SCHEDULE:
{initial_schedule as JSON}

TRAINING TARGET:
[THOUGHT]
{teacher-generated constraint-checking trace
     and edit operations}
[/THOUGHT]

[JSON]
{final refined schedule matching ground
    truth}
[/JSON]
```

A.3.5 Prompt of GRPO Training

```
You are a daily activity schedule generator.
     Your task is to generate a realistic
    daily schedule for a person based on
    their profile.

OUTPUT FORMAT:
[THOUGHT]
Brief reasoning about the person's schedule
    patterns.
[/THOUGHT]

[JSON]
[
  {"activity": "home", "start_time": "00:00",
     "end_time": "07:00"}
]
[/JSON]

The schedule must start at 00:00 and end at
    24:00, covering the full day without
    gaps or overlaps.

USER MESSAGE FORMAT:
Generate a daily schedule for this person:
{user_profile as JSON}
```

A.4 Detailed Parameter Settings

To ensure the reproducibility of our experiments, we provide the complete hyperparameter configurations for both the Supervised Fine-Tuning (SFT) and Group Relative Policy Optimization (GRPO) stages in Table 4.

All models were trained using the bfloat16 precision format to optimize memory usage while maintaining numerical stability. The AdamW optimizer

Hyperparameter	SFT Stage
Base Model	Qwen3-8B
LoRA Rank (r)	32
LoRA Alpha	64
LoRA Dropout	0.05
Training Epochs	3 epochs
Peak Learning Rate	5e-5
Learning Rate Scheduler	Cosine
Warmup	50 steps
Per-device Batch Size	2
Gradient Accumulation Steps	8
Max Sequence Length	4096

Table 4: Hyperparameter settings for the SFT stage.

Hyperparameter	GRPO Stage
Base Model	Qwen3-8B
LoRA Rank (r)	32
LoRA Alpha	64
LoRA Dropout	0.05
Training Steps	1000 steps
Peak Learning Rate	1e-5
Learning Rate Scheduler	Cosine
Warmup	0.1 ratio
Gradient Accumulation Steps	4
Max Sequence Length	4096
Roll-out Samples per Prompt	8
Sampling Temperature	0.7
Min_p	0.1

Table 5: Hyperparameter settings for the GRPO stage.

was utilized across all training phases, with the beta1 and beta2 parameters set to 0.9 and 0.99, respectively, alongside a weight decay coefficient of 0.1. To prevent gradient explosion, a gradient clipping threshold of 1.0 was universally applied across all parameter updates. As stated in the main text, both stages utilized the exact same Low-Rank Adaptation configuration to efficiently update the attention and feed-forward modules. The specific trainable adapters were applied to the q_proj, k_proj, v_proj, o_proj, gate_proj, up_proj, and down_proj layers. The dataset was split into training, validation, and testing sets with a ratio of 8:1:1, and all reported results were averaged over three independent runs.

A.5 Evaluation Metrics Formulation

Before computing the metrics, each continuous daily schedule is discretized into a fixed-length sequence of 96 time slots, representing 15-minute intervals. The assigned activity for each slot is the one with the longest temporal overlap. Let $G \in \mathbb{Z}^{N\times 96}$ and $T \in \mathbb{Z}^{N\times 96}$ denote the generated and ground truth slot matrices for N users, respectively. We group the 12 metrics into three mathematical categories.

Sequence Similarity Metrics. These metrics evaluate how closely the generated activity sequence matches the ground-truth sequence at the token and structural levels.

Accuracy (Acc) calculates the slot-level exact match rate between the generated and ground truth sequences:

$$\text{Acc} = \frac{1}{N \times 96} \sum_{i=1}^{N} \sum_{t=1}^{96} \mathbf{1}[G_{i,t} = T_{i,t}] \quad (4)$$

Macro F1-Score computes the balanced accuracy over $C = 11$ activity classes:

$$\text{F1}_{\text{macro}} = \frac{1}{K} \sum_{k=1}^{K} \frac{2 \cdot \text{Prec}_k \cdot \text{Recall}_k}{\text{Prec}_k + \text{Recall}_k} \quad (5)$$

Edit Distance (edit_dist) calculates the minimum edit cost, normalized by the sequence length:

$$\text{edit_dist} = \frac{1}{N} \sum_{i=1}^{N} \frac{\text{Levenshtein}(G_i, T_i)}{96} \quad (6)$$

BLEU Score applies n-gram sequence overlap precision up to 4-gram:

$$\text{BLEU} = \frac{1}{N} \sum_{i=1}^{N} \text{sentence_BLEU}(T_i, G_i) \quad (7)$$

Temporal Alignment Metrics. These metrics measure whether the generated schedules preserve realistic temporal patterns, including activity durations and starting times.

These metrics utilize the Jensen-Shannon Divergence (JSD) to evaluate duration and start-time distributions.

Macro_int evaluates the marginal distribution of continuous activity durations (run lengths), denoted as frequency vector m:

$$\text{macro_int} = \text{JSD}\left(\frac{m^{\text{gt}}}{\|m^{\text{gt}}\|_1} \middle\| \frac{m^{\text{gen}}}{\|m^{\text{gen}}\|_1} \right) \quad (8)$$

Micro_int computes the divergence on the flattened joint distribution matrix M of activity types and durations:

$$\text{micro_int} = \text{JSD}\left(\frac{M^{\text{gt}}_{\text{flat}}}{\|M^{\text{gt}}\|_1} \middle\| \frac{M^{\text{gen}}_{\text{flat}}}{\|M^{\text{gen}}\|_1} \right) \quad (9)$$

Initial Draft	Reasoning Cues	Edited Output	Ground Truth
[00:00–07:45] Home		[00:00–07:45] Home	[00:00–07:45] Home
[07:45–16:45] Work	occupation → Evening Work	[07:45–16:30] Work	[07:45–16:30] Work
[16:45–18:00] Home	relationship → Shopping	[16:30–17:20] Shopping	[16:30–17:30] Shopping
[17:45–19:00] Work	constraint → Fix overlap	[17:20–19:50] Work	[17:30–17:45] Service
[19:00–19:35] Shopping	short commute → no extra Home stay	[19:50–24:00] Home	[17:45–20:30] Work
[19:35–24:00] Home			[20:30–24:00] Home
Status: invalid	**Signals:** profile + constraint check	**Status:** valid	**Status:** observed

Table 6: Case study of profile-grounded trajectory editing.

Macro_hour measures the divergence of the marginal distribution of activity onset times (start slots), denoted as vector h:

$$\text{macro_hour} = \text{JSD}\left(\frac{h^{\text{gt}}}{\|h^{\text{gt}}\|_1} \middle\| \frac{h^{\text{gen}}}{\|h^{\text{gen}}\|_1}\right) \quad (10)$$

Micro_hour calculates the divergence on the joint distribution matrix H of activity types and their onset times:

$$\text{micro_hour} = \text{JSD}\left(\frac{H^{\text{gt}}_{\text{flat}}}{\|H^{\text{gt}}\|_1} \middle\| \frac{H^{\text{gen}}_{\text{flat}}}{\|H^{\text{gen}}\|_1}\right) \quad (11)$$

Distributional Consistency Metrics. These metrics assess whether the generated population reproduces the global statistical properties and behavioral diversity of the real-world data.

Data_jsd evaluates population-level trajectory diversity by treating each full 96-slot sequence as a single token with probability p:

$$\text{data_jsd} = \text{JSD}(p^{\text{gt}} \| p^{\text{gen}}) \quad (12)$$

Act_type calculates the divergence of the total time budget allocated to each activity category p_k:

$$\text{act_type} = \text{JSD}(p_k^{\text{gt}} \| p_k^{\text{gen}}) \quad (13)$$

Uni_act_type computes the divergence based on the frequency of distinct activity episodes per category q_k:

$$\text{uni_act_type} = \text{JSD}(q_k^{\text{gt}} \| q_k^{\text{gen}}) \quad (14)$$

Traj_len measures the divergence in the distribution of the total number of distinct activity episodes per user, denoted as count vector r:

$$\text{traj_len} = \text{JSD}\left(\frac{r^{\text{gt}}}{\|r^{\text{gt}}\|_1} \middle\| \frac{r^{\text{gen}}}{\|r^{\text{gen}}\|_1}\right) \quad (15)$$

A.6 Case Study: Profile-Grounded Trajectory Editing

We present a case study based on a real user from the NHTS California subset. The profile indicates that the user isrelationship a female, a mother, employed full-time in a professional, managerial, or technical occupation, does not work from home, drives on the travel day, and has a short commute distance of 4.15 miles. Her primary activity is working.

Reasoning trace. Reasoning process is as follows:

[THOUGHT]
Constraint Checking (5 types):

1. **Physical Constraints (Hard):** - Detected overlap: Home 16:45–18:00 and Work 17:45–19:00
Result: FAIL

2. **Logical Constraints (Hard):** - Starts at Home? Yes - Ends at Home? Yes - Full-day coverage? Yes
Result: PASS

3. **Common Sense Constraints (Soft):** - relationship → Shopping - occupation → Evening Work
Result: informative cues

4. **Temporal Constraints (Soft):** - Shopping is more plausible after daytime Work than after evening Work
Result: needs adjustment

5. **Coherence Constraints (Soft):** - short commute → avoid extra Home stay before returning to Work
Result: needs adjustment

Editing decision: - shift the first Work block earlier, - move Shopping to the in-

terval after daytime Work, - delay the second Work block until after Shopping, - remove the redundant late Shopping segment.

Output: - Home 00:00–07:45 - Work 07:45–16:30 - Shopping 16:30–17:20 - Work 17:20–19:50 - Home 19:50–24:00

Result: valid schedule with improved profile consistency and closer structural alignment to the observed trajectory. [/THOUGHT]

This case study shows the role of the Editor Agent: improving a profile-informed draft by fixing hard errors and refining activity order with profile-conditioned commonsense. The result is not an exact match to the observed day, but a more realistic and coherent mobility trajectory.