MOTIVEMOB: MOTIVATION AS SEMANTIC ACTION FOR CLOSED-LOOP HUMAN MOBILITY GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Human mobility generation, as an important task in urban system research, synthesizes trajectory data that can be used for urban planning, transportation management, etc. Human mobility can be characterized as a "why–where–when" decision process: people form an intention to move and then determine where and when the corresponding activity will take place. Trajectory generation under user-level and temporal distribution shifts may benefit from explicitly modeling this decision structure. However, many existing human mobility generation methods either represent behavioral intent at a coarse granularity, such as a daily plan or a trajectory-level description, or directly predict future locations without explicitly reasoning about a possible motivation for each movement step. We introduce **MotiveMob**, a motivation-driven autoregressive framework for human mobility generation that first forms a hypothesis about why the next movement may occur and then jointly generates where and when it may occur. At each step, a motivation predictor conditions on the current mobility state, a long-term behavioral report, and the mobility history to infer a plausible motivation or determine whether the trajectory should terminate. Given the hypothesized motivation, a state predictor grounds it in a candidate next location and arrival time. Before being fed back as the current mobility state for the next "why–where–when" decision, the candidate is subject to speed-feasibility and repetition checks. We evaluate MotiveMob under distribution shifts involving unseen users and unseen temporal periods, including seasonal changes and the substantial behavioral disruption caused by the COVID-19 pandemic. Experiments show that MotiveMob consistently achieves better distributional fidelity than competitive pretraining-based and prompting-based methods under user-level and temporal distribution shifts, demonstrating robust generalization to out-of-distribution mobility patterns.

1 INTRODUCTION

Human mobility trajectory generation aims to synthesize realistic sequences of visits, including the locations individuals visit and the corresponding timestamps (Luca et al., 2021). High-quality synthetic trajectories support a wide range of applications, including urban planning, transportation management, and epidemic modeling (Zhu et al., 2023a; Lopez et al., 2018; Chang et al., 2021). Recent advances in large language models (LLMs) have inspired new approaches to this task. Some methods fine-tune LLMs for next-point-of-interest (POI) prediction, while others prompt LLMs to generate daily mobility plans from semantic descriptions and historical observations (Li et al., 2024; Wang et al., 2025; 2024; 2026).

From a behavioral perspective, human mobility can be viewed as a motivation-first decision process: people form an intention to move and then determine both where to go and when to arrive. This factorization is particularly important for human mobility generation because similar mobility states may lead to different next movements depending on the latent motivation. However, many existing methods either predict future locations directly from observed trajectories or represent intent only at the coarse granularity of a daily plan, without explicitly modeling the step-level motivation associated with each transition. As a result, they may conflate behaviorally distinct transitions and struggle to maintain consistent activity and temporal patterns over extended generation horizons. This limitation becomes particularly consequential under user-level and temporal distribution shifts, as mobility patterns vary across individuals and evolve with seasonal changes and disruptions such as the COVID-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

19 pandemic. Generalization under these shifts may benefit from representations of users' long-term behavioral patterns and recent activities, together with a transition model that maps hypothesized motivations to plausible next states. Although prior work suggests that LLMs can encode latent environment dynamics and support implicit world modeling (Abdou et al., 2021; Li et al., 2022; Zhang et al., 2025), mobility data record only realized transitions and provide no direct evidence of what would have occurred under alternative motivations, making such dynamics difficult to learn from observed trajectories alone.

Seeing this limitation in existing studies, we introduce **MotiveMob**, an LLM-based autoregressive framework that generates daily mobility trajectories through step-wise "why–where–when" decisions. At each step, a motivation predictor P_M conditions on the current mobility state, a long-term behavioral report $\mathcal{R}$, and a recent mobility history $\mathcal{H}$ to predict the motivation for the next movement or determine whether the trajectory should terminate. A state predictor P_S then conditions on the same user context and treats the predicted motivation as a semantic action, grounding it in a feasible next location and arrival time. To train P_S beyond observed transitions, we construct pseudo alternative transitions by pairing observed states with alternative motivation labels, generating candidate next states with a lightweight predictor, and using an LLM-based judge to select the most plausible candidate. The observed and alternative transitions are then used to train P_S as a motivation-conditioned transition model, which serves as the dynamics component of an implicit world model (IWM). During inference, P_M and P_S operate alternately. Speed-infeasible candidates are regenerated, whereas repeated states terminate the rollout. Each accepted state then becomes the current mobility state for the next decision.

We evaluate MotiveMob under both user-level and temporal distribution shifts, covering seen and unseen users across an observed period, a regular unseen period, and a COVID-19 period characterized by substantial changes in mobility behavior. Across these settings, MotiveMob consistently achieves better distributional fidelity than competitive pretraining-based and prompting-based methods. The advantage is particularly evident in time–activity distribution fidelity: averaged across the three temporal periods, MotiveMob reduces the corresponding divergence by 10.7% for seen users and 18.3% for unseen users relative to the best-performing baseline in each setting. Compared with the variant without P_M, the corresponding reductions are 21.6% and 29.1%, respectively, supporting the benefit of reasoning about why a movement may occur before generating where and when it occurs. In addition, the evaluation analyzes the roles and metric-specific trade-offs of implicit world modeling, user-specific context in trajectory generation, and inference-time rollout validation.

Our contributions are summarized as follows:

- We formulate daily mobility trajectory generation as a step-level, motivation-first "why–where–when" process and introduce **MotiveMob**, which separates potential motivation prediction from the joint generation of the next location and arrival time.
- We develop an offline alternative transition augmentation procedure for training the state predictor as an IWM. A lightweight candidate generator proposes outcomes for hypothetical motivations, while an LLM-based judge filters plausible transitions, enabling the model to learn beyond observed state–motivation pairs without requiring online environment interaction.
- We conduct a systematic evaluation under user-level and temporal distribution shifts, covering unseen users, regular seasonal variation, and pandemic-induced behavioral disruption. MotiveMob consistently achieves better distributional fidelity than competitive pretraining-based and prompting-based baselines.

2 Related Work

2.1 Human Mobility Modeling

Human mobility modeling encompasses next-location prediction and multi-step trajectory generation (Luca et al., 2021; Kong et al., 2023). Next-location prediction infers a user's next destination from historical visits and contextual information. Early approaches used probabilistic transition models (Gambs et al., 2012; Lu et al., 2013), followed by recurrent and attention-based architectures that capture long-term dependencies and individual preferences (Feng et al., 2018; Luo et al., 2021).

Recent LLM-based methods employ time-aware prompting or zero-shot candidate ranking (Wang et al., 2023; Feng et al., 2024), while others adapt LLMs through task-specific fine-tuning (Li et al., 2024; Wang et al., 2025); Mobility-LLM further learns intention- and preference-aware representations for predictive check-in tasks (Gong et al., 2024). These methods primarily target one-step or related predictive objectives. Trajectory generation instead synthesizes complete location–time sequences and aims to reproduce population-level spatiotemporal regularities. Existing generators employ variational autoencoders (Huang et al., 2019), adversarial imitation learning (Zhang et al., 2020; Choi et al., 2021), and diffusion models (Zhu et al., 2023b; Chu et al., 2024; Long et al., 2026). More recent LLM-based methods formulate trajectories as conditional sequences or agent-generated plans (Ohara et al., 2024; Zhang et al., 2024; Li et al., 2025; Wang et al., 2026; Li et al., 2026), as summarized in a recent survey (Gao & Wu, 2026). Particularly relevant activity-centric approaches generate activity–time chains before assigning spatial locations (Liao et al., 2026), while retrieval-augmented methods synthesize household-coordinated activity chains from aggregate statistics and sociodemographic profiles (Liu et al., 2025). LLMob also infers a daily-level motivation to guide whole-day trajectory generation (Wang et al., 2024). However, existing activity-centric generators typically introduce semantic conditioning at the level of whole-day motivations or activity chains, rather than explicitly modeling the motivation associated with each individual transition. Moreover, how such models generalize to unseen users and future or disrupted periods also remains underexplored.

2.2 World Modeling with LLMs

In reinforcement learning (RL), it is often desirable for an agent to have knowledge of the environment dynamics, enabling it to model state transitions and predict future states and rewards, as in model-based RL (Luo et al., 2024). Traditionally, such knowledge is acquired by learning a world model from observed state transitions, ranging from simple lookup-based models (Sutton, 1991) to deep generative models (Ha & Schmidhuber, 2018). With the emergence of LLMs, agents have begun to employ LLMs as world models to support planning and decision making (Gu et al., 2024; Feng et al., 2025). However, most prior approaches treat the world model as an explicit and separate simulator, echoing classical control pipelines that rely on planning over imagined rollouts. In contrast, Zhang et al. (2025) propose an early experience paradigm in which the policy is trained to predict its own future states from an IWM, thereby internalizing environment dynamics without constructing a standalone simulator. Learning action-conditioned dynamics, however, requires transition data that cover the consequences of different actions under comparable states. Human mobility records contain only real transitions and do not reveal what might have occurred under alternative motivations. This lack of counterfactual observations, together with the absence of an interactive mobility environment, presents a key challenge for learning IWMs in trajectory generation.

3 Preliminaries

3.1 Daily Trajectory Generation

For a user u and a target date d, we represent a daily trajectory as an ordered sequence of location–time pairs,

$$\tau_{u,d} = \left\langle (l_0, t_0), (l_1, t_1), \ldots, (l_{n_{u,d}-1}, t_{n_{u,d}-1}) \right\rangle, \tag{1}$$

where l_i denotes the POI identifier of the i-th check-in and t_i denotes its arrival timestamp, (l_i, t_i) denotes the user's i-th check-in, and $n_{u,d}$ denotes the number of check-ins in the trajectory. Given the user's long-term behavioral report $\mathcal{R}_u$, recent mobility history $\mathcal{H}_{u,d}$, and target date d, our objective is to generate $\tau_{u,d}$, describing the sequence of locations visited by the user and their corresponding visit times during the target mobility day. The recent mobility history $H_{u,d}$ always precedes the target date, whereas R_u is constructed from a query-disjoint support set from the seen period. The construction and textual representations of $\mathcal{R}_u$ and $\mathcal{H}_{u,d}$ are detailed in Appendix A.

3.2 Mobility Dynamics as a World Model

From a world model perspective, each check-in is represented as a state $s_i = (l_i, t_i) \in \mathcal{S}$, and each transition from s_i to s_{i+1} is conditioned on a semantic action $m_i \in \mathcal{M}$. We interpret m_i as a transition-level latent variable representing a plausible motivation for the user's next movement,

such as work, shopping, or healthcare. Because psychological intent is not directly observed in mobility traces, we weakly supervise m_i using labels derived from destination categories, which serve as operational proxies rather than ground-truth annotations of human intent; the full taxonomy is provided in Appendix B.For a user u and a target date d, the corresponding mobility dynamics can be expressed as

$$s_{i+1} \sim \mathcal{T}\left(\cdot \mid s_i, m_i, \mathcal{R}_u, \mathcal{H}_{u,d}, d\right). \tag{2}$$

which characterizes the distribution of the next location and arrival time resulting from a motivation under the user's behavioral context.

4 METHODOLOGY

As illustrated in Figure 1, we introduce **MotiveMob**, an LLM-based framework that generates daily mobility trajectories by repeatedly making "why–where–when" decisions. Unlike existing approaches that reason globally about an entire trajectory or daily plan (Wang et al., 2024; 2026), MotiveMob decomposes mobility generation into a sequence of motivation-conditioned check-in transitions. At each step, the motivation predictor P_M conditions on the current mobility state, the user's long-term behavioral report $\mathcal{R}_u$, and recent mobility history $\mathcal{H}_{u,d}$ to infer the latent motivation for the next move or determine whether the trajectory should terminate. The state predictor P_S then treats the inferred motivation as a semantic action and grounds it in a candidate next location and arrival time, after which the candidate transition undergoes inference-time validation. The two predictors are trained separately: P_S learns from observed and alternative transitions, whereas P_M is fine-tuned with step-level motivation labels. During inference, each accepted state becomes the current mobility state for the next decision.

4.1 STATE PREDICTOR

4.1.1 ALTERNATIVE TRANSITION CONSTRUCTION

Because human mobility records contain only real transitions and do not reveal the outcomes of alternative motivations, we augment the observed data with model-synthesized alternative transitions constructed through a two-stage pipeline. Let $\mathcal{D}_{\text{obs}} = \{(s_i, m_i, s_{i+1})\}_{i=1}^{N}$ denote the set of observed transitions. For each observed transition in $\mathcal{D}_{\text{obs}}$, we sample three alternative motivations $\{m_i^j\}_{j=1}^{3} \subset \mathcal{M} \setminus \{m_i\}$, each representing a motivation that the user could have adopted from the same state s_i. In the first stage, a lightweight Transformer-based candidate generator (Vaswani et al., 2017) conditions on the trajectory history, the current state s_i, and an alternative motivation m_i^j to produce a set of candidate next states $\mathcal{C}_i^j$. In the second stage, an LLM-based judge evaluates these candidates using the user's historical information $\mathcal{H}_i$, the specified motivation m_i^j, and their spatial and temporal distances from the current state, denoted by $\boldsymbol{\delta}_i^j$. It then selects the candidate most consistent with the user's behavioral profile, mobility context, and the alternative motivation:

$$\tilde{s}_{i+1}^j = \text{Judge}_{\text{LLM}}\left(\mathcal{C}_i^j, \mathcal{H}_i, m_i^j, \boldsymbol{\delta}_i^j\right). \tag{3}$$

The selected states form a pool of pseudo-labeled alternative transitions, denoted by $\mathcal{D}_{\text{alt}}$. To balance observed and alternative supervision during training, we uniformly retain a fraction ρ of this pool, yielding $\mathcal{D}_{\text{alt}}^{(\rho)} \subseteq \mathcal{D}_{\text{alt}}$. The augmented training set is then defined as $\mathcal{D}_{\text{aug}} = \mathcal{D}_{\text{obs}} \cup \mathcal{D}_{\text{alt}}^{(\rho)}$.

4.1.2 IMPLICIT WORLD MODELING

Using the augmented transition set $\mathcal{D}_{\text{aug}}$, we train the state predictor P_S as an IWM of motivation-conditioned mobility dynamics. Let $c_i = (s_i, \mathcal{R}_u, \mathcal{H}_{u,d}, d)$ denote the context available at step i, including the current mobility state, the behavioral report of user u, the recent mobility history preceding the target date, and the target date d. Given a context c_i and a motivation m treated as a semantic action, P_S models the resulting next-state distribution as $p_{\theta_S}(s' \mid c_i, m)$. For training, we associate each transition with its corresponding context and represent each instance in $\mathcal{D}_{\text{aug}}$ as (c_i, m, s'). We train P_S by minimizing the average negative log-likelihood:

$$\mathcal{L}_S = -\frac{1}{|\mathcal{D}_{\text{aug}}|} \sum_{(c_i, m, s') \in \mathcal{D}_{\text{aug}}} \log p_{\theta_S}(s' \mid c_i, m). \tag{4}$$

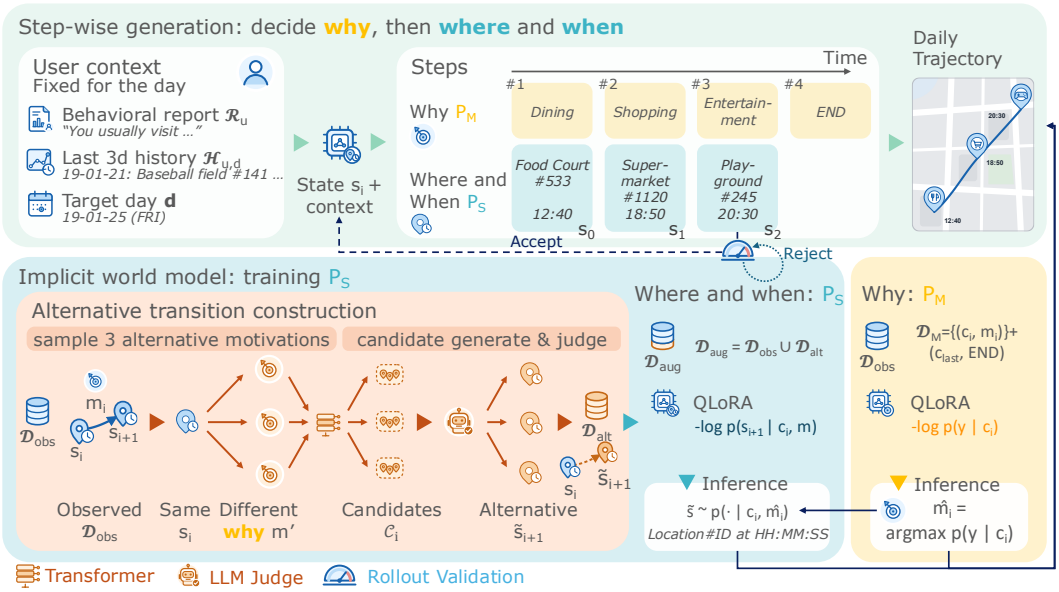

Figure 1: Overview of the MotiveMob framework. Given the user context, MotiveMob generates a daily trajectory step by step, first predicting *why* (P_M) and then *where and when* (P_S). The implicit world model P_S is trained using observed and alternative transitions. The marker denotes *rollout validation*, which performs speed-feasibility checking and repetition-based termination, as described in Section 4.3.

This yields an offline, reward-free world modeling objective that combines factual grounding with broader coverage of motivation-conditioned mobility dynamics.

4.2 Motivation Predictor

The state predictor P_S grounds a given motivation in a next mobility state but does not determine which motivation should drive the next move. We therefore train a dedicated motivation predictor P_M to infer a step-level potential motivation, or predict trajectory termination, from the context $c_i = (s_i, \mathcal{R}_u, \mathcal{H}_i)$. To prevent future information leakage, c_i contains only information available up to step i and excludes the next state s_{i+1} and all subsequent states. Besides, for each observed trajectory, we append a terminal training instance whose label is `END`. We construct the motivation prediction dataset as $\mathcal{D}_M = \{(c_i, y_i)\}_{i=1}^{N_M}$, where $y_i = m_i$ for an observed transition and $y_i = \texttt{END}$ when the trajectory should terminate. We fine-tune P_M by minimizing the average negative log-likelihood:

$$\mathcal{L}_M = -\frac{1}{|\mathcal{D}_M|} \sum_{(c_i, y_i) \in \mathcal{D}_M} \log p_{\theta_M}(y_i \mid c_i). \tag{5}$$

This supervised objective enables P_M to infer the behavioral intent underlying the next move or determine that no further transition should be generated. During generation, a predicted motivation is passed to P_S for state grounding, whereas `END` terminates the trajectory.

4.3 Motivation-Guided Trajectory Generation

At inference time, P_M and P_S operate sequentially to generate a daily trajectory through step-level"why–where–when" decisions. Generation begins with a `START` token in the textual context. Given this context, P_M predicts the first motivation and P_S generates the initial physical state s_0. Speed validation is applied from the subsequent transition onward. At each step i, P_M greedily predicts $\hat{m}_i = \arg\max_{y \in \mathcal{M} \cup \{\texttt{END}\}} p_{\theta_M}(y \mid c_i)$. If $\hat{m}_i = \texttt{END}$, generation terminates. Otherwise, P_S conditions on c_i and $\hat{m}_i$ to sample a provisional next state $\tilde{s}_{i+1} \sim p_{\theta_S}(\cdot \mid c_i, \hat{m}_i)$, where $\tilde{s}_{i+1}$ specifies the next location and arrival time. Once accepted, $\tilde{s}_{i+1}$ becomes the current state in c_{i+1}, and the same procedure is repeated for the next transition.

Inference-Time Rollout Validation. Before a provisional state is added to the trajectory, it is validated for temporal and travel feasibility. Specifically, its arrival time must not precede the current time, and its implied travel speed must not exceed $v_{\max} = 50$ km/h. The selection of this threshold is discussed in Appendix C. An invalid proposal is discarded and resampled from P_S. If no proposal satisfies the constraints within a fixed number of attempts, the last proposal is re-grounded to a feasible location of the same activity category using the user's observed mobility information (Li et al., 2026). If no such feasible location is available, the final model proposal is retained unchanged. We additionally apply repetition-aware termination once the generated trajectory reaches a minimum length: the rollout terminates if the candidate-augmented location sequence contains a short consecutively repeating pattern. Generation also terminates when the prescribed maximum trajectory length is reached.

Through this iterative process, P_M determines why the next move occurs, while P_S grounds the predicted motivation in where and when it occurs. Additional implementation details are provided in the appendices. Appendix D provides implementation details for alternative transition construction, predictor training, and autoregressive trajectory inference; Appendix E provides the prompts used for training and inference; and Appendix F presents pseudocode for inference-time trajectory generation.

5 Experiments

5.1 Experimental Setup

Dataset. All experiments are conducted on a human mobility dataset collected in Tokyo through the Twitter and Foursquare APIs. The original dataset consists of individual check-ins with location and timestamp information. We transform consecutive check-ins from the same user into transitions, each comprising the origin and destination states and one of 11 behavioral motivation labels derived from the category of the destination venue. We additionally append an `END` label after the final check-in to mark trajectory termination. For the seen-user, seen-time setting, we randomly hold out 20% of each user's trajectories for evaluation and use the remaining 80% for training.

Distribution-Shift Settings. We evaluate generalization along both user and temporal dimensions. The training set contains trajectories from 25 seen users collected between January and October 2019. For user-level generalization, we evaluate exclusively on an additional 50 users whose trajectories are not included in training. For temporal generalization, we consider two unseen periods: November 2019 to February 2020, representing a regular seasonal shift, and April to October 2020, representing a pandemic-induced mobility shift following the onset of COVID-19 restrictions. Detailed split construction and leakage prevention are provided in Appendix G.

Baseline Methods. We compare MotiveMob with two categories of baselines: (1) pretraining-based models, including LSTM (Hochreiter & Schmidhuber, 1997), MHSA (Vaswani et al., 2017), DeepMove (Feng et al., 2018), GetNext (Yang et al., 2022), TrajGAIL (Choi et al., 2021), Diff-Traj (Zhu et al., 2023b), GNPR-SID (Wang et al., 2025), and LLM4POI (Li et al., 2024); and (2) prompting-based models, including LLM-Mob (Wang et al., 2023), LLMMove (Feng et al., 2024), LLM-ZS (Beneduce et al., 2025), LLMob (Wang et al., 2024), and ELLMob (Wang et al., 2026).

Evaluation Metrics. Following Yuan et al. (2022) and Wang et al. (2024), we evaluate generated trajectories using four distributional metrics: *Step Distance (SD)*, the geographic distance between consecutive locations; *Step Interval (SI)*, the elapsed time between consecutive check-ins; *Daily Activity Routine Distribution (DARD)*, the joint distribution of discretized time slot t and activity category c; and *Spatiotemporal Visit Distribution (STVD)*, the joint distribution of time and location, represented by $(t, \mathrm{lat}, \mathrm{lng})$. For each metric, we compute the Jensen–Shannon divergence (JSD) between the generated and real distributions, with lower values indicating better performance.

5.2 Results and Analysis

Tables 1a and 1b report the results for seen and unseen users across the three temporal settings. All methods use the same validation and termination rules. MotiveMob achieves lower JSD than all baselines across both user groups and all three temporal settings. Its advantage persists for users excluded from training, indicating that its motivation-conditioned modeling transfers beyond observed user identities. Although temporal shifts generally increase divergence, with the COVID-19

Table 1: Evaluation results for seen and unseen users across the three temporal settings. Panels (a) and (b) report results for seen and unseen users, respectively. In each panel, the upper and lower baseline sections correspond to pretraining-based and prompting-based methods, respectively. All values are JSD; lower is better. The best result in each column of each panel is shown in bold, and the runner-up is underlined. Speed and Rep. denote speed validation and repetition-aware termination, respectively. Best results are in bold, and runner-up results are underlined.

(a) Seen users

Method	Seen period (908 trajectories)				Regular unseen period (690 trajectories)				COVID-19 unseen period (392 trajectories)			
	SD	SI	DARD	STVD	SD	SI	DARD	STVD	SD	SI	DARD	STVD
LSTM	0.0915	0.1042	0.2510	0.4079	0.1014	0.1530	0.3046	0.5159	0.0823	0.1600	0.3713	0.6267
MHSA	0.0319	0.0507	0.1898	0.4049	0.0393	0.0590	0.2157	0.4753	0.0424	0.0621	0.2638	0.6201
DeepMove	0.0830	0.0990	0.2334	0.4991	0.0593	0.1305	0.3592	0.5331	0.0911	0.1372	0.4313	0.6422
GetNext	0.0440	0.2414	0.2126	0.4917	0.0559	0.2387	0.2114	0.5659	0.0773	0.2046	0.1991	0.6613
TrajGAIL	0.1079	0.1382	0.1140	0.5910	0.1183	0.1295	0.1360	0.6367	0.1311	0.1363	0.2136	0.6570
DiffTraj	0.2000	0.1204	0.4777	0.6931	0.2193	0.1125	0.5157	0.6898	0.2595	0.1052	0.5716	0.6903
GNPR-SID	0.0171	0.0163	0.1006	0.4167	0.0145	0.0207	0.1103	0.4799	0.0491	0.0310	0.1445	0.5723
LLM4POI	0.0098	0.0236	0.1090	0.3788	0.0138	0.0340	0.1329	0.4622	0.0452	0.0217	0.1405	0.5750
LLMMove	0.0839	0.1164	0.1906	0.5637	0.0830	0.1091	0.1901	0.5607	0.1137	0.0937	0.2572	0.6041
LLM-ZS	0.0306	0.1034	0.2369	0.5936	0.0441	0.1016	0.2505	0.6014	0.0732	0.0813	0.2817	0.6295
LLM-Mob	0.0951	0.1171	0.3242	0.5639	0.1166	0.1135	0.3336	0.5856	0.1328	0.0941	0.3501	0.6201
LLMob	0.0919	0.2496	0.1819	0.5461	0.0820	0.2422	0.2353	0.5972	0.0982	0.2610	0.2013	0.6229
ELLMob	0.0501	0.1053	0.2791	0.4598	0.0476	0.1085	0.2863	0.4937	0.0583	0.1245	0.3309	0.5836
MotiveMob	**0.0062**	**0.0100**	0.0765	**0.3658**	**0.0102**	0.0161	0.0995	**0.4301**	0.0312	**0.0147**	0.1377	**0.5487**
MotiveMob w/o P_M	0.0106	0.0168	0.1092	0.3757	0.0179	0.0174	0.1300	0.4541	**0.0253**	0.0233	0.1610	0.5530
MotiveMob w/o $\mathcal{R}_u$	0.0168	0.0219	0.0912	0.5142	0.0185	0.0259	0.1030	0.5583	0.0525	0.0193	**0.1266**	0.5836
MotiveMob w/o $\mathcal{H}_{u,d}$	0.0120	0.0291	0.1422	0.4291	0.0142	0.0343	0.1760	0.4887	0.0805	0.0299	0.1875	0.5945
MotiveMob w/o Speed	0.0276	0.0109	**0.0735**	0.3762	0.0250	**0.0152**	**0.0951**	0.4351	0.0878	0.0241	0.1300	0.5548
MotiveMob w/o Rep.	0.0074	0.0161	0.0891	0.3749	0.0125	0.0235	0.1114	0.4423	0.0281	0.0245	0.1461	0.5578

(b) Unseen users

Method	Seen period (1345 trajectories)				Regular unseen period (795 trajectories)				COVID-19 unseen period (541 trajectories)			
	SD	SI	DARD	STVD	SD	SI	DARD	STVD	SD	SI	DARD	STVD
LSTM	0.3607	0.3354	0.5257	0.6193	0.3610	0.3556	0.5601	0.6515	0.3456	0.3554	0.5648	0.6786
MHSA	0.1779	0.0733	0.3335	0.5796	0.1510	0.0895	0.3538	0.6122	0.1557	0.0918	0.3639	0.6716
DeepMove	0.2247	0.1621	0.2977	0.6660	0.2517	0.1896	0.4468	0.6883	0.2393	0.1889	0.4370	0.6897
GetNext	0.0715	0.1930	0.2303	0.6223	0.0627	0.1818	0.2419	0.6561	0.0760	0.1931	0.2246	0.6703
TrajGAIL	0.0999	0.0801	0.1051	0.5941	0.1237	0.0808	0.1273	0.6222	0.1079	0.0866	0.1540	0.6638
DiffTraj	0.2176	0.0671	0.5589	0.6915	0.2082	0.0708	0.5681	0.6914	0.2327	0.0773	0.5459	0.6921
GNPR-SID	0.0270	0.0488	0.1428	0.4762	0.0263	0.0508	0.1514	0.5356	0.0388	0.0543	0.1706	0.6269
LLM4POI	0.0469	0.0329	0.1152	0.5391	0.0516	0.0339	0.1452	0.5873	0.0553	0.0369	0.1502	0.6386
LLMMove	0.0914	0.1376	0.1523	0.5303	0.0787	0.1330	0.1612	0.5921	0.1044	0.1557	0.1802	0.6253
LLM-ZS	0.0313	0.1049	0.2171	0.5713	0.0373	0.1221	0.2394	0.6212	0.0442	0.1122	0.2449	0.6473
LLM-Mob	0.0948	0.0991	0.3200	0.5314	0.1007	0.1015	0.3284	0.5846	0.1196	0.1091	0.3540	0.6271
LLMob	0.0555	0.1943	0.1647	0.5449	0.0677	0.1849	0.1860	0.5851	0.0445	0.1830	0.2048	0.6422
ELLMob	0.0437	0.0751	0.2159	0.4830	0.0401	0.0750	0.2341	0.5428	0.0364	0.0861	0.2875	0.6336
MotiveMob	0.0213	**0.0204**	**0.0868**	**0.4498**	0.0256	**0.0273**	0.1103	**0.5137**	0.0273	**0.0303**	0.1153	**0.6060**
MotiveMob w/o P_M	0.0201	0.0311	0.1219	0.4627	0.0200	0.0354	0.1471	0.5153	0.0225	0.0454	0.1714	0.6253
MotiveMob w/o $\mathcal{R}_u$	**0.0142**	0.0385	0.1023	0.5082	**0.0146**	0.0357	0.1243	0.5700	**0.0154**	0.0445	0.1485	0.6149
MotiveMob w/o $\mathcal{H}_{u,d}$	0.0289	0.0258	0.1193	0.4777	0.0351	0.0309	0.1443	0.5394	0.0306	0.0495	0.1385	0.6272
MotiveMob w/o Speed	0.0633	0.0287	0.0870	0.4527	0.0698	0.0373	0.1145	0.5230	0.0775	0.0454	0.1164	0.6163
MotiveMob w/o Rep.	0.0181	0.0262	0.0925	0.4535	0.0227	0.0329	**0.1079**	0.5216	0.0210	0.0326	**0.1122**	0.6082

period producing the largest mismatch in most dimensions, MotiveMob maintains its advantage under both regular seasonal and pandemic-induced shifts. In contrast, the baselines exhibit metric-specific strengths but do not perform consistently well across mobility dimensions. To further examine MotiveMob across both urban contexts and backbone choices, Appendix H reports additional results for Osaka, while Appendix I evaluates MotiveMob with different LLM backbones.

5.3 Analysis of Prediction Modules

Step-level Motivation Predictor. The *w/o* P_M rows in Tables 1a and 1b evaluate a direct predictor that generates the next mobility state from the current context without first inferring a step-level motivation. This variant retains a learned termination decision by predicting either the next state or

Table 2: State predictor comparison without rollout validation for seen and unseen users across the three temporal settings. All values are JSD; lower is better. The best result in each column of each panel is shown in bold, and the runner-up is underlined.

State predictor P_S	Seen period				Regular unseen period				COVID-19 unseen period			
	SD	SI	DARD	STVD	SD	SI	DARD	STVD	SD	SI	DARD	STVD
Seen users												
MotiveMob (IWM P_S)	**0.0241**	0.0159	0.0857	**0.3841**	**0.0307**	0.0212	0.1054	0.4558	0.0880	0.0309	0.1632	0.5745
w/ positive-only P_S	0.0328	0.0115	**0.0707**	0.3870	0.0376	0.0185	**0.0943**	**0.4469**	0.0914	0.0390	0.1586	**0.5602**
w/ rule-based P_S	0.0374	**0.0101**	0.0992	0.4286	0.0405	**0.0165**	0.1111	0.4932	**0.0687**	**0.0198**	**0.1576**	0.6230
Unseen users												
MotiveMob (IWM P_S)	0.0460	0.0356	0.0906	**0.4559**	0.0473	0.0422	0.1120	**0.5240**	0.0597	0.0359	0.1209	**0.6109**
w/ positive-only P_S	0.0713	**0.0244**	**0.0878**	0.4875	0.0876	**0.0300**	0.1181	0.5417	0.1045	**0.0220**	**0.1165**	0.6440
w/ rule-based P_S	**0.0388**	0.0333	0.0916	0.5663	**0.0375**	0.0341	**0.1097**	0.5926	**0.0434**	0.0317	0.1399	0.6478

END; the comparison therefore isolates the effect of motivation-mediated state prediction rather than removing trajectory termination.

Removing P_M increases both SI and DARD in all six evaluation settings. The mean relative increases in SI and DARD are 44.9% and 30.1% for seen users and 44.0% and 40.8% for unseen users, respectively. In contrast, the effects of P_M on the spatial metrics are mixed: the mean absolute relative change in STVD is only 2.6%, while P_M reduces SD divergence in two of the three seen-user settings but increases it in all three unseen-user settings. These improvements in SI and DARD are not simply explained by termination behavior or average trajectory length. As detailed in Appendix J, the direct predictor generates 7.52 states per day on average, compared with 8.08 for MotiveMob and 6.82 in the ground truth. Although the direct predictor is closer to the ground-truth mean length, it still performs worse on SI and DARD. These results suggest that step-level motivation primarily helps preserve temporal and activity semantics by narrowing spatially plausible alternatives according to behavioral intent, which helps determine when the next move should occur and how it fits into the user's daily activity sequence. To visualize the effect of motivation predictor, Appendix K presents representative trajectory-level case studies comparing MotiveMob w/ and w/o motivation predictor.

IWM for State Prediction. We compare the IWM-based state predictor, trained on both observed and alternative transitions, with two alternatives while keeping the motivation predictor P_M fixed and inference-time rollout validation disabled. The positive-only predictor is trained solely on observed transitions, whereas the rule-based predictor retrieves observed mobility transitions compatible with each predicted motivation instead of learning a generative transition model.

As shown in Table 2, the IWM-based predictor achieves lower SD than the positive-only predictor in all six settings and lower STVD in four, including all three unseen-user settings, whereas the positive-only predictor achieves lower SI and DARD in five settings. Relative to the rule-based predictor, the IWM-based predictor achieves lower STVD in all six settings and lower DARD in four, whereas the rule-based predictor achieves lower SD in four and lower SI in all six. These results highlight a trade-off between matching individual mobility marginals and preserving user-conditioned spatiotemporal structure. The clearest advantage of IWM training lies in modeling the joint time–location distribution, particularly for unseen users, suggesting that alternative transitions broaden the motivation-conditioned state distribution and improve spatiotemporal generalization beyond observed-transition learning or rule-based retrieval alone. Appendix L further shows that IWM initialization provides no consistent final improvement for the motivation predictor.

5.4 Ablation Studies

User-specific information. Removing the behavioral report $\mathcal{R}_u$ worsens SI and STVD in all six settings, confirming its role as a long-term prior over when and where a user typically performs an activity. Its effect on DARD is weaker for seen users but more pronounced for unseen users. The mixed SD results further suggest that displacement patterns encoded in the report are less reliably calibrated under user-level shifts. Recent mobility history provides more immediate conditioning: removing $\mathcal{H}_{u,d}$ worsens SI in all settings and substantially increases DARD and SD for both user

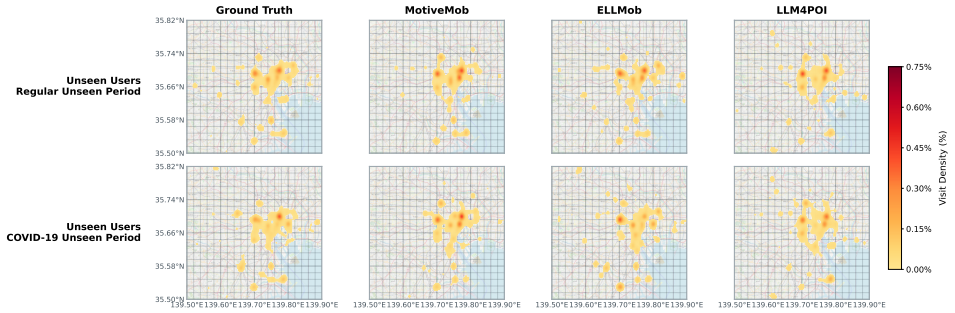

Figure 2: Ground-truth and predicted spatial mobility distributions for unseen users across two unseen periods in Tokyo. Darker red indicates higher visit density.

groups. Its effect on STVD is smaller than that of $\mathcal{R}_u$, indicating that recent history mainly locates the user within the current routine and constrains the next activity, its timing, and the associated displacement.

Inference-time rollout validation. Removing speed validation increases SD and STVD in all six settings, whereas its effects on SI and DARD are smaller and less uniform. Speed validation therefore primarily prevents physically implausible spatial transitions. Repetition-aware termination serves a different function: disabling it consistently worsens SI and STVD but produces smaller and more mixed changes in SD and DARD. Its primary role is therefore to prevent repeated location cycles from extending trajectories to the maximum length, rather than driving MotiveMob's overall distributional gains. Detailed resampling, correction, and termination statistics are reported in Appendix M.

5.5 Spatial Distribution Alignment under User and Temporal Shifts

As shown in Figure 2, all three methods recover the primary activity centers in the ground-truth distribution. Compared with LLM4POI, MotiveMob forms more compact hotspots and exhibits less peripheral dispersion, thereby better preserving the concentrated structure of the ground-truth distribution. Compared with ELLMob, MotiveMob more closely matches the ground-truth allocation of visit mass across activity regions, particularly under the COVID-19 shift. These results suggest that MotiveMob combines compact hotspot structure with more robust spatial alignment under joint user and temporal shifts. Additional spatial-distribution results are provided in Appendix N.

6 Conclusions and Limitations

By combining step-level prediction of motivation proxies, motivation-conditioned state generation, and inference-time rollout validation, our motivation-first framework, **MotiveMob**, achieved the strongest overall distributional fidelity across the evaluated user groups and temporal periods. Spatial analysis further showed that MotiveMob better preserves core activity regions for unseen users under temporal shifts. Additional analyses indicate that step-level motivation prediction primarily improves temporal and activity fidelity, while IWM training with alternative transitions improves joint spatiotemporal generalization, particularly for unseen users. Together, these findings support motivation proxies as useful intermediate representations for mobility trajectory generation.

As for limitations, MotiveMob relies on a predefined motivation taxonomy and pseudo-labeled alternative transitions whose correspondence to unobserved user intentions cannot be directly verified. The current IWM augmentation also exhibits a metric-specific trade-off: improvements in the joint time–location distribution do not consistently coincide with improvements in the step-distance distribution, suggesting that the generated alternative transitions may not simultaneously preserve local movement geometry and global spatiotemporal structure. Moreover, the step-level training objectives do not explicitly optimize trajectory-level coherence or capture the delayed effects of earlier decisions. Future work could explore adaptive motivation representations, improved augmentation strategies that better preserve both local and global mobility patterns, and hierarchical sequential objectives with trajectory-level feedback or reinforcement learning.

AI USE STATEMENT

We used OpenAI Codex to assist with improving the grammar and readability of the manuscript, identifying potentially relevant literature, and editing experimental code. All suggested references were manually verified against their original sources. All AI-assisted code was manually reviewed and tested by the authors. We did not use generative AI tools to formulate the research hypotheses, design the experiments, interpret the results, or develop mathematical claims or proofs.

ETHICS STATEMENT

All data used in our study is fully anonymized. The dataset was collected through Twitter Inc.'s Academic Research Product Track and Foursquare in strict compliance with their privacy policies and terms of service. Our data collection and preprocessing methodology follows the established work Wang et al. (2024). While the data sources are legitimate, we recognize that cross-referencing with original Twitter posts can potentially enable re-identification. To mitigate this risk, we implemented the following anonymization measures: (1) User IDs are replaced with random numbers; (2) Specific venue names are generalized to categories; (3) Exact timestamps are discretized into time intervals; and (4) User-generated comments are provided only in translated form to prevent linguistic fingerprinting. These comprehensive measures ensure that reverse identification is computationally infeasible even with access to the original platforms.

REPRODUCIBILITY STATEMENT

To ensure the reproducibility of our work, we have made our code and data publicly available in the supplementary materials. Our experimental setup, including evaluation metrics and parameter settings, are described in Section 5 and Appendix D. The anonymized code repository is available at `https://anonymous.4open.science/r/MotiveMob-F6E0/`.

REFERENCES

Mostafa Abdou, Artur Kulmizev, Daniel Hershcovich, Stella Frank, Ellie Pavlick, and Anders Søgaard. Can language models encode perceptual structure without grounding? a case study in color. In *Proceedings of the 25th conference on computational natural language learning*, pp. 109–132, 2021.

Ciro Beneduce, Bruno Lepri, and Massimiliano Luca. Large language models are zero-shot next location predictors. *IEEE Access*, 13:77456–77467, 2025.

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

Seongjin Choi, Jiwon Kim, and Hwasoo Yeo. Trajgail: Generating urban vehicle trajectories using generative adversarial imitation learning. *Transportation Research Part C: Emerging Technologies*, 128:103091, 2021.

Chen Chu, Hengcai Zhang, Peixiao Wang, and Feng Lu. Simulating human mobility with a trajectory generation framework based on diffusion model. *International Journal of Geographical Information Science*, 38(5):847–878, 2024.

Jichen Feng, Yifan Zhang, Chenggong Zhang, Yifu Lu, Shilong Liu, and Mengdi Wang. Web world models. *arXiv preprint arXiv:2512.23676*, 2025.

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

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

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

Jie Gao and Yaoxin Wu. LLMs for human mobility: Opportunities, challenges, and future directions, 2026. URL https://arxiv.org/abs/2603.12420.

Letian Gong, Yan Lin, Xinyue Zhang, Yiwen Lu, Xuedi Han, Yichen Liu, Shengnan Guo, Youfang Lin, and Huaiyu Wan. Mobility-LLM: Learning visiting intentions and travel preference from human mobility data with large language models. In *Advances in Neural Information Processing Systems*, volume 37, pp. 36185–36217, 2024. doi: 10.52202/079017-1141. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/3fb6c52aeb11e09053c16eabee74dd7b-Abstract-Conference.html.

Yu Gu, Kai Zhang, Yuting Ning, Boyuan Zheng, Boyu Gou, Tianci Xue, Cheng Chang, Sanjari Srivastava, Yanan Xie, Peng Qi, et al. Is your llm secretly a world model of the internet? model-based planning for web agents. *arXiv preprint arXiv:2411.06559*, 2024.

David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. *Advances in neural information processing systems*, 31, 2018.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8): 1735–1780, 1997.

Dou Huang, Xuan Song, Zipei Fan, Renhe Jiang, Ryosuke Shibasaki, Yu Zhang, Haizhong Wang, and Yugo Kato. A variational autoencoder based generative model of urban human mobility. In *2019 IEEE conference on multimedia information processing and retrieval (MIPR)*, pp. 425–430. IEEE, 2019.

Xiangjie Kong, Qiao Chen, Mingliang Hou, Hui Wang, and Feng Xia. Mobility trajectory generation: A survey. *Artificial Intelligence Review*, 56(Suppl 3):3057–3098, 2023. doi: 10.1007/s10462-023-10598-x. URL https://doi.org/10.1007/s10462-023-10598-x.

Kenneth Li, Aspen K Hopkins, David Bau, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Emergent world representations: Exploring a sequence model trained on a synthetic task. *arXiv preprint arXiv:2210.13382*, 2022.

Peibo Li, Maarten De Rijke, Hao Xue, Shuang Ao, Yang Song, and Flora D Salim. 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*, pp. 1463–1472, 2024.

Siyu Li, Toan Tran, Haowen Lin, John Krumm, Cyrus Shahabi, Lingyi Zhao, Khurram Shafique, and Li Xiong. Geo-llama: Leveraging llms for human mobility trajectory generation with constraints. In *2025 26th IEEE International Conference on Mobile Data Management (MDM)*, pp. 20–31. IEEE, 2025.

Siyu Li, Toan Tran, Lingyi Zhao, Khurram Shafique, and Li Xiong. TrajGenAgent: A hierarchical LLM agent for human mobility trajectory generation. In *Proceedings of the 27th IEEE International Conference on Mobile Data Management (MDM)*, 2026.

Xishun Liao, Qinhua Jiang, Brian Yueshuai He, Yifan Liu, Chenchen Kuai, and Jiaqi Ma. Deep activity model: A generative deep learning approach for human mobility pattern synthesis. *IEEE Transactions on Intelligent Transportation Systems*, 27(6):6519–6536, 2026. doi: 10.1109/TITS.2026.3678542. URL https://doi.org/10.1109/TITS.2026.3678542.

Yifan Liu, Xishun Liao, Haoxuan Ma, Brian Yueshuai He, Chris Stanford, and Jiaqi Ma. Human mobility modeling with household coordination activities under limited information via retrieval-augmented LLMs. In *2025 IEEE 28th International Conference on Intelligent Transportation Systems*, pp. 951–958. IEEE, 2025. doi: 10.1109/ITSC60802.2025.11423539. URL https://doi.org/10.1109/ITSC60802.2025.11423539.

Qingyue Long, Can Rong, Tong Li, and Yong Li. Dynamic population distribution aware human trajectory generation with diffusion model. *ACM Transactions on Intelligent Systems and Technology*, 17(3):69:1–69:23, 2026. doi: 10.1145/3787459. URL https://doi.org/10.1145/3787459.

Pablo Alvarez Lopez, Michael Behrisch, Laura Bieker-Walz, Jakob Erdmann, Yun-Pang Flötteröd, Robert Hilbrich, Leonhard Lücken, Johannes Rummel, Peter Wagner, and Evamarie Wießner. Microscopic traffic simulation using sumo. In *2018 21st international conference on intelligent transportation systems (ITSC)*, pp. 2575–2582. Ieee, 2018.

Xin Lu, Erik Wetter, Nita Bharti, Andrew J Tatem, and Linus Bengtsson. Approaching the limit of predictability in human mobility. *Scientific reports*, 3(1):2923, 2013.

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

Fan-Ming Luo, Tian Xu, Hang Lai, Xiong-Hui Chen, Weinan Zhang, and Yang Yu. A survey on model-based reinforcement learning. *Science China Information Sciences*, 67(2):121101, 2024.

Yingtao Luo, Qiang Liu, and Zhaocheng Liu. Stan: Spatio-temporal attention network for next location recommendation. In *Proceedings of the web conference 2021*, pp. 2177–2185, 2021.

Ryotaro Ohara, Hideaki Uchida, Yohei Yamaguchi, Shinya Yoshizawa, Katsuya Sakai, and Yoshiyuki Shimoda. Generating individual trajectories for synthetic populations with a large language model. In *Proceedings of the 5th Asia Conference of the International Building Performance Simulation Association*, pp. 1044–1051, 2024. doi: 10.69357/asim2024.1291. URL https://doi.org/10.69357/asim2024.1291.

Richard S Sutton. Dyna, an integrated architecture for learning, planning, and reacting. *ACM Sigart Bulletin*, 2(4):160–163, 1991.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Dongsheng Wang, Yuxi Huang, Shen Gao, Yifan Wang, Chengrui Huang, and Shuo Shang. Generative next poi recommendation with semantic id. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 2904–2914, 2025.

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

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

Yusong Wang, Chuang Yang, Jiawei Wang, Xiaohang Xu, Jiayi Xu, Dongyuan Li, Chuan Xiao, and Renhe Jiang. Ellmob: Event-driven human mobility generation with self-aligned llm framework. *arXiv preprint arXiv:2603.07946*, 2026.

Song Yang, Jiamou Liu, and Kaiqi Zhao. Getnext: Trajectory flow map enhanced transformer for next poi recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on research and development in information retrieval*, pp. 1144–1153, 2022.

Yuan Yuan, Jingtao Ding, Huandong Wang, Depeng Jin, and Yong Li. Activity trajectory generation via modeling spatiotemporal dynamics. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, pp. 4752–4762, 2022.

Kai Zhang, Xiangchao Chen, Bo Liu, Tianci Xue, Zeyi Liao, Zhihan Liu, Xiyao Wang, Yuting Ning, Zhaorun Chen, Xiaohan Fu, et al. Agent learning via early experience. *arXiv preprint arXiv:2510.08558*, 2025.

Kunyi Zhang, Yanbo Pang, Yurong Zhang, and Yoshihide Sekimoto. Mobglm: A large language model for synthetic human mobility generation. In *Proceedings of the 32nd ACM International Conference on Advances in Geographic Information Systems*, pp. 629–632, 2024.

Xin Zhang, Yanhua Li, Xun Zhou, Ziming Zhang, and Jun Luo. Trajgail: Trajectory generative adversarial imitation learning for long-term decision analysis. In *2020 IEEE International Conference on Data Mining (ICDM)*, pp. 801–810. IEEE, 2020.

Yuanshao Zhu, Yongchao Ye, Ying Wu, Xiangyu Zhao, and James Yu. Synmob: Creating high-fidelity synthetic gps trajectory dataset for urban mobility analysis. *Advances in Neural Information Processing Systems*, 36:22961–22977, 2023a.

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

A User-Specific Context Representation

MotiveMob conditions both predictors on two sources of user-specific information: a long-term behavioral report $\mathcal{R}_u$ and a recent mobility history $\mathcal{H}_i$. This section specifies their textual representations.

A.1 Behavioral Report

The behavioral report summarizes recurring mobility regularities derived from the user's training trajectories. It contains the typical starting and ending check-ins on weekdays and weekends, together with frequently visited POIs and their representative visit times. The report supplied to the predictors follows the template below.

```
During weekday, you usually begin your daily trip at
<weekday-start-POI> at <HH:MM:SS> and end your daily trip at
<weekday-end-POI> at <HH:MM:SS>.

During weekend, you usually begin your daily trip at
<weekend-start-POI> at <HH:MM:SS> and end your daily trip at
<weekend-end-POI> at <HH:MM:SS>.

You usually visit:
<frequent-POI-1> at <HH:MM:SS>
<frequent-POI-2> at <HH:MM:SS>
...
```

Each POI is represented as `<category>#<ID>`, where the ID distinguishes venues belonging to the same category.

A.2 Recent Mobility History

The mobility history contains up to the 30 most recent check-ins, retained in chronological order, from the three active days immediately preceding the target day:

```
<POI-1>#<ID> at <HH:MM:SS>
<POI-2>#<ID> at <HH:MM:SS>
...
<POI-K>#<ID> at <HH:MM:SS>
```

where $K \leq 30$.

B Mobility Motivation Taxonomy

Table 3 summarizes the motivation taxonomy used in our experiments. Each behavioral motivation is deterministically assigned according to the category of the destination venue. The `END` label is added separately after the final check-in of each trajectory.

Table 3: Mobility motivation taxonomy and its mapping from destination venue categories. The listed venues summarize the category groups associated with each motivation.

Motivation	Behavioral interpretation	Destination venue categories
Return	Returning to a residence or temporary accommodation	Residential venues, hotels, hostels, inns, resorts, and other lodging venues
Personal Service	Obtaining a personal, household, financial, repair, or travel-related service	Salons, banks, repair shops, rental services, post offices, travel agencies, and related service venues
Entertainment	Participating in leisure, cultural, or recreational activities	Arts and entertainment, landmarks and outdoors, sports and recreation, cruises, auditoriums, and related venues
Dining	Eating or drinking	Dining and drinking venues, food courts, and cafeterias
Transit	Traveling or transferring between places	Airports, stations, public transport facilities, roads, and other transportation venues
Religious	Participating in religious or spiritual activities	Shrines, temples, churches, mosques, monasteries, and other religious venues
Education	Attending educational or learning activities	Schools, universities, libraries, classrooms, laboratories, and training institutions
Work	Engaging in occupational or professional activities	Offices, government facilities, administrative buildings, and other professional workplaces
Social	Participating in social, civic, or community activities	Event venues, wedding halls, funeral homes, and community or civic venues
Shopping	Purchasing goods	Retail venues and food-and-beverage retailers
Healthcare	Receiving medical or health-related care	Hospitals, clinics, pharmacies, and other health and medicine venues
`END`	Terminating the daily trajectory	Added after the final check-in; not associated with a destination venue

C Calibration of the Speed-Feasibility Threshold

We calibrate the speed threshold using only the observed transitions of the 25 seen users from January 1 to October 31, 2019. After removing one `START` row from each of the 3,559 training trajectories, the data contain 27,504 consecutive check-in transitions. For a transition from (l_i, t_i) to (l_{i+1}, t_{i+1}), its implied speed is computed as

$$v_i = \frac{d_{\mathrm{hav}}(l_i, l_{i+1})}{\max(\Delta q_i, 1) \times 10/60}, \tag{6}$$

where d_{hav} is the Haversine distance in kilometers and Δq_i is the difference between the corresponding 10-minute time slots. This definition is identical to that used by the speed-feasibility check during inference.

We exclude 14 transitions involving non-spatial states, such as `Home (private)`, and 418 transitions with displacements exceeding 100 km, leaving 27,072 transitions for calibration. The 100 km criterion prevents a small number of extreme long-distance transitions from dominating the threshold estimate and is applied only during calibration, not as a distance constraint during inference.

Because the resulting speed distribution remains strongly right-skewed, we apply the transformation

$$y_i = \log\left(1 + \frac{v_i}{v_0}\right), \qquad v_0 = 1\ \mathrm{km/h}, \tag{7}$$

which retains zero-speed transitions while compressing the upper tail.

As shown in Figure 3, the transformation substantially reduces the skewness of the speed distribution, although the Q–Q plot still shows deviations from exact Gaussianity.

$$\mu_y = 1.4927, \qquad \sigma_y = 1.2196, \tag{8}$$

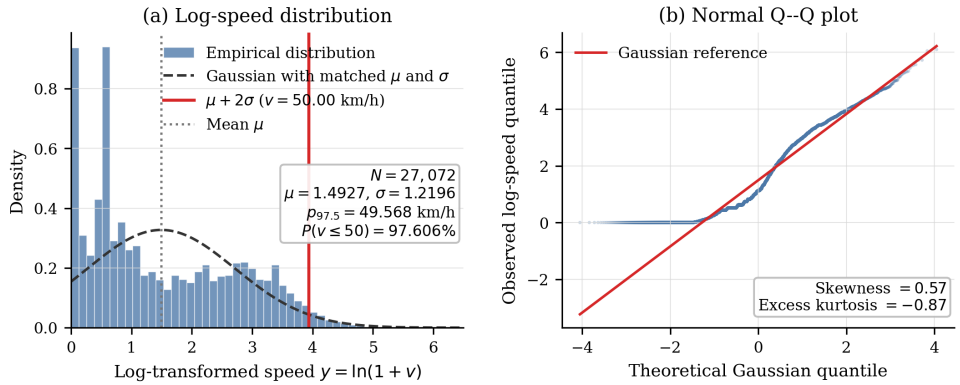

Figure 3: Distributional diagnostics for the log-transformed transition speeds used to calibrate $v_{\max}$. (a) Empirical distribution and a Gaussian distribution with the same mean and standard deviation. The red line marks the two-standard-deviation upper bound. (b) Normal Q–Q plot showing the remaining deviations from Gaussianity, particularly in the tails.

giving the following upper bound on the original speed scale:

$$v_0 \left[\exp(\mu_y + 2\sigma_y) - 1\right] = 50.003 \text{ km/h}. \tag{9}$$

This value closely matches the empirical 97.5th percentile,

$$Q_{0.975}(\{v_i\}) = 49.568 \text{ km/h}. \tag{10}$$

We therefore set $v_{\max} = 50$ km/h, which covers 97.606% of the cleaned training transitions. The empirical percentile provides the primary calibration criterion, while the log-space two-standard-deviation bound serves as a consistent distributional check. The resulting threshold is fixed across all user groups and temporal periods without using any unseen-user or unseen-period data.

D Implementation Details and Configurations

We use Llama-3.1-8B-Instruct as the base model for the LLM judge, the state predictor P_S, and the motivation predictor P_M. The LLM judge remains frozen during alternative transition construction, whereas P_S and P_M are independently adapted from the base model using QLoRA. This appendix provides implementation details and configurations for three stages: alternative transition construction, predictor training, and autoregressive trajectory inference.

D.1 Alternative Transition Construction

Table 4 summarizes the data, candidate-generator, and LLM-judge configurations used to construct alternative transitions. For each observed transition, we sample three distinct motivations other than the observed one. The candidate generator first proposes possible next states for each alternative motivation, after which the LLM judge selects the most plausible candidate under the user's behavioral context.

The judge receives the target date and weekend indicator, the user's long-term behavioral report, recent mobility history, earlier check-ins on the target day, the current location and time, the alternative motivation, and the candidate states. Each candidate is represented by its location, coordinates, arrival time, activity category, distance from the current state, and candidate-generator score. The judge is instructed to return only the index of the candidate that is most consistent with temporal validity, spatial feasibility, the specified motivation, and the user's historical behavior.

D.2 Predictor Training

The state predictor and motivation predictor are trained independently. The state predictor is trained on a balanced mixture of observed and judged alternative transitions, preventing the larger set of

Table 4: Configuration for alternative-transition construction.

Parameter	Setting
Construction data	25 seen users from January–October 2019, containing 3,559 daily trajectories and 31,063 observed transition instances
Alternative motivations	Three distinct motivations per observed transition, sampled without replacement while excluding the observed motivation
Prepared alternative cases	93,189 cases before candidate generation and judging; 31,063 judged alternative transitions are used for balanced state-predictor training
Candidate-generator architecture	Three-layer pre-layer-normalized Transformer encoder with hidden dimension 128, 4 attention heads, feed-forward dimension 256, and dropout 0.1
Prediction heads	Separate classification heads for location subcategory, spatial grid, and arrival-time bin
Candidate-generator objective	Equal-weighted cross-entropy losses for the subcategory, spatial-grid, and time-bin prediction heads
Candidate-generator split	90% training and 10% validation dates within each user, split by complete user–date groups with seed 42
Candidate-generator optimization	AdamW with learning rate 1×10^{-3}, weight decay 1×10^{-4}, batch size 2,048, and 30 training epochs
Learning-rate schedule	Two warm-up epochs followed by cosine decay with a minimum learning-rate ratio of 0.05
Additional regularization	Gradient clipping at 1.0 and label smoothing of 0.05
Checkpoint selection	The checkpoint with the highest mean top-1 validation accuracy over the subcategory, spatial-grid, and time-bin heads
Candidate enumeration	Top 8 subcategories, top 8 spatial grids, and top 8 time bins are combined into at most 50 preliminary triples, from which at most 10 candidates are retained for judging
Candidate-scoring weights	$\alpha_{\mathrm{sub}} = 0.8$, $\alpha_{\mathrm{grid}} = 0.9$, $\alpha_{\mathrm{time}} = 0.55$, $\alpha_{\mathrm{user}} = 1.0$
Candidate filtering	Motivation-strict filtering and behavioral-report-based distance capping are disabled; equal-time alternative transitions are not allowed
Candidate context	The previous three active mobility days, the preceding check-ins on the target day, the current state, the alternative motivation, and the user's long-term behavioral report
LLM judge	Frozen Llama-3.1-8B-Instruct loaded with 4-bit NF4 quantization, double quantization, and bfloat16 computation when supported
Judge decoding	Greedy decoding with at most 8 newly generated tokens
Judge decision	The judge must select one candidate when the candidate set is nonempty. An invalid or unparsable output defaults to the highest-ranked candidate; no alternative transition is retained when the candidate set is empty
ρ	0.33
Construction seed	3407

initially constructed alternatives from dominating its training objective. The motivation predictor is trained only on observed step-level motivations and trajectory-termination labels. Table 5 reports their training configurations.

Table 5: Training configurations for the state predictor P_S and motivation predictor P_M.

Parameter	State predictor P_S	Motivation predictor P_M
Initialization	Llama-3.1-8B-Instruct	Independently initialized from Llama-3.1-8B-Instruct
Training data	31,063 observed transitions and 31,063 judged alternative transitions	34,622 observed instances, comprising 31,063 step-level motivations and 3,559 `END` labels
Observed-to-alternative ratio	1:1	Not applicable
Training objective	Assistant-token-only causal language-modeling loss	Assistant-token-only causal language-modeling loss
Quantization	4-bit NF4 with double quantization and bfloat16 computation	4-bit NF4 with double quantization and bfloat16 computation
QLoRA configuration	Rank $r = 16$, scaling factor $\alpha = 16$, dropout 0, and no bias	Rank $r = 16$, scaling factor $\alpha = 16$, dropout 0, and no bias
LoRA target modules	`q_proj`, `k_proj`, `v_proj`, `o_proj`, `gate_proj`, `up_proj`, and `down_proj`	`q_proj`, `k_proj`, `v_proj`, `o_proj`, `gate_proj`, `up_proj`, and `down_proj`
Optimizer	8-bit paged AdamW	8-bit paged AdamW
Learning rate	1×10^{-4}	2×10^{-4}
Weight decay	0.01	0.01
Learning-rate schedule	Linear decay with 200 warm-up steps	Linear decay with 200 warm-up steps
Batch size	2 per GPU with 8 gradient-accumulation steps on 8 GPUs, giving an effective batch size of 128	2 per GPU with 8 gradient-accumulation steps on 8 GPUs, giving an effective batch size of 128
Training budget	4,000 optimization steps	12 epochs; the checkpoint at step 3,084 is used for inference
Maximum sequence length	2,048 tokens	2,048 tokens
Mobility-history context	Up to 30 check-ins from the three active days preceding the target day	Up to 30 check-ins from the three active days preceding the target day
Training seed	3407	3407

D.3 Trajectory Inference

Trajectory generation starts from a `START` state and alternates between motivation prediction and state prediction. The motivation predictor either produces the next behavioral motivation or returns `END`. Given a non-terminal motivation, the state predictor generates a candidate location and arrival time. The candidate is then validated before being appended to the trajectory. Table 6 summarizes the inference configuration.

Table 6: Configuration for autoregressive trajectory inference.

Parameter	Setting
Initial state	A `START` state; the first motivation and check-in are generated without applying a travel-speed constraint because no previous spatial state is available
Mobility-history context	Up to 30 check-ins from the three active days preceding the target day
Motivation decoding	Greedy decoding with at most 8 newly generated tokens
State decoding	Sampling with temperature 0.2, top-p 0.9, and at most 32 newly generated tokens
State-generation attempts	At most 8 proposals per step, consisting of one initial proposal and up to seven retries
Retry decoding	Temperature increases from 0.3 to 0.9 in increments of 0.1; top-p remains 0.9 and the maximum output length remains 32 tokens
Output validation	The generated output must be parseable as a valid location–time state with a recognized location identifier and timestamp
Temporal validation	The generated arrival time must not precede the current time; generation terminates when the predicted state crosses into the next mobility day
Travel-speed calculation	Haversine distance divided by the elapsed time between consecutive states, with time represented using 10-minute slots
Speed threshold	$v_{\max} = 50$ km/h; its selection is described in Appendix C
Speed-based retry	A proposal exceeding $v_{\max}$ is rejected and regenerated using the retry schedule above
Re-grounding after retries	If all eight proposals violate the speed constraint, the final proposal is re-grounded, when possible, to a location of the same broad activity category that appears in the user's history or behavioral report and satisfies $v_{\max}$; feasible locations are ranked by time-of-day similarity with historical frequency used as a tie-breaker
Re-grounding failure	If no feasible location is available for re-grounding, the final state-predictor proposal is retained
Additional fallbacks	The legacy distance-based fallback and history-trajectory replacement fallback are disabled
Repetition termination	Activated after 15 accepted check-ins. Generation terminates before appending a candidate if the candidate-augmented location sequence contains a block of length 1–4 repeated for at least three consecutive rounds
Online repetition guard	Disabled; repetition is handled only by the termination rule above
Maximum trajectory length	25 generated check-ins

E Prompt Templates

All prompts are rendered using the chat template of Llama-3.1-8B-Instruct. Text enclosed by angle brackets below denotes sample-specific content.

E.1 LLM Judge for Alternative Transition Construction

For each alternative motivation, the lightweight Transformer produces a candidate set of next states. The LLM judge selects one candidate without generating a new location or timestamp.

System prompt

```
You are a careful urban mobility candidate judge.
You must choose exactly one candidate next check-in from a provided candidate list.
Do not invent a new place or time.
```

User prompt

```
<INPUT 0> Behavior Pattern:
```

```
<BEHAVIORAL_REPORT>

<INPUT 1> Recent History Before Today:
<RECENT_HISTORY_OR_NONE>

<INPUT 2> Earlier Check-ins Today:
<EARLIER_CHECKINS_OR_NONE>

<INPUT 3> Current Location & Time:
<CURRENT_STATE> [lat=<LATITUDE>, lng=<LONGITUDE>]

<INPUT 4> Target Motivation for the NEXT transition:
<ALTERNATIVE_MOTIVATION>

<INPUT 5> Date and Weekend Flag:
date=<YYYY-MM-DD>, is_weekend=<0_OR_1>

<CANDIDATES>
1. <LOCATION>#<ID> at <HH:MM:SS>
   [transformer_score=<SCORE>;
    top_category=<CATEGORY>;
    candidate_motivation=<MOTIVATION>;
    lat=<LATITUDE>, lng=<LONGITUDE>;
    distance_from_current_km=<DISTANCE>]
...
K. <LOCATION>#<ID> at <HH:MM:SS> [<METADATA>]

Task:
Choose the single best candidate for the next check-in. Prefer candidates
that fit the target motivation, the user's recent behavior, the current
state, and temporal order.

Output format:
Output ONLY the candidate number, such as 1 or 7. Do not output any other text.
```

E.2 STATE PREDICTOR

The same state-prediction prompt is used for observed and alternative transitions during training. At inference time, the target motivation is the output of P_M.

System prompt

```
You are an urban mobility transition model.
Given a behavior pattern, recent history checkins, current location and
time, current motivation, today's date, and is_weekend, predict the next
location and time.
Follow the required output format exactly.
```

User prompt

```
<INPUT 0> (Behavior Pattern - long-term prior):
<BEHAVIORAL_REPORT>

<INPUT 1> (Recent History Checkins):
<RECENT_HISTORY>

<INPUT 2> (Current Location & Time):
<CURRENT_STATE>

<INPUT 3> (Target Motivation for the NEXT transition):
<MOTIVATION>

<INPUT 4> (Today's Date):
<YYYY-MM-DD>

<INPUT 5> (Is Weekend, 1=yes 0=no):
<0_OR_1>

Task:
Predict the NEXT location and time.

Rules:
- Recent history and current state are more important than the behavior pattern.
- Use the behavior pattern only as a general long-term reference.
- The next timestamp must not be earlier than the current timestamp under
  the day definition starting at 04:00:00.
```

```
Output format (IMPORTANT):
<Location>#<ID> at HH:MM:SS
Do NOT output any extra text.

Example:
Park#12 at 09:10:00
```

E.3 MOTIVATION PREDICTOR

The motivation predictor does not observe the next state or any subsequent state. It predicts either the motivation underlying the next transition or the termination label END.

System prompt

```
You are an urban mobility intention inference model.
Given a behavior pattern, recent history Checkins, the current location
and time, today's date, and is_weekend, infer the motivation for the next
transition from the current state.
Output exactly one label from the provided list.
```

User prompt

```
<INPUT 0> (Behavior Pattern - long-term prior):
<BEHAVIORAL_REPORT>

<INPUT 1> (Recent History Checkins):
<RECENT_HISTORY>

<INPUT 2> (Current Location & Time):
<CURRENT_STATE>

<INPUT 3> (Today's Date):
<YYYY-MM-DD>

<INPUT 4> (Is Weekend, 1=yes 0=no):
<0_OR_1>

Task:
Infer the motivation for the next transition.

Rules:
- Recent history is more important than the behavior pattern.

Allowed labels (choose exactly ONE):
- Return
- Personal Service
- Entertainment
- Dining
- Transit
- Religious
- Education
- Work
- Social
- Shopping
- Healthcare
- END

Output format (IMPORTANT):
- Output EXACTLY ONE line.
- The line MUST be exactly one of the allowed labels.
- Do NOT output any extra text.
```

F MOTIVATION-GUIDED TRAJECTORY GENERATION PSEUDO CODES

This section presents the pseudocode for the complete inference-time rollout, including feasibility-aware resampling, repetition-based termination, and mobility-grounded recovery.

Algorithm 1 Motivation-guided trajectory generation

Require: Models P_M and P_S; user context x_u; observed locations $\mathcal{O}_u$
Require: $v_{\max} = 50$ km/h; $R = 8$; $L_{\min} = 15$; $L_{\max} = 25$
Ensure: Generated trajectory $\mathcal{T}$

$\mathcal{T} \leftarrow [\,]$
$c_0 \leftarrow \textsc{BuildContext}(x_u, \texttt{START}, \mathcal{T})$
for $i = 0, \ldots, L_{\max} - 1$ **do**
  $\hat{m}_i \leftarrow \arg \max_{y \in \mathcal{M} \cup \{\texttt{END}\}} p_{\theta_M}(y \mid c_i)$
  if $\hat{m}_i = \texttt{END}$ **then**
    return $\mathcal{T}$
  end if
  $s^\star \leftarrow \varnothing$
  $\tilde{s}^{\text{last}} \leftarrow \varnothing$
  for $r = 1, \ldots, R$ **do**
    if $r = 1$ **then**
      $T_r \leftarrow 0.2$
    else
      $T_r \leftarrow 0.3 + 0.6(r-2)/(R-2)$
    end if
    $\tilde{s} \sim p_{\theta_S}^{(T_r,\, p=0.9)}(\cdot \mid c_i, \hat{m}_i)$
    $\tilde{s} \leftarrow \textsc{ParseState}(\tilde{s})$
    if $\tilde{s} = \varnothing$ **then**
      continue
    end if
    $\tilde{s}^{\text{last}} \leftarrow \tilde{s}$
    if $i > 0$ and $\textsc{ArrivalTime}(\tilde{s}) < \textsc{ArrivalTime}(s_{i-1})$ **then**
      continue
    end if
    if $i > 0$ and $\textsc{ImpliedSpeed}(s_{i-1}, \tilde{s}) > v_{\max}$ **then**
      continue
    end if
    $s^\star \leftarrow \tilde{s}$
    break
  end for
  if $s^\star = \varnothing$ **then**
    if $\tilde{s}^{\text{last}} = \varnothing$ **then**
      return $\mathcal{T}$
    end if
    $s^\star \leftarrow \textsc{ReGround}(\tilde{s}^{\text{last}}, s_{i-1}, \mathcal{O}_u, v_{\max})$
    if $s^\star = \varnothing$ **then**
      return $\mathcal{T}$
    end if
  end if
  $\mathcal{L}^+ \leftarrow \textsc{Locations}(\mathcal{T} \,\|\, [s^\star])$
  if $|\mathcal{T}| \geq L_{\min}$ and $\textsc{RepeatedCycle}(\mathcal{L}^+, 4, 3, \{1,2,3,4\})$ **then**
    return $\mathcal{T}$
  end if
  $\mathcal{T} \leftarrow \mathcal{T} \,\|\, [s^\star]$
  $s_i \leftarrow s^\star$
  $c_{i+1} \leftarrow \textsc{BuildContext}(x_u, s_i, \mathcal{T})$
end for
return $\mathcal{T}$

Algorithm 2 Repetition-aware termination

function REPEATEDCYCLE($\mathcal{L}, B, K, \mathcal{P}$)
  $n \leftarrow |\mathcal{L}|$
  for $b = 1, \ldots, \min\{B, \lfloor n/K \rfloor\}$ **do**
    for $a = 1, \ldots, n - Kb + 1$ **do**
      $W \leftarrow \mathcal{L}[a : a + Kb - 1]$
      $Q \leftarrow W[1 : b]$
      if $W = Q^K$ and PRIMITIVEPERIOD(W) $\in \mathcal{P}$ **then**
        return true
      end if
    end for
  end for
  return false
end function

Algorithm 3 Mobility-grounded recovery after retry exhaustion

function REGROUND($\tilde{s}, s_{\text{prev}}, \mathcal{O}_u, v_{\max}$)
  $(\tilde{\ell}, \tilde{t}) \leftarrow$ SPLITSTATE($\tilde{s}$)
  $a \leftarrow$ ACTIVITYCATEGORY($\tilde{\ell}$)
  $\mathcal{C} \leftarrow$ OBSERVEDCANDIDATES($\mathcal{O}_u, a$)
  $\mathcal{F} \leftarrow \emptyset$
  for all $(\ell, t^{\text{obs}}) \in \mathcal{C}$ **do**
    $s' \leftarrow (\ell, \tilde{t})$
    if $s_{\text{prev}} =$ `START` or IMPLIEDSPEED(s_{prev}, s') $\leq v_{\max}$ **then**
      $\mathcal{F} \leftarrow \mathcal{F} \cup \{(\ell, t^{\text{obs}})\}$
    end if
  end for
  if $\mathcal{F} = \emptyset$ **then**
    return $\emptyset$
  end if
  $(\ell^\star, t^\star) \leftarrow \arg\min_{(\ell, t^{\text{obs}}) \in \mathcal{F}} \left(\left| \text{TIMESLOT}(t^{\text{obs}}) - \text{TIMESLOT}(\tilde{t}) \right|, -\text{VISITFREQUENCY}(u, \ell) \right)$
  return $(\ell^\star, \tilde{t})$
end function

G Evaluation-Split Construction and Leakage Prevention

We distinguish three roles of mobility trajectories in our experiments. *Parameter-training trajectories* are used to optimize the candidate generator, the state predictor P_S, and the motivation predictor P_M. *Support trajectories* are used only to construct user-specific context at inference time. *Query trajectories* are reserved exclusively as evaluation targets. In particular, support trajectories from unseen users never participate in parameter optimization.

G.1 Seen Users in the Seen Period

For each of the 25 seen users, we randomly divide the daily trajectories collected between January and October 2019 into disjoint training and query subsets using an approximately 80/20 ratio. After per-user rounding, this produces 3,559 parameter-training trajectories and 908 held-out query trajectories. We verify that the two subsets contain no common $(\text{user}, \text{date})$ pair.

Both P_S and P_M are trained exclusively from the 3,559 training trajectories. Each trajectory contributes one `END` instance to the training data of P_M; therefore, the 3,559 `END` instances reported in Table 5 correspond exactly to the parameter-training trajectories and do not include any of the 908 evaluation trajectories.

The candidate generator further divides the same 3,559 trajectories into 90% training dates and 10% validation dates for checkpoint selection. This is an internal training/validation division and is independent of the 908-trajectory final evaluation set. We therefore treat the seen-user, seen-period setting as an *in-distribution held-out* evaluation rather than a user- or time-shift generalization setting.

G.2 Inference Context for Unseen Users

The 50 unseen users are disjoint from the 25 seen users, and none of their trajectories is used to optimize the candidate generator, P_S, or P_M. For the unseen-user, seen-period evaluation, their January–October 2019 trajectories are divided into 5,431 support trajectories and 1,345 held-out query trajectories. These subsets contain no common $(\text{user}, \text{date})$ pair.

For an unseen user u, the behavioral report R_u is constructed only from that user's support trajectories. For a query trajectory $\tau_{u,d}$, the recent mobility history $H_{u,d}$ contains up to 30 check-ins from the three most recent active support days strictly preceding the target date d. The target trajectory $\tau_{u,d}$ and all query trajectories are excluded from both report and history construction.

Thus, "unseen user" means unseen during parameter learning, rather than a zero-history cold-start setting. The corresponding experiment evaluates generalization to new users given a limited, inference-only mobility context.

G.3 Context Used in the Temporal-Shift Settings

For the regular unseen period, spanning November 2019 through February 2020, the behavioral report is constructed from January–October 2019 support trajectories. The report window therefore precedes the complete evaluation period. The recent history $H_{u,d}$ is selected from a support pool disjoint from the query trajectories and contains only observations strictly preceding d.

For the COVID-19 period, spanning April through October 2020, the behavioral report is updated using only mobility observations from March 2020. The recent history is again restricted to support trajectories strictly preceding the target date. Consequently, neither the target-day trajectory nor a future trajectory enters R_u or $H_{u,d}$ in either temporal-shift evaluation.

The two seen-period settings use an IID trajectory holdout rather than a chronological forecasting split. In these settings, R_u is a static profile constructed from the user's complete January–October support subset. Accordingly, it may summarize a support trajectory dated after a particular query date, but it never contains the query trajectory itself or any other trajectory from the query subset. We therefore do not interpret the seen-period results as causal future forecasting. By contrast, $H_{u,d}$ is always restricted to support observations preceding d.

Table 7: Data separation and information available under the evaluation protocol.

Protocol component	Data usage and restriction
Predictor training	Trajectories from 25 seen users between January and October 2019. For the seen-user/seen-period evaluation, 20% of user–date trajectories are held out and excluded from predictor training.
User-level shift	The 50 unseen users are disjoint from the 25 users used to train P_M and P_S.
Temporal shift	The regular unseen period (November 2019–February 2020) and the COVID-19 period (April–October 2020) do not overlap with the predictor-training period.
Recent history $\mathcal{H}_{u,d}$	Contains at most 30 check-ins from the three active days strictly preceding target date d. Target-day and future observations are never included.
Behavioral report R_u	In the temporal-shift settings, the report is constructed only from observations preceding the evaluation period. In the seen-period settings, it is a static aggregate profile constructed from the query-disjoint support subset and may include support dates later than the target date.
COVID-19 update	For targets in April–October 2020, the report is updated using observations available through March 2020, without using any check-ins from the target period.
Re-grounding	Candidate locations are drawn only from the same R_u and $H_{u,d}$ provided to the predictors. Re-grounding does not access the query trajectory or any additional trajectory database.

G.4 Information Available to Re-grounding

The inference-time re-grounding procedure does not access the complete trajectory database or the ground-truth query trajectory. Its candidate pool contains only POI–time pairs extracted from the same behavioral report R_u and recent mobility history $H_{u,d}$ supplied to P_S and P_M. Re-grounding therefore introduces no additional user information beyond the context already visible to the two predictors.

The two seen-period settings use non-chronological held-out splits under the observed-period distribution. In both settings, R_u is constructed exclusively from the user's support subset, but it may aggregate behavior from a support trajectory dated after a particular query date. Importantly, R_u contains only aggregate behavioral regularities—such as typical starting and ending activities and frequently visited POIs with representative visit times—rather than any complete day-level trajectory, and it never contains the target query trajectory or any other trajectory from the query subset. We therefore interpret the seen-user/seen-period setting as an in-distribution held-out evaluation and the unseen-user/seen-period setting as a user-level transfer evaluation under the observed-period distribution; neither is intended as causal future forecasting. By contrast, $H_{u,d}$ is always restricted to support observations strictly preceding d.

H Osaka Results

We report the evaluation results on the Osaka human check-in dataset, following the same user-level and temporal splitting protocols used for the Tokyo dataset. To evaluate user-level generalization, we use trajectories from 25 seen users to train the base model and reserve a disjoint set of 50 additional users exclusively for evaluation. No mobility records from these unseen users are available during training. To evaluate temporal generalization, check-ins collected from January to October 2019 constitute the seen training period. We further consider two unseen periods: November 2019 to February 2020, which captures a regular seasonal shift, and April to October 2020, which captures the pandemic-induced mobility shift following the onset of COVID-19 restrictions.

Across all 24 combinations of user group, time period, and metric, MotiveMob consistently achieves the best performance. For seen users, it reduces the four divergences by an average of 36.3% relative to the corresponding runner-up, with particularly large gains on SD, SI, and DARD. The improvements on STVD are smaller (0.7–2.8%), but remain consistent across all three periods. For unseen users, MotiveMob retains the best results in every setting and yields an average relative error reduction of 31.1% over the runner-up. Although its advantages on SI and STVD narrow under the

Table 8: Evaluation results on seen users in Osaka across seen and unseen time periods. The upper and lower baseline sections correspond to pretraining-based and prompting-based methods, respectively. Lower values are better; the best and second-best results are highlighted in bold and underlined, respectively.

Method	Seen period (1186 trajectories)				Regular unseen period (466 trajectories)				COVID-19 unseen period (689 trajectories)			
	SD	SI	DARD	STVD	SD	SI	DARD	STVD	SD	SI	DARD	STVD
LSTM	0.0914	0.1375	0.2673	0.4161	0.1109	0.1698	0.3230	0.5779	0.1259	0.2073	0.3556	0.5725
MHSA	0.0908	0.1357	0.2468	0.4286	0.1210	0.1817	0.2706	0.5803	0.1262	0.2152	0.3044	0.5552
DeepMove	0.1777	0.1024	0.2348	0.4933	0.1645	0.1267	0.3180	0.6065	0.2308	0.1595	0.3340	0.6029
GetNext	0.0395	0.2820	0.2149	0.4635	0.0433	0.2834	0.2057	0.5601	0.0455	0.2952	0.1953	0.5294
TrajGAIL	0.2251	0.1256	0.0799	0.5836	0.2458	0.1227	0.1250	0.6576	0.2494	0.1118	0.1137	0.6424
DiffTraj	0.2002	0.1217	0.1407	0.6874	0.2092	0.1165	0.1662	0.6924	0.1869	0.1025	0.1907	0.6904
GNPR-SID	0.1121	0.0212	0.0988	0.4642	0.1319	0.0346	0.1262	0.5786	0.1889	0.0434	0.1711	0.5956
LLM4POI	0.0323	0.0192	0.0900	0.4093	0.0463	0.0340	0.1309	0.5408	0.0424	0.0512	0.1278	0.5139
LLMMove	0.0440	0.0453	0.1102	0.5022	0.0528	0.0518	0.1866	0.5907	0.0455	0.0630	0.1556	0.5856
LLM-ZS	0.0352	0.0521	0.1881	0.5124	0.0300	0.0534	0.2272	0.5880	0.0274	0.0648	0.1958	0.5596
LLM-Mob	0.0338	0.0554	0.2117	0.4977	0.0309	0.0615	0.2741	0.5901	0.0382	0.0656	0.2819	0.5777
LLMob	0.1153	0.2542	0.1435	0.5276	0.0537	0.0980	0.1334	0.5827	0.0530	0.0858	0.1225	0.5433
ELLMob	0.0160	0.0274	0.0939	0.4127	0.0205	0.0281	0.1496	0.5302	0.0209	0.0334	0.1442	0.5405
MotiveMob	**0.0038**	**0.0078**	**0.0340**	**0.4035**	**0.0097**	**0.0165**	**0.0884**	**0.5153**	**0.0125**	**0.0207**	**0.0723**	**0.5102**

Table 9: Evaluation results on unseen users in Osaka across seen and unseen time periods. The upper and lower baseline sections correspond to pretraining-based and prompting-based methods, respectively. Lower values are better; the best and second-best results are highlighted in bold and underlined, respectively.

Method	Seen period (2804 trajectories)				Regular unseen period (1655 trajectories)				COVID-19 unseen period (1301 trajectories)			
	SD	SI	DARD	STVD	SD	SI	DARD	STVD	SD	SI	DARD	STVD
LSTM	0.1489	0.2437	0.3872	0.6207	0.1740	0.2420	0.3810	0.6569	0.1493	0.2645	0.4358	0.6551
MHSA	0.1897	0.1965	0.3578	0.5961	0.1715	0.2352	0.3762	0.6466	0.1582	0.2500	0.4182	0.6589
DeepMove	0.2754	0.2267	0.3548	0.6510	0.2746	0.2257	0.3876	0.6714	0.2709	0.2338	0.3918	0.6704
GetNext	0.0832	0.3220	0.2568	0.5752	0.0752	0.3215	0.2376	0.6408	0.0994	0.3351	0.2511	0.6438
TrajGAIL	0.1737	0.0825	0.1287	0.5750	0.1954	0.0912	0.1578	0.6513	0.1915	0.0943	0.1427	0.6532
DiffTraj	0.1977	0.0719	0.0932	0.6894	0.1944	0.0832	0.1283	0.6919	0.2159	0.0815	0.1405	0.6899
GNPR-SID	0.1367	0.0706	0.1290	0.4673	0.1353	0.0626	0.1272	0.5632	0.1481	0.0643	0.1325	0.5925
LLM4POI	0.0334	0.0699	0.1282	0.4435	0.0382	0.0763	0.1477	0.5682	0.0535	0.0859	0.1595	0.5663
LLMMove	0.0467	0.0602	0.1255	0.4674	0.0504	0.0663	0.1608	0.5834	0.0427	0.0737	0.1615	0.5831
LLM-ZS	0.0226	0.0709	0.1762	0.5005	0.0345	0.0909	0.2069	0.5759	0.0328	0.0875	0.1733	0.5910
LLM-Mob	0.0734	0.0625	0.2746	0.4667	0.0715	0.0576	0.2992	0.5824	0.0821	0.0761	0.3141	0.5747
LLMob	0.0583	0.0948	0.0902	0.4756	0.0702	0.1130	0.1337	0.5766	0.0623	0.1243	0.1415	0.5865
ELLMob	0.0227	0.0366	0.1226	0.4255	0.0260	0.0383	0.1820	0.5467	0.0264	0.0408	0.1657	0.5514
MotiveMob	**0.0075**	**0.0251**	**0.0363**	**0.3872**	**0.0143**	**0.0268**	**0.0696**	**0.5149**	**0.0214**	**0.0385**	**0.0665**	**0.5219**

COVID-19 shift, MotiveMob remains strongest overall, demonstrating robust generalization to both unseen users and temporal distribution shifts. Beyond user groups and temporal shifts, MotiveMob also performs consistently well across cities, achieving competitive results on both Tokyo and Osaka.

I MotiveMob Results Across LLM Backbones

Table 10 shows that MotiveMob achieves consistently strong results across different LLM backbones, indicating that its effectiveness is not dependent on a particular backbone model.

Table 10: MotiveMob results with different LLM backbones for seen and unseen users across the three temporal settings. All values are JSD; lower is better.

Backbone	Seen period				Regular unseen period				COVID-19 unseen period			
	SD	SI	DARD	STVD	SD	SI	DARD	STVD	SD	SI	DARD	STVD
Seen users												
Qwen3.5-9B	0.0080	0.0088	0.0671	0.3552	0.0096	0.0116	0.0902	0.4464	0.0452	0.0173	0.1396	0.5728
Gemma-4-E4B	0.0081	0.0191	0.0828	0.4041	0.0114	0.0192	0.0953	0.4731	0.0425	0.0200	0.1300	0.5431
Gemma-2-9B	0.0099	0.0064	0.0641	0.3760	0.0097	0.0118	0.0889	0.4296	0.0482	0.0138	0.1319	0.5696
Unseen users												
Qwen3.5-9B	0.0253	0.0268	0.0663	0.4798	0.0358	0.0288	0.0913	0.5411	0.0308	0.0278	0.1198	0.6431
Gemma-4-E4B	0.0166	0.0373	0.0857	0.4913	0.0184	0.0382	0.1022	0.5514	0.0242	0.0299	0.1121	0.6267
Gemma-2-9B	0.0362	0.0153	0.0727	0.4675	0.0366	0.0152	0.0918	0.5348	0.0244	0.0216	0.1184	0.6308

J Analysis of Trajectory Termination

In MotiveMob, the motivation predictor P_M has two related responsibilities. At each generation step, it either predicts a semantic motivation for the next transition or emits `END` to terminate the trajectory. The state predictor P_S is invoked only when a non-terminal motivation is produced. In the direct ablation without P_M, a single predictor instead outputs either the next state or `END`. Thus, all configurations retain a learned termination decision, but differ in whether termination is separated from state generation through the motivation interface.

Since the timing of `END` determines the temporal span and number of generated visits, it may affect both SI and DARD. We therefore compare the trajectory lengths and termination behavior of the direct model and three variants sharing the same motivation predictor.

Table 11: Trajectory-length and termination statistics for the motivation-mediation ablation. Mean length and termination rates are aggregated over all 4,671 test days from the six evaluation splits. SI and DARD are macro averages over the six splits. The maximum rollout length is 25 states.

State generation	**`END` predictor**	**Mean length**	**Max-step rate**	**SI ↓**	**DARD ↓**
Direct, w/o P_M (IWM)	Direct predictor	7.52	1.48%	0.0282	0.1401
P_S (IWM)	P_M	8.08	2.27%	0.0205	**0.1087**
P_S (positive-only)	P_M	7.42	2.06%	**0.0176**	0.1132
Rule-based state predictor	P_M	9.06	1.76%	0.0215	0.1163
Ground truth	–	6.82	–	–	–

Table 11 shows that the improvements in SI and DARD cannot be explained solely by matching the average trajectory length. The direct model generates 7.52 states per day on average, which is closer to the ground-truth mean of 6.82 than the 8.08 states generated by the motivation-guided model with the IWM state predictor. Nevertheless, the direct model obtains substantially worse SI and DARD (0.0282 and 0.1401, respectively) than its motivation-guided counterpart (0.0205 and 0.1087). This result suggests that the semantic interface provided by P_M contributes information beyond a simple control of trajectory length.

Reaching the rollout limit is also uncommon: only 1.48–2.27% of trajectories reach the 25-state limit across the learned predictors. Consequently, differences in the number of capped trajectories are unlikely to be the main cause of the SI and DARD improvements. The same P_M paired with different state predictors still produces noticeable metric variation. In particular, the positive-only state predictor achieves the best SI, whereas the IWM predictor achieves the best DARD. We therefore interpret motivation predictor as an important contributor of the observed improvements.

K Trajectory-Level Effects of the Motivation Predictor

Figure 4 provides three examples comparing MotiveMob with and without the trained motivation predictor. The cases are selected from the same evaluation inputs used in the quantitative experiments. Both variants use the same trained state predictor, so the comparison isolates the contribution of motivation prediction. Lines connect consecutive visits and indicate visit order.



Figure 4: Trajectory comparisons with and without the trained motivation predictor. From left to right, the columns show the ground-truth trajectory, MotiveMob with the motivation predictor, and the corresponding ablation without it. Numbered markers indicate visit order, and labels show the visit time and POI.

Case 1: Station-centered return pattern. The ground-truth trajectory revisits *Rail Station#1* from 17:20 to 20:50. With the motivation predictor, MotiveMob generates *Rail Station#1* at 17:00, 17:50, and 20:30, retaining both the station-centered activity and its evening timing. Without the predictor, the trajectory becomes *Rail Station#1* $\rightarrow$ *Hotel#1952* $\rightarrow$ *Rail Station#1* $\rightarrow$ *Hotel#2867* $\rightarrow$ *Rail Station#1* $\rightarrow$ *Hotel#2867*, introducing repeated hotel detours.

Case 2: Dining activity consistency. The reference trajectory moves from *Japanese Restaurant#416* at 12:10 to *Sake Bar#1466* at 18:30. MotiveMob with the predictor generates *Ramen Restaurant#1972* at 12:10, preserving the dining motivation and the time of the first activity, although it produces a shorter trajectory. The ablation instead alternates between *Children's Clothing Store#20* and *Supermarket#1289* six times, shifting the trajectory from dining to retail and adding repetitive visits.

Case 3: Commute pattern preservation. The ground truth contains *Metro Station#472* followed by *Platform#360*. With the predictor, MotiveMob retains this ordered transit pair and adds one nearby visit to *Shopping Mall#1254*. Without the predictor, the sequence repeatedly moves between *Platform#360* and the distant *Community Center#281* before ending at *Apartment or Condo#86*, producing a longer and less coherent chain.

Overall, these examples show that the motivation predictor helps preserve coherent activity timing and semantics, whereas removing it more often produces unrelated POIs, repetitive transitions, and implausible activity sequences. This pattern is consistent with the ablation results, where removing P_M increases both SI and DARD across all six evaluation settings. Although some cases also exhibit fewer spatial detours with P_M, its primary contribution is improved temporal and activity-semantic coherence rather than a uniform improvement in spatial distribution.

L Motivation Predictor Sensitivity to Early-Experience Supervision

Following the early-experience training procedure Zhang et al. (2025), we investigate that whether one epoch of implicit world modeling (IWM) helps the subsequent motivation predictor.

We hold the state predictor fixed, using the IWM-trained LLM state engine in every experiment. We compare three initialization schemes for the motivation predictor: (1) direct imitation learning (IL) without IWM pretraining; (2) one epoch of IWM pretraining on a balanced mixture of observed and alternative transitions ($p_+ = 0.5$); and (3) one epoch of IWM pretraining on observed transitions only ($p_+ = 1.0$). For each initialization scheme, we subsequently train the motivation predictor for 4, 8, or 12 epochs. All other inference settings are held constant.

Figure 5 reports the four JSD metrics across the six evaluation splits. The horizontal axis indicates the motivation predictor initialization scheme, while the vertical axis indicates the number of motivation predictor training epochs. Each cell displays the corresponding raw JSD value.

The benefit of one epoch of IWM is most visible after 8 epochs of motivation predictor training. At this checkpoint, the $p_+ = 0.5$ and $p_+ = 1.0$ models outperform the epoch-matched IL baseline in 15 and 16 of the 24 test–metric comparisons, respectively. In particular, $p_+ = 0.5$ reduces the mean SI from 0.0208 to 0.0192 and the mean DARD from 0.1200 to 0.1134. However, its SD and STVD are slightly higher. Therefore, IWM initialization helps the motivation predictor learn SI and DARD more quickly, but does not improve every metric.

At epoch 12, the IWM advantage is no longer consistent. The selected IL model obtains the best overall mean rank (3.81) and the lowest mean DARD (0.1043), whereas the $p_+ = 1.0$ model obtains the lowest mean SI (0.0169). Thus, one epoch of IWM provides an intermediate-training advantage rather than a uniform improvement at the final checkpoint.

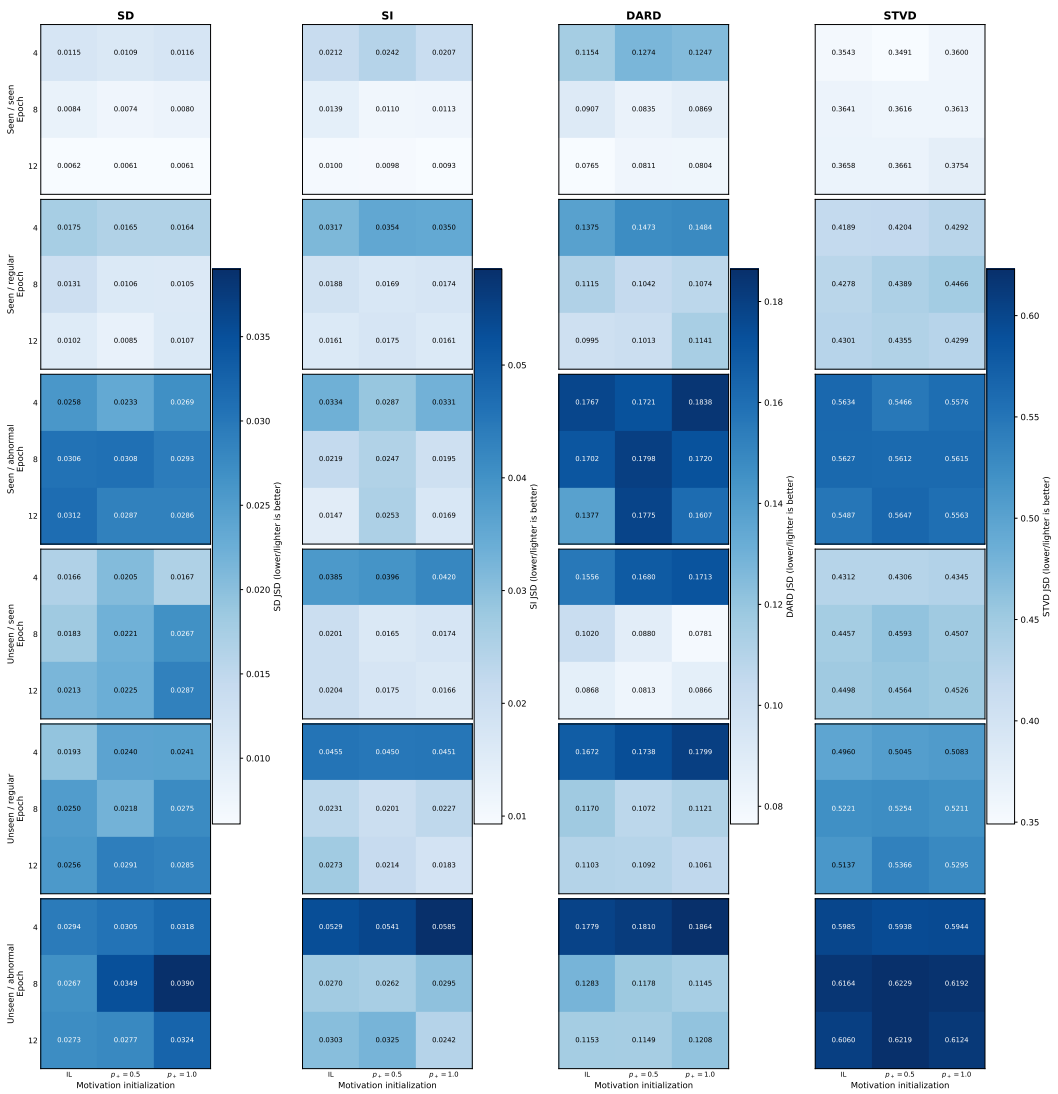

Figure 5: Motivation-predictor sensitivity to IWM initialization and training duration. Each evaluation split contains four 3×3 matrices corresponding to SD, SI, DARD, and STVD. Rows show results after 4, 8, and 12 epochs of motivation-predictor training, while columns show direct IL, balanced IWM initialization ($p_+ = 0.5$), and observed-only IWM initialization ($p_+ = 1.0$). Cell annotations report raw JSD values; lower values and lighter shading indicate better distributional alignment.

Table 12: Mean JSD over the six evaluation splits. All metrics are lower-is-better. Mean rank is calculated by ranking the nine configurations separately for every test split and metric and then averaging the resulting 24 ranks.

Initialization	Epoch	SD	SI	DARD	STVD	Mean rank
IL	4	**0.0200**	0.0372	0.1550	0.4771	5.42
IL	8	0.0204	0.0208	0.1200	0.4898	5.04
IL	12	0.0203	0.0198	**0.1043**	0.4857	**3.81**
IWM, $p_+ = 0.5$	4	0.0209	0.0378	0.1616	**0.4742**	5.42
IWM, $p_+ = 0.5$	8	0.0213	0.0192	0.1134	0.4949	4.92
IWM, $p_+ = 0.5$	12	0.0204	0.0207	0.1109	0.4969	5.02
IWM, $p_+ = 1.0$	4	0.0213	0.0391	0.1658	0.4807	6.33
IWM, $p_+ = 1.0$	8	0.0235	0.0196	0.1118	0.4934	4.92
IWM, $p_+ = 1.0$	12	0.0225	**0.0169**	0.1115	0.4927	4.13

M Inference-Time Validation and Termination Statistics

We report the operational statistics of MotiveMob's inference-time validation and termination mechanisms. Table 13 reports the intervention statistics for the speed constraint. A *step requiring resampling* is a transition for which at least one state proposal exceeds the speed limit. A step is considered *resolved by resampling* if a valid proposal is obtained within the attempt budget. Otherwise, the same-category feasibility correction is invoked. We separately report whether the correction successfully identifies a feasible substitute.

We additionally apply repetition-aware termination after the generated trajectory reaches a minimum length of 15 states. Generation terminates when appending a candidate would extend a consecutively repeating location pattern. This operation preserves the generated prefix and is equivalent to emitting `END` before the repeated candidate.

Table 13: Inference-time validation and termination statistics. All percentages are normalized by generated states, except repetition termination, which is normalized by generated trajectories.

User status	Period	Days	States	Resampling required	Resampling resolved	Correction invoked	Feasibility corrected	No feasible substitute	Repetition terminated
Seen	Seen	908	7,647	932 (12.2%)	528 (6.9%)	404 (5.3%)	231 (3.0%)	173 (2.3%)	121 (13.3%)
Seen	Regular unseen	690	5,446	686 (12.6%)	415 (7.6%)	271 (5.0%)	134 (2.5%)	137 (2.5%)	88 (12.8%)
Seen	COVID-19 unseen	392	3,218	830 (25.8%)	459 (14.3%)	371 (11.5%)	237 (7.4%)	134 (4.2%)	77 (19.6%)
Unseen	Seen	1,345	10,066	2,582 (25.7%)	1,534 (15.2%)	1,048 (10.4%)	629 (6.2%)	419 (4.2%)	207 (15.4%)
Unseen	Regular unseen	795	6,016	1,610 (26.8%)	929 (15.4%)	681 (11.3%)	432 (7.2%)	249 (4.1%)	150 (18.9%)
Unseen	COVID-19 unseen	541	4,294	1,224 (28.5%)	716 (16.7%)	508 (11.8%)	260 (6.1%)	248 (5.8%)	97 (17.9%)
All splits		**4,671**	**36,687**	**7,864 (21.4%)**	**4,581 (12.5%)**	**3,283 (8.9%)**	**1,923 (5.2%)**	**1,360 (3.7%)**	**740 (15.8%)**

N Additional Spatial Distribution Alignment Results

Figure 6 presents the four evaluation settings omitted from the main spatial-distribution analysis. All three methods broadly reproduce the major ground-truth activity centers, with relatively modest differences when users and temporal patterns are familiar. Compared with LLM4POI, MotiveMob generally forms more compact hotspots with less peripheral dispersion. MotiveMob more consistently preserves the allocation of visit mass across the main activity regions, particularly under the COVID-19 shift. Overall, these results support the main finding that MotiveMob maintains robust spatial alignment as the user or temporal setting changes.



Figure 6: Ground-truth and predicted spatial mobility distributions for four additional evaluation settings. From top to bottom, the rows show seen users in the seen period, seen users in the regular unseen period, seen users in the COVID-19 unseen period, and unseen users in the seen period. Darker red indicates higher visit density.