

UrbanLLMind: Scalable LLM-Powered Urban Mobility Simulation with Open-Weight Models

Prabin Bhandari, Sandro M. Reia, Dieter Pfoser, Antonios Anastasopoulos

George Mason University

{pbhanda2, smarti71, dpfoser, antonis}@gmu.edu

Abstract

Simulating human behavior in real-world environments is important for synthetic data generation, policy analysis, and scenario planning. Agent-based models (ABMs) are widely used for this purpose, but they often rely on hand-crafted rules and simplified assumptions that limit their realism, behavioral diversity, and adaptation to changing conditions. Large language models (LLMs) offer a promising alternative as decision engines for ABMs since they can condition agent behavior on agent state, environmental context, memory, and world knowledge. We present UrbanLLMind, a scalable generative agent-based model for urban mobility simulation. In UrbanLLMind, agents inhabit a synthetic city constructed from geographic and demographic data and act through a daily loop of LLM-powered planning, step-level mobility decisions, and end-of-day reflection. Agents also maintain a long-term memory system that allows them to recall past experiences and preserve consistent personas over time. UrbanLLMind is designed to run with open-weight LLMs on local cluster infrastructure. In our evaluations, UrbanLLMind generates plausible mobility patterns and, when powered by a fine-tuned LLM, aligns closely with observed distributions (surveys). Moreover, the model is sensitive to external behavioral drivers, such as extreme weather. We show that UrbanLLMind is capable of simulating 10,000 agents powered by an open-weight 120B-parameter model using a vLLM architecture on local cluster infrastructure. Together, these results position UrbanLLMind as the first controllable, city-grounded testbed for large-scale urban mobility simulation and scenario analysis.¹

1 Introduction

Simulating human behavior is crucial for a wide range of applications, from synthetic data generation to policy analysis and scenario planning. This is especially true in domains where collecting real-world data is costly, privacy-sensitive, and affected by noncompliance. Human behavior simulation helps researchers and decision-makers understand how individual choices shape broader population patterns. Urban mobility is an important example of this setting. Simulating how people move within a city generates synthetic mobility traces that support analysis of travel demand, congestion, and accessibility. These insights can then inform transportation planning, infrastructure investment, and policy design.

Urban mobility simulation is difficult because travel behavior depends on preferences, routines, context, and environment. These decisions are sequential: each action depends on past choices and current context. Agent-based models (ABMs; Bastariento et al., 2023), which represent individuals as autonomous agents embedded in a shared environment, are a common tool for this setting. However, ABMs often rely on handcrafted rules, fixed decision logic, or heuristics, which can limit their ability to capture real human behavior and adapt to dynamic environments (Lempert, 2002; Conte & Paolucci, 2014).

An alternative to the rule-based behavioral modeling component of ABMs is using large language models (LLMs) as decision engines for agents, known as generative agent-based

¹Code available at: <https://github.com/prabin525/UrbanLLMind>

models (GABMS; Lu et al., 2024; Adornetto et al., 2025). In such systems, the LLM powers the agent’s decision engine, generating actions based on the agent’s current state and context. Encouragingly, LLMs have demonstrated the ability to generate realistic travel diaries from contextual prompts (Bhandari et al., 2024; Li et al., 2024), suggesting they can capture complex mobility behaviors. However, prior work operates in one-shot settings, motivating the development of urban mobility ABMs with LLM-powered agents (Reia et al., 2025).

This paper introduces UrbanLLMind, the first scalable generative agent-based model for urban mobility simulation that is grounded on real-world data. UrbanLLMind populates a synthetic city, built from infrastructure data and population information, with LLM-powered agents. Each agent has a demographic profile sampled from census (or similar) data, a fixed home location, and a work or school location consistent with their profile. Agents simulate daily mobility through an iterative decision process (generate a rough daily plan, move through the city by deciding on next activities and dwell times, and reflect on the day when it ends). To support consistent behavior, agents are equipped with long-term memory that stores decisions and retrieves relevant past experiences, allowing prior behavior to inform subsequent planning and decisions. The system is designed to run with open-weight models, served through distributed inference on local cluster infrastructure. On our local GPU cluster, this setup enables UrbanLLMind to simulate 10,000 agents with GPT-OSS-120B.

Unlike prior work, we extensively evaluate UrbanLLMind by comparing its simulations with observed mobility distributions at the pattern, trip, and activity-chain levels. We additionally evaluate agents’ responses to adverse weather and their longitudinal persona consistency across simulated days. The adapted models improve population-level transition and activity-chain agreement, while decision-stack ablations show that daily planning provides the largest improvement in aggregate mobility structure and that memory is most effective when paired with reflection to support persistent individual routines. Taken together, these findings position UrbanLLMind as a controllable and empirically grounded framework for urban mobility simulation and scenario analysis.

2 Related work

UrbanLLMind is rooted in a long line of work on synthetic urban mobility generation. Earlier research developed methods for generating synthetic daily activity-travel patterns, synthetic household travel surveys, and sequential activity diaries from survey structure, demographic information, or statistical rules (Kitamura et al., 1997; Greaves & Stopher, 2000; Janssens et al., 2004; Mohammadian et al., 2010). Later work covers richer, activity-based, and geo-social simulation (Jiang et al., 2016; Reia et al., 2026; Kim et al., 2020). Recent work has shown that LLMs can model mobility behavior more directly. *Where Would I Go Next?* (Wang et al., 2024b) and AgentMove (Feng et al., 2025) use LLMs to predict next locations from mobility history and contextual cues. However, they are prediction systems rather than simulators. Bhandari et al. (2024) move closer to our setting by using prompted LLMs to generate travel-survey diaries and evaluating them at the pattern, trip, and activity-chain levels, making it the clearest predecessor to our work. LLMob (Wang et al., 2024a) extends this direction to personal mobility generation with an explicit agentic framing, but remains centered on individual trajectory generation and uses closed-source LLMs served through APIs. *Be More Real* (Li et al., 2024) similarly strengthens the profile-conditioned diary-generation line, but still focuses on travel diary synthesis rather than persistent agents embedded in a shared world. Compared to prior work, UrbanLLMind maintains the mobility focus and evaluation logic, but shifts from one-shot generation to a simulation loop with decisions, planning, memory, and reflection.

Trajllm (Ju et al., 2025) and GATSim (Liu et al., 2026) are examples of *generative* agent-based simulation: they explicitly frame urban mobility simulation around generative agents with cognitive structures, memory, and reflections. However, their evaluations remain rather qualitative, limited to small settings with fewer than 100 agents, and do not compare the resulting simulations with real-world data; instead, they focus on anecdotal examples from the agents’ behavior. Song et al. (2025) move towards large-scale simulation but do not describe the deployed LLM or serving setup, and also lack evaluation against real-world

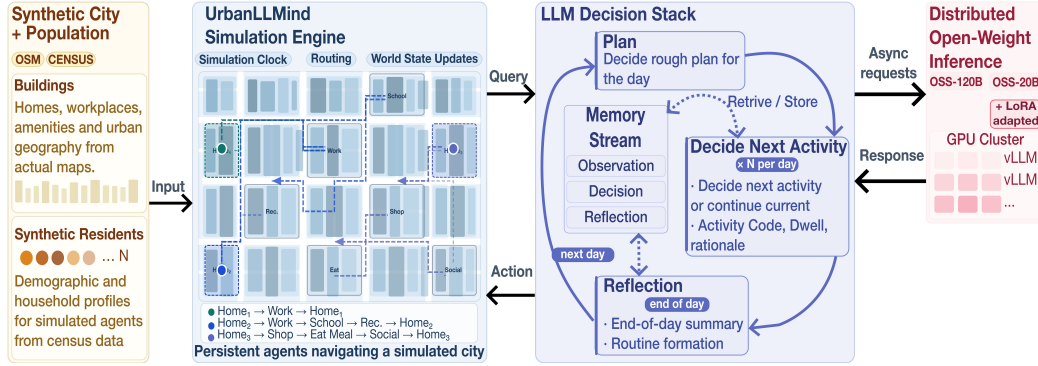


Figure 1: Overview of UrbanLLMind. A synthetic city and population are constructed from geographic and demographic data, then simulated through a world engine in which persistent agents move, dwell, and update their state over time. Agent planning and decision-making use an LLM decision stack that issues batched, asynchronous requests to a pool of open-weight models, served through multiple vLLM servers.

data. MobiVerse (Liu et al., 2025) pushes strongly on scale through a hybrid design in which a lightweight domain-specific generator produces base activity chains and LLMs modify them under changing conditions, whereas UrbanLLMind uses the LLM as the primary decision-making core within the agent loop. Further, MobiVerse relies on closed-source LLMs served through APIs. More broadly, UrbanLLMind and related systems draw inspiration from *Generative Agents* (Park et al., 2023), which introduced the planning-memory-reflection paradigm for persistent agents in shared worlds and inspired many subsequent LLM-agent simulation systems (Lu et al., 2024; Reia et al., 2025). OpenCity (Yan et al., 2024) focuses on reducing token usage and maximizing throughput for large-scale simulations built around closed-source LLMs served through APIs. AgentSociety (Piao et al., 2025) is a broader social simulation system that emphasizes multi-agent communication and simulation efficiency over urban mobility realism. Unlike these broader generative-agent systems, UrbanLLMind is tightly centered on realistic urban mobility simulation, controlled intervention response, and city-grounded spatial structure. It also explicitly centers open-weight local deployment.

3 The UrbanLLMind framework

UrbanLLMind is a generative agent-based model (GABM) for urban mobility simulation grounded in real-world city infrastructure and population data. It combines infrastructure information, a synthetic population, a shared multi-agent simulation engine, an LLM-based decision-making stack, and distributed open-weight model inference into a unified end-to-end system. This section describes these components in order, and the unified system is depicted in Figure 1, which summarizes the overall architecture and serves as a reference for the following detailed descriptions.

Synthetic city and population UrbanLLMind builds a virtual urban area from population and infrastructure data. We first define the area of interest, here San Francisco County, and use its shapefile (U.S. Census Bureau, 2023) to extract building footprints from OpenStreetMap with OSMnx (Boeing, 2017). Using footprint metadata and the approach of F. de Arruda et al. (2024), we classify buildings into nine categories: residential, workplaces, community, restaurants, schools, health care, shopping, recreation, and other. These categories define the available activity locations for agents, with social visits assigned to residential locations.

The synthetic population is generated from Census-tract-level data (U. S. Census Bureau, 2026). Following Reia et al. (2026), we use Census tract population data to allocate agents to residential buildings within each tract in proportion to building footprint area. Additional demographic data define each agent’s profile, including gender, age, employment status,

household composition, household size, income band, and vehicle ownership. These attributes determine the agent types, such as workers, students, and homemakers (including retired and unemployed agents), and guide the LLM-generated personas and behaviors.

UrbanLLMind simulation engine. The simulation engine of UrbanLLMind is implemented in Repast4Py (Collier & Ozik, 2023), which enables parallel execution across multiple processing cores. At each simulation tick, corresponding to 5 minutes of real time, agents decide whether and where to move. As they move across the city, agents may also transition between processing cores according to the simulation’s spatial decomposition.

UrbanLLMind instantiates the infrastructure, defining the locations agents can visit, and initializes the synthetic population at the first time step. At runtime, each agent alternates between staying at a location, traveling between locations, and awaiting outputs from the LLM decision stack, which returns the next activity and dwell time.

LLM outputs from the decision stack are treated as structured intents rather than free-form commands. Once the decision stack proposes a next activity and intended dwell time, the engine validates the activity type, grounds it to an eligible destination using the agent’s anchors and candidate buildings, converts the stay duration into simulation ticks, and executes the resulting travel-and-stay process. The engine also determines when calls to the LLM decision stack are needed. Day planning is triggered at the start of a new day, action selection is triggered when a dwell period ends, and reflection is triggered at the transition to a new day. By restricting LLM calls to these event boundaries and gathering them asynchronously, the engine enables scalable multi-day simulation across agents.

LLM decision stack. UrbanLLMind adopts an LLM decision stack as the cognitive policy layer for each simulated agent. This follows the planning-memory-reflection paradigm of *Generative Agents* (Park et al., 2023). The stack operates on a simulator-maintained agent state in combination with a persistent memory stream. The stack consists of three stages. The day planning and step-level mobility decisions share a common system instruction that situates the LLM as the agent’s cognitive engine (Figure A1). The reflection stage uses instructions tailored for extracting routine-level patterns from memory (Figure A4).

For every agent u and day i , and within-day decision index j , let x_j denote the current simulator-maintained context at that point, $h_{<j}$ the same-day activity history up to decision j , $\mathcal{R}_{<i}$ the retrieved reflection memories from previous days, and $\mathcal{M}_j$ the retrieved memories relevant to the current decision context. Note that all decisions are local to each agent (no agent-agent interactions). The three stages of the decision stack are as follows.

The first stage, **day planning**, is triggered at the start of each new simulated day. Conditioned on the agent’s start-of-day context x_0 and prior reflection memories (example prompt in Fig. A2), it produces a narrative plan p_i that serves as soft guidance for later decisions: $p_i = f_{\text{plan}}(x_0, \mathcal{R}_{<i})$. The resulting plan p_i is not a rigid schedule, but it solely provides day-level intent that is carried forward into subsequent step-level decisions.

The second stage, **step-level mobility decisions**, is triggered whenever an agent’s current dwell period ends. At each decision boundary, the model uses the agent’s state x_j , activity history so far $h_{<j}$, day plan p_i , and retrieved memories $\mathcal{M}_j$ to choose the next action (example prompt in Fig. A3): $(r_j, a_j, \tau_j) = f_{\text{decide}}(x_j, h_{<j}, p_i, \mathcal{M}_j)$. Here, r_j denotes the rationale, a_j the selected activity type, and τ_j the intended dwell time in minutes. This output is passed to the simulation engine for execution, and the resulting action and rationale are written back to memory as a decision record.

The final stage, **daily reflection**, is triggered at the transition to a new simulated day after the current day has concluded. Here, the model uses the day’s non-reflection memories (the list of simulated states x_j , corresponding actions a_j , and associated rationales r_j) together with a small number of prior reflections $\mathcal{R}_{<i}$ to summarize routine-level structure (example prompt in Fig. A5): $q_i = f_{\text{reflect}}(\mathcal{H}_i, \mathcal{R}_{<i})$. Here, $\mathcal{H}_i$ denotes the non-reflection memories from the day i , and q_i denotes the resulting reflection summary. This reflection is written back to

memory, to be potentially included in $\mathcal{R}_{>i}$ and inform future day plans and subsequent step-level decisions.

The **memory stream** stores three types of records. These include *observations* (from simulator state snapshots), *decisions* (from actions and rationales), and *reflections* (from the daily reflection stage). Each memory is timestamped, typed, tagged, and assigned an importance weight, making the memory stream a structured longitudinal record rather than a raw interaction history. For a candidate memory m and current focus context c , retrieval ranks memories using a weighted score

$$\text{score}(m \mid c) = \lambda_{\text{rec}} \text{Recency}(m) + \lambda_{\text{imp}} \text{Importance}(m) + \lambda_{\text{rel}} \text{Relevance}(m, c),$$

where recency, stored importance, and contextual relevance are combined to determine which memories are retrieved (implementation details given in Appendix B). In our experiments, we use $\lambda_{\text{rec}} = 1.5$, $\lambda_{\text{imp}} = 1.0$, and $\lambda_{\text{rel}} = 1.2$. These values were chosen through iterative manual inspections, where they yielded the most stable and behaviorally plausible agent behavior. This design keeps routine summaries, recent context, and earlier choices accessible throughout planning, decision-making, and reflection stages.

Distributed open-weight inference. UrbanLLMind leverages open-weight reasoning models including GPT-OSS-120B and GPT-OSS-20B, served via OpenAI-compatible vLLM endpoints (Kwon et al., 2023). At each simulation tick, agents first advance through physical execution inside the ABM. Only those that reach a planning, step-level decision, or reflection boundary require LLM calls. Inference is performed only for the subset of agents that require it at that particular step. The requests accumulate at the end of a tick and are dispatched asynchronously in batches. The serving layer schedules and batches requests across the available vLLM endpoints, providing a scalable backend for the LLM decision stack.

The simulator maintains one asynchronous client per inference endpoint and assigns agents to endpoints in a round-robin fashion during initialization. During each tick, reflections are processed first, then day plans for agents entering a new day, and finally mobility decisions for agents whose dwell times have expired. This order maintains the daily cycle of reflection, planning, and action at time boundaries. Taken together, selective event-triggered querying, end-of-tick batching, asynchronous dispatch, and distributed serving across multiple vLLM endpoints make UrbanLLMind scalable to large agent populations. On our local GPU cluster, this design supports simulations with 10,000 agents using a GPT-OSS-120B model across eight vLLM endpoints, each backed by four A100 80GB GPUs, and allows for substantially larger populations when serving 20B models. In our experiments, we report detailed behavioral results at 1K-agent scale and treat this larger-population execution as an infrastructure measurement. Further, this same architecture can host both base and adapted open-weight models, which motivates the adaptation path described next.

Synthetic supervision and reasoning-aligned model adaptation. We can adapt our LLMs to ensure that UrbanLLMind better captures real-world mobility patterns. Since no existing supervised dataset directly matches the planner, step-level decision, and reflection format required by UrbanLLMind’s LLM decision stack, we generate a synthetic dataset using a stronger frontier-reasoning model. To align with real-world data, we ground this dataset in observed mobility data.

For **replay-based supervision**, we generate a synthetic dataset by converting observed human mobility into training examples that match the reasoning loop used by the decision stack. We use person-day trip diaries from the 2017 National Household Travel Survey (NHTS; Westat, 2017). Since these records are organized as trip chains, we first reconstruct each sampled day into a sequence of locations and stays. We then replay them using the same runtime stages as UrbanLLMind. A stronger frontier reasoning model generates supervised assistant turns for each replay state. Using Gemini 3.1 Pro (Google DeepMind, 2026), our implementation wraps the runtime prompts with an outer supervision layer prompt that specifies the training task, required output format, and hidden behavioral constraints. The planner is guided by a gold-day outline. Each sampled day yields one planner example, one decision example per dwell block, and one reflection example. The full outer supervision prompts used in this step are shown in Figures A6 and A7.

Since GPT-OSS-20B generates thinking traces before the final answer, we use both outputs for supervision. Each assistant turn therefore contains a visible content field and a hidden thinking field that records a short first-person reasoning trace. This design helps align the adapted model with the required planner, decision, and reflection outputs. It also promotes a first-person-centered reasoning style that keeps outputs internally coherent over time.

The **synthetic dataset is used to adapt the GPT-OSS-20B model** to the urban mobility setting without changing the agent loop or deployment setting of UrbanLLMind. We train a parameter-efficient LoRA adapter (Hu et al., 2021) using NeMo AutoModel (NVIDIA, 2025–2026) over the base model. After adaptation, the LoRA-enhanced GPT-OSS-20B model is served through the same distributed inference stack described in Section 3.

NHTS consequently has two roles in our study: its person-day records ground the synthetic supervision examples, and its aggregate distributions provide the comparison target for our experiments in Section 4. This is not an exact diary-reconstruction or held-out person-day prediction task. Adaptation is intended to teach associations among demographics, household context, constraints, and mobility decisions in the simulator’s output format; the evidence we report is improved aggregate distributional agreement, not reproduction of unseen individual diaries.

4 Experiments

This section evaluates UrbanLLMind in a synthetic San Francisco setting with base and adapted open-weight LLMs in its decision stack. We first present the common experimental setup before evaluating the framework along four complementary dimensions: its ability to produce real-world-like mobility distributions; the plausibility of agent responses to external conditions; longitudinal persona consistency across simulated days; and the contribution of planning, memory, and reflection quantified through decision-stack ablations.

4.1 Experimental setup

We simulate a synthetic San Francisco city with 1,000 agents over seven days using four models: two prompt-only models, GPT-OSS-120B (**OSS-120B**) and GPT-OSS-20B (**OSS-20B**), and two LoRA-adapted GPT-OSS-20B variants trained on 3,950 San Francisco NHTS-derived samples (**OSS-20B-FT-SF**) and 4,000 samples from the national pool (**OSS-20B-FT-US**), respectively. Both adapters are trained for five epochs on one H100 GPU with rank 8, scaling factor 16, AdamW at 10^{-4} , and global batch size 8.

For validation against observed mobility, we follow the pattern-, trip-, and activity-chain-level evaluation framework of Bhandari et al. (2024) and compare the generated traces against San Francisco NHTS distributions. We also include the Patterns of Life (POL) simulation (Kim et al., 2019; 2020) as a conventional ABM baseline. To evaluate responsiveness to changing conditions, we repeat the simulations with a storm intervention on two days and compare behavior with the surrounding non-storm days.

Finally, we evaluate longitudinal persona consistency from agents’ weekday activity histories and examine the contribution of individual decision-stack components through four OSS-120B configurations: Reactive Only, Planner Only, Planner + Memory, and the Full Stack.

4.2 Validation against observed mobility distributions

We validate the mobility traces generated by UrbanLLMind against San Francisco travel diaries from the 2017 NHTS. Following Bhandari et al. (2024), we compare the simulated outputs at three levels of granularity: pattern level, trip level, and activity-chain level. Table 1 summarizes the results across these levels.

At the *pattern level*, we compare broad aggregate patterns of daily mobility, including the average number of visited locations and mean travel time. All LLM-based models are substantially closer to the observed average number of locations than the POL baseline.

MODEL	PATTERN				TRIP	ACTIVITY-CHAIN			
	Avg visited loc.	(Δ)	Travel hrs.	(Δ)	Frobenius norm ($\downarrow$)	Prec. _{loc}	Prec. _{all}	Recall	Overlap
Actual	5.37		1.74						
POL	4.26	(-1.11)	N/A	N/A	0.2208	84.98	96.05	36.60	26.57
OSS-120B	5.22	(-0.15)	1.14	(-0.60)	0.1471	77.20	93.70	46.34	23.28
OSS-20B	5.66	(0.28)	1.28	(-0.46)	0.1791	67.70	86.70	46.21	28.23
OSS-20B-FT-SF	5.21	(-0.16)	1.14	(-0.60)	0.0843	66.60	82.70	52.92	43.74
OSS-20B-FT-US	4.86	(-0.51)	1.02	(-0.72)	0.0776	68.60	84.40	54.68	44.18

Table 1: Validation against observed mobility at the pattern, trip, and activity-chain levels. Bold values indicate the best simulated result for each metric.

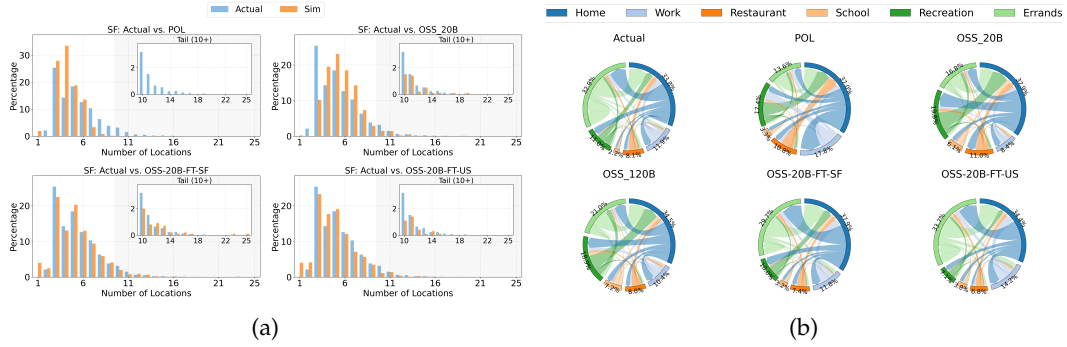


Figure 2: Adapted open-weight models better match observed mobility patterns: (a) LLM-based models recover more of the right tail (gray shaded area in histogram shown enlarged in the inset) in visited-location counts than the Patterns of Life baseline, and (b) the adapted GPT-OSS-20B models more closely match observed activity-transition structure.

Figure 2a shows that this difference is not only an average effect: the LLM-based models recover more of the right tail (gray shaded area in histograms showing higher number of visited locations) of the distribution, while the POL baseline remains concentrated around a narrower set of common patterns. This shows that LLMs can capture the rare patterns that the classical ABMs fail to capture. Mean travel time remains lower than observed across all LLM variants, likely because the simulator’s straight-line travel calculation omits real-world routing and travel frictions. Incorporating network-based travel-time estimation is therefore an important direction for future work.

At the *trip level*, we compare the transition probability matrices between activity types using the Frobenius norm of their difference, where lower values indicate a closer match. The base LLMs improve over POL, reducing the norm from 0.2208 to 0.1471 for OSS-120B and 0.1791 for OSS-20B. The adapted models improve further, reaching 0.0843 for OSS-20B-FT-SF and 0.0776 for OSS-20B-FT-US. Figure 2b illustrates this closer agreement between observed and simulated activity-transition flows.

At the *activity-chain level*, we compare full-day trip chains. Following Bhandari et al. (2024), precision measures the proportion of generated unique chains observed in the local San Francisco or broader national NHTS data, recall measures the proportion of San Francisco chains recovered by the simulation, and overlap additionally accounts for their relative frequencies. POL achieves the highest precision but substantially lower recall and overlap, indicating that it reproduces a narrow subset of common chains while missing many of the observed chain diversity. The LLM-based models recover a broader variety of observed chains, with the adapted variants achieving the highest recall and overlap and OSS-20B-FT-US performing best overall on these measures.

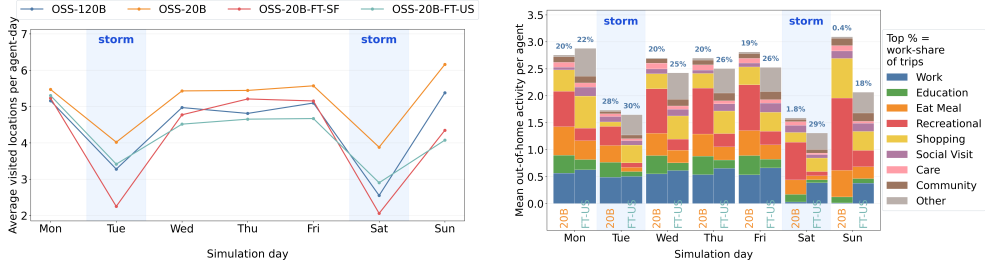


Figure 3: Storm-conditioned agents reduce travel and shift toward more necessary activity patterns. Left: all four models show clear drops in the average number of visited locations on the storm days. Right: representative base and adapted models show a shift away from optional out-of-home activities under the storm intervention.

Together, these results show that the base LLMs already improve over the conventional ABM baseline on several mobility measures. Adaptation provides additional gains in activity-transition structure and full-day activity-chain coverage.

4.3 Scenario intervention and behavioral plausibility

A key advantage of LLM-based agents over traditional ABMs is that they can condition their decisions on new environmental contexts. To test this capability, we rerun the same one-week simulation with a storm intervention appended to the prompts for Tuesday and Saturday: “A severe storm has been forecast for today. Travel across the city is expected to be heavily disrupted. The city emergency management department has advised residents to limit travel as much as possible.”

We observe a clear shift in the agents’ travel behavior in response to the storm. Figure 3 (left) shows that the average number of visited locations per agent drops sharply on both Tuesday and Saturday across all four models. We also observe a change in the agents’ activity composition under storm conditions. Figure 3 (right) illustrates this effect for two representative models, OSS-20B and OSS-20B-FT-US, with the percentage at the top of each bar showing the work-related trip share. On stormy days, the ratio of work-related trips increases relative to adjacent non-storm days, e.g., on Tuesday for OSS-20B, the ratio is 28%. In contrast, on Monday and Wednesday, it is only 20%. This focus on mandatory travel is another sign of behavioral plausibility.

The agents’ plans provide complementary evidence that these shifts result from conditioning on the intervention. As shown in Appendix Table A2, one agent retains a school drop-off obligation before returning home to work remotely, while another reframes an optional Saturday schedule as a stay-at-home day. In both cases, the response to the storm remains grounded in the agent’s profile, responsibilities, and day-specific context.

Together, the population-level changes and representative planning traces show that UrbanLLMind agents adapt their mobility behavior to an external disruption without requiring predefined disruption-specific behavioral rules.

4.4 Persona consistency and memory

UrbanLLMind also supports persistent agent behavior across multiple days through long-term memory and reflection. We evaluate whether agents maintain stable but non-rigid routines that remain consistent with their roles and individual histories.

We use two role-based metrics: role-anchor adherence measures whether workers and students maintain substantial Work or Education commitments, while anchor-start stability measures how consistently those activities begin across weekdays. We additionally measure weekday continuity among each agent’s activity chains and compare it with a role- and weekday-matched control. A positive matched-control delta indicates that an agent’s

Model	W anc.↑	S anc.↑	W start↓	S start↓	Cont.↑	Δ_{ctrl} ↑
OSS-120B	1.000	.980	22.2	33.5	.743	.126
OSS-20B	.998	.968	24.5	26.8	.654	.086
OSS-20B-FT-SF	.785	.702	256.1	210.7	.493	.054
OSS-20B-FT-US	.853	.748	215.2	190.2	.505	.058

Table 2: Longitudinal persona consistency across models. OSS-120B demonstrates the strongest overall persona consistency. W and S denote workers and students, respectively; *anc.* denotes role-anchor adherence, *start* denotes anchor start stability, *Cont.* denotes weekday continuity, and Δ_{ctrl} denotes the matched-control delta.

routine is more similar to its own previous routines than to those of other agents with the same broad role. Formal definitions are provided in Appendix C.

Table 2 reports the results for the four evaluation metrics. OSS-120B has the strongest overall longitudinal consistency, with near-complete role-anchor adherence, the highest weekday continuity, and the largest matched-control delta. OSS-20B performs second best overall, with similarly high role-anchor adherence and relatively stable Work and Education start times. Moreover, the matched-control delta is positive for every model, indicating that an agent’s routine is consistently more similar to its own routines on other days than to those of different agents with the same role.

In contrast, the adapted 20B models exhibit substantially lower role-anchor adherence, greater variation in anchor start times, and lower weekday continuity. These results reveal a tradeoff between population-level fit and longitudinal consistency: although the adapted models achieve stronger aggregate transition and activity-chain agreement, they are less effective at maintaining stable individual routines across multiple days. One possible explanation is that these models were fine-tuned using single-day trajectories but evaluated over seven-day simulations. Consequently, the fine-tuning objective may improve local, day-level behavior without explicitly encouraging consistency across longer temporal horizons. Future work could address this limitation by training within a multi-day reinforcement-learning environment and introducing objectives that reward persona-consistent behavior while penalizing unjustified deviations from established routines.

The qualitative example in Appendix Table A3 illustrates this distinction between consistency and rigidity. The agent establishes a childcare-and-work routine on Monday, recognizes it as a weekday pattern by Wednesday, and appropriately changes the routine on Saturday when work and daycare constraints no longer apply. The routine therefore remains persona-consistent while adapting to the day-specific context.

Overall, the results show that UrbanLLMind maintains coherent, agent-specific behavior across days. The following ablation examines how planning, episodic memory, and reflection contribute to this behavior.

4.5 Decision-stack ablation

We ablate the components of UrbanLLMind’s decision stack using OSS-120B to determine how planning, episodic memory, and reflection contribute to population-level mobility realism and longitudinal persona consistency. All four configurations use the same population of 1,000 agents, seven-day simulation horizon, activity taxonomy, and simulator settings. *Reactive Only* disables day planning and all retrieved memory, *Planner Only* adds day planning; *Planner + Memory* additionally retrieves episodic observations and decisions but excludes reflection summaries, and the *Full Stack* combines day planning, episodic retrieval, and reflection summaries.

Table 3 shows that day planning provides the largest improvement in population-level mobility structure. Relative to *Reactive Only*, *Planner Only* reduces the error in average visited locations from -1.045 to $+0.180$ locations per day and lowers the activity-transition distance from 0.250 to 0.153 . This indicates that explicit day planning gives agents coherent day-level structure beyond a sequence of locally plausible decisions.

Configuration	Loc./day	Loc. Δ	Trans. L2 $\downarrow$	Cont. $\uparrow$	$\Delta_{\text{ctrl}} \uparrow$
Reactive Only	4.328	-1.045	.250	.708	.026
Planner Only	5.553	+0.180	.153	.614	.039
Planner + Memory	5.357	-0.016	.132	.652	.050
Full Stack	5.245	-0.128	.144	.737	.127

Table 3: Ablation of UrbanLLMind’s decision stack. Planning improves population-level mobility structure, Planner + Memory best matches aggregate observations, and the Full Stack achieves the strongest longitudinal persona consistency.

Adding episodic retrieval further improves aggregate realism. Planner + Memory nearly reproduces the observed average number of visited locations and achieves the lowest activity-transition distance. Access to previous observations and decisions therefore helps agents align their current plans with their prior experiences and established constraints.

The longitudinal metrics reveal a different pattern. Although Reactive Only obtains a relatively high weekday continuity score, its matched-control delta is only 0.026. This suggests that its apparent regularity largely reflects repetitive or generic behavior shared across agents rather than persistent, agent-specific routines. In contrast, the Full Stack achieves both the highest weekday continuity, at 0.737, and the largest matched-control delta, at 0.127, while retaining a strong aggregate mobility fit. The substantial increase in matched-control delta relative to Planner + Memory indicates that reflection summaries help consolidate episodic experiences into stable routines that remain specific to each agent.

Overall, the ablation shows that planning structures population-level mobility, episodic memory improves aggregate alignment, and reflection strengthens persistent, agent-specific behavior across days.

5 Discussion and future work

Larooij & Törnberg (2026) reviewed 35 studies of LLM-powered agent-based models for generative social simulation and identified *validation* as the field’s central unresolved challenge. They found that many generative ABMs still rely on plausibility- or face-validity checks rather than validation against external data. Our work addresses this directly, combining external validation with empirical grounding: we adapt our base LLMs using real mobility survey data, and show that UrbanLLMind produces simulations that, on multiple levels, are similar to real-world observations of mobility distributions from NHTS travel diaries. Our ablations further show that daily planning provides the largest improvement in aggregate mobility structure, whereas memory is most effective when paired with reflection to support persistent individual routines. The storm intervention additionally demonstrates that agents respond plausibly and directionally to changing external conditions. Taken together, these results establish UrbanLLMind as a controllable and empirically grounded framework for urban mobility simulation and scenario analysis. Moreover, its use of open-weight reasoning models makes agent plans, decision rationales, and reflections available for inspection, providing greater visibility into the simulated decision process.

We highlight three directions for future work. First, UrbanLLMind should be evaluated across additional cities, demographic settings, languages, and longer simulation horizons. Prior work suggests that LLM performance can degrade in geographically underrepresented settings (Kim et al., 2024; Lin et al., 2025), while recent multilingual open-weight models (Sarvam Foundation Models Team, 2026; Anwar et al., 2025; Ramos et al., 2026) offer a promising foundation for extending the framework to regional languages. Second, the framework should support richer multi-agent interaction in households, workplaces, and other shared spaces, allowing agents to coordinate responsibilities and shared activities. Third, plans, rationales, reflections, and memory retrieval should be evaluated more systematically, including sensitivity to the retrieval count and the recency, importance, and relevance weights used for memory selection.

Acknowledgments

This work was supported by the National Science Foundation (Award Numbers 2127901 and 2109647). Additionally, this work was supported by resources provided by the Office of Research Computing, George Mason University and by the National Science Foundation (Award Numbers 1625039, 2018631).

Ethics Statement

UrbanLLMind is a research framework for synthetic urban mobility simulation and scenario analysis. Such systems can help study travel behavior under changing conditions but also pose risks. Because the agents are powered by large language models, the simulator may inherit hallucinations, cultural biases, and geographic blind spots, producing inaccurate or misleading behavioral patterns. Although the agents are synthetic, their plans, rationales, and reflections may be over-interpreted as accurate models of human cognition or decision-making. We partially address these risks by grounding the simulator in observed mobility distributions from NHTS travel diaries, studying behavior under controlled synthetic interventions, and using open-weight local deployment for greater transparency into outputs and reasoning traces. These measures do not remove the possibility of bias or misuse, and we do not treat UrbanLLMind as a substitute for real-world behavioral evidence or as a tool for high-stakes policy decisions without further validation. We instead present it as a controllable research testbed whose outputs require cautious interpretation.

References

- Carlo Adornetto, Adrian Mora, Kai Hu, Leticia Izquierdo Garcia, Parfait Atchade-Adelomou, Gianluigi Greco, Luis Alberto Alonso Pastor, and Kent Larson. Generative agents in agent-based modeling: Overview, validation, and emerging challenges. *IEEE Transactions on Artificial Intelligence*, 6(12):3165–3183, 2025. doi: 10.1109/TAI.2025.3566362.
- Mohamed Anwar, Abdelhakim Freihat, George Ibrahim, Mostafa Awad, Abdelrahman Atef Mohamed Ali Sadallah, Gurpreet Gosal, Gokul Ramakrishnan, Sarath Chandran, Biswajit Mishra, Rituraj Joshi, Ahmed Frikha, Etienne Goffinet, Abhishek Maiti, Ali El Filali, Sarah Al Barri, Samuijwal Ghosh, Rahul Pal, Parvez Mullah, Awantika Shukla, Sajid Siddiki, Samta Kamboj, Onkar Pandit, Sunil Sahu, Abdelrahman El Badawy, Amr Mohamed, Ahmad Chamma, Evan Dufraisie, Abdelaziz Bounhar, Dani Bouch, Hadi Abdine, Guokan Shang, Fajri Koto, Yuxia Wang, Zhuohan Xie, Ali Mekky, Rania Hossam Elmohamady Elbadry, Sarfraz Ahmad, Momina Ahsan, Omar Emad Mohamed El-Herraoui, Daniil Orel, Hasan Iqbal, Kareem Mohamed Naguib Abdelmohsen Fahmy Elzeky, Mervat Abassy, Kareem Ali, Saadeldine Eletter, Farah Atif, Nurdaulet Mukhituly, Haonan Li, Xudong Han, Aaryamonvikram Singh, Zain Quraishi, Neha Sengupta, Larry Murray, Avraham Sheinin, Joel Hestness, Natalia Vassilieva, Hector Ren, Zhengzhong Liu, Michalis Vazirgiannis, and Preslav Nakov. Jais 2: A family of Arabic-centric open large language models. Technical report, IFM, December 2025.
- Faza Fawzan Bastariant, Thomas O Hancock, Charisma Farheen Choudhury, and Ed Manley. Agent-based models in urban transportation: review, challenges, and opportunities. *European Transport Research Review*, 15(1):19, 2023.
- Prabin Bhandari, Antonios Anastasopoulos, and Dieter Pfoser. Urban mobility assessment using llms. In *Proceedings of the 32nd ACM International Conference on Advances in Geographic Information Systems, SIGSPATIAL '24*, pp. 67–79, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400711077. doi: 10.1145/3678717.3691221. URL <https://doi.org/10.1145/3678717.3691221>.
- Geoff Boeing. Osmnx: New methods for acquiring, constructing, analyzing, and visualizing complex street networks. *Computers, environment and urban systems*, 65:126–139, 2017.

- Nicholson Collier and Jonathan Ozik. Distributed agent-based simulation with repast4py. In *Proceedings of the Winter Simulation Conference, WSC '22*, pp. 192–206. IEEE Press, 2023.
- Rosaria Conte and Mario Paolucci. On agent-based modeling and computational social science. *Frontiers in psychology*, 5:668, 2014.
- Henrique F. de Arruda, Sandro M Reia, Shiyang Ruan, Kuldip S Atwal, Hamdi Kavak, Taylor Anderson, and Dieter Pfoser. An openstreetmap derived building classification dataset for the united states. *Scientific Data*, 11(1):1210, 2024.
- Jie Feng, Yuwei Du, Jie Zhao, and Yong Li. Agentmove: A large language model based agentic framework for zero-shot next location prediction, 2025. URL <https://arxiv.org/abs/2408.13986>.
- Google DeepMind. Gemini 3.1 pro - model card. <https://deepmind.google/models/model-cards/gemini-3-1-pro/>, 2026. Model card, published February 19, 2026.
- S. P. Greaves and P. R. Stopher. Creating a synthetic household travel and activity survey: Rationale and feasibility analysis. *Transportation Research Record*, 1706(1):82–91, 2000. doi: 10.3141/1706-10. URL <https://doi.org/10.3141/1706-10>.
- Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021. URL <https://arxiv.org/abs/2106.09685>.
- Davy Janssens, Geert Wetsand Tom Brijsand Koen Vanhoof, and Harry Timmermans. The simulation of activity diary data using sequential probability distributions. In *Proceedings of the ISCTSC Triennial International Conference on Transport Survey Methods*, 2004. *Proceedings of the ISCTSC Triennial International Conference on Transport Survey Methods*².
- Shan Jiang, Yingxiang Yang, Siddharth Gupta, Daniele Veneziano, Shounak Athavale, and Marta C González. The TimeGeo modeling framework for urban motility without travel surveys. *Proc Natl Acad Sci U S A*, 113(37):E5370–8, August 2016.
- Chenlu Ju, Jiaxin Liu, Shobhit Sinha, Hao Xue, and Flora Salim. Trajllm: A modular llm-enhanced agent-based framework for realistic human trajectory simulation. In *Companion Proceedings of the ACM on Web Conference 2025, WWW '25*, pp. 2847–2850, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400713316. doi: 10.1145/3701716.3715201. URL <https://doi.org/10.1145/3701716.3715201>.
- Joon-Seok Kim, Hamdi Kavak, Umar Manzoor, Andrew Crooks, Dieter Pfoser, Carola Wenk, and Andreas Züfle. Simulating urban patterns of life: A geo-social data generation framework. In *Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, SIGSPATIAL '19*, pp. 576–579, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450369091. doi: 10.1145/3347146.3359106. URL <https://doi.org/10.1145/3347146.3359106>.
- Joon-Seok Kim, Hyunjee Jin, Hamdi Kavak, Ovi Chris Rouly, Andrew Crooks, Dieter Pfoser, Carola Wenk, and Andreas Züfle. Location-based social network data generation based on patterns of life. In *2020 21st IEEE International Conference on Mobile Data Management (MDM)*, pp. 158–167, 2020. doi: 10.1109/MDM48529.2020.00038.
- Junghwan Kim, Jinhyung Lee, Kee Moon Jang, and Imini Lourentzou. Exploring the limitations in how chatgpt introduces environmental justice issues in the united states: A case study of 3,108 counties. *Telematics and Informatics*, 86:102085, 2024. ISSN 0736-5853. doi: <https://doi.org/10.1016/j.tele.2023.102085>. URL <https://www.sciencedirect.com/science/article/pii/S0736585323001491>.
- Ryuichi Kitamura, Cynthia Chen, and Ram M Pendyala. Generation of synthetic daily activity-travel patterns. *Transportation research record*, 1607(1):154–162, 1997.

- Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.
- Maik Larooij and Petter Törnberg. Validation is the central challenge for generative social simulation: a critical review of LLMs in agent-based modeling. *Artif. Intell. Rev.*, 59(1):15, 2026.
- Robert Lempert. Agent-based modeling as organizational and public policy simulators. *Proceedings of the National Academy of Sciences*, 99(suppl_3):7195–7196, 2002. doi: 10.1073/pnas.072079399. URL <https://www.pnas.org/doi/abs/10.1073/pnas.072079399>.
- Xuchuan Li, Fei Huang, Jianrong Lv, Zhixiong Xiao, Guolong Li, and Yang Yue. Be more real: Travel diary generation using llm agents and individual profiles, 2024. URL <https://arxiv.org/abs/2407.18932>.
- Yue Lin, Crystal J. Bae, and Emily Talen. How would ai define neighborhood boundaries? a comparison with human-crowdsourced data. *Environment and Planning B: Urban Analytics and City Science*, 0(0):23998083251369570, 2025. doi: 10.1177/23998083251369570. URL <https://doi.org/10.1177/23998083251369570>.
- Qi Liu, Can Li, and Wanjing Ma. Gatsim: Urban mobility simulation with generative agents. *Transportation Research Part C: Emerging Technologies*, 186:105576, 2026. ISSN 0968-090X. doi: <https://doi.org/10.1016/j.trc.2026.105576>. URL <https://www.sciencedirect.com/science/article/pii/S0968090X26000641>.
- Yifan Liu, Xishun Liao, Haoxuan Ma, Jonathan Liu, Rohan Jadhav, and Jiaqi Ma. Mobiverse: Scaling urban mobility simulation with hybrid lightweight domain-specific generator and large language models. *arXiv preprint arXiv:2506.21784*, 2025.
- Yikang Lu, Alberto Aleta, Chunpeng Du, Lei Shi, and Yamir Moreno. Llms and generative agent-based models for complex systems research. *Physics of Life Reviews*, 51:283–293, 2024. ISSN 1571-0645. doi: <https://doi.org/10.1016/j.plrev.2024.10.013>. URL <https://www.sciencedirect.com/science/article/pii/S1571064524001386>.
- Abolfazl (Kouros) Mohammadian, Mahmoud Javanmardi, and Yongping Zhang. Synthetic household travel survey data simulation. *Transportation Research Part C: Emerging Technologies*, 18(6):869–878, 2010. ISSN 0968-090X. doi: <https://doi.org/10.1016/j.trc.2010.02.007>. URL <https://www.sciencedirect.com/science/article/pii/S0968090X10000252>. Special issue on Transportation Simulation Advances in Air Transportation Research.
- NVIDIA. Nemo automodel: Dtensor-native spmd library for scalable and efficient training. <https://github.com/NVIDIA-NeMo/Automodel>, 2025–2026. GitHub repository.
- Joon Sung Park, Joseph O’Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology, UIST ’23, New York, NY, USA, 2023*. Association for Computing Machinery, ISBN 9798400701320. doi: 10.1145/3586183.3606763. URL <https://doi.org/10.1145/3586183.3606763>.
- Jinghua Piao, Yuwei Yan, Jun Zhang, Nian Li, Junbo Yan, Xiaochong Lan, Zhihong Lu, Zhiheng Zheng, Jing Yi Wang, Di Zhou, Chen Gao, Fengli Xu, Fang Zhang, Ke Rong, Jun Su, and Yong Li. Agentsociety: Large-scale simulation of llm-driven generative agents advances understanding of human behaviors and society, 2025. URL <https://arxiv.org/abs/2502.08691>.
- Miguel Moura Ramos, Duarte M. Alves, Hippolyte Gisserot-Boukhlef, João Alves, Pedro Henrique Martins, Patrick Fernandes, José Pombal, Nuno M. Guerreiro, Ricardo Rei, Nicolas Boizard, Amin Farajian, Mateusz Klimaszewski, José G. C. de Souza, Barry Haddow, François Yvon, Pierre Colombo, Alexandra Birch, and André F. T. Martins. Eurollm-22b: Technical report, 2026. URL <https://arxiv.org/abs/2602.05879>.

- Sandro M. Reia, Dieter Pfoser, and Henrique F. de Arruda. Opportunities and challenges of llms in urban science: Comment on “llms and generative agent-based models for complex systems research” by yikang lu et al. *Physics of Life Reviews*, 53:305–306, 2025. ISSN 1571-0645. doi: <https://doi.org/10.1016/j.plrev.2025.04.004>. URL <https://www.sciencedirect.com/science/article/pii/S1571064525000557>.
- Sandro M Reia, Henrique F de Arruda, Shiyang Ruan, Taylor Anderson, Hamdi Kavak, and Dieter Pfoser. Towards universal urban patterns-of-life simulation. *arXiv preprint arXiv:2601.22099*, 2026.
- Sarvam Foundation Models Team. Introducing sarvam’s sovereign models. <https://www.sarvam.ai/blogs/sarvam-30b-105b>, 2026. Accessed: 2026-03-03.
- Yu-Lun Song, Chung-En Tsern, Che-Cheng Wu, Yu-Ming Chang, Syuan-Bo Huang, Wei-Chu Chen, Michael Chia-Liang Lin, and Yu-Ta Lin. Incorporating llms for large-scale urban complex mobility simulation. *arXiv preprint arXiv:2505.21880*, 2025.
- U. S. Census Bureau. Index of /geo/tiger. <https://www2.census.gov/geo/tiger/>, 2026. Accessed: 2026-03-30.
- U.S. Census Bureau. 2023 cartographic boundary file, counties for United States, 1:500,000. <https://www.census.gov/geographies/mapping-files/time-series/geo/cartographic-boundary.html>, 2023.
- 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, 2024a.
- Xinglei Wang, Meng Fang, Zichao Zeng, and Tao Cheng. Where would i go next? large language models as human mobility predictors, 2024b. URL <https://arxiv.org/abs/2308.15197>.
- Westat. 2017 nhts data user guide, 2017. URL <https://nhts.ornl.gov/assets/2017UsersGuide.pdf>.
- Yuwei Yan, Qingbin Zeng, Zhiheng Zheng, Jingzhe Yuan, Jie Feng, Jun Zhang, Fengli Xu, and Yong Li. Opencity: A scalable platform to simulate urban activities with massive llm agents, 2024. URL <https://arxiv.org/abs/2410.21286>.

A Prompts

Figure A1. Shared system prompt used for the day-planning and step-level mobility decision stages of the LLM decision stack, shown here as rendered for a representative agent.

```

You are the cognitive engine (the "brain") for a simulated agent within a high-fidelity urban mobility
environment. Your goal is to generate realistic, logically consistent daily behaviors and movement
decisions.
Simulation Lifecycle:
1. Daily Planning: You start each new day by creating a rough "Anchor Plan." This is a flexible sketch of your
intended sequence, not a rigid script; real-world context and time constraints may cause you to deviate.
2. Decision Execution: At each decision point, you choose an activity and duration. The simulation handles your
travel to the destination, adds the corresponding travel time to the world clock, and initiates your stay
for the specified duration.
3. Reflection: At the end of the day, you analyze your memories to extract routine patterns that inform future
plans.
- Time Flow: The simulation advances in 5-minute increments.
- Consistency: Each decision is logged and informs your next state.
World Constraints:
- Travel friction: Switching activities requires travel time. Avoid unrealistic "teleporting" or rapid back-and-
forth switching.
- Activity Vocabulary: [Home, Work, Eat Meal, Education, Recreational, Shopping, Care, Community, Other, Social
Visit].

```

Behavioral Logic:

- Holistic Decision Influence: Every plan and decision must be fundamentally shaped by your demographic profile, temporal context (time and day of week), and the specific urban environment of your city.
- Identity-Driven Action: Act as a person of your specific age and role would.
- Workers and students should prioritize consistent blocks for their primary responsibilities during their typical active schedule.
- Homemakers should maintain realistic, home-centered daily rhythms and adapt plans to household or family needs as they arise.
- Continuity & Realism: Aim for meaningful, continuous blocks of time (e.g., 30 minutes to 8 hours depending on the activity). Specifically, recognize that nighttime rest at 'home' is a long, singular event that should span several hours.
- Frequency & Periodic Activities: Differentiate between daily staples (e.g., work, school, sleep) and periodic tasks (e.g., grocery shopping, social visits, or healthcare). Periodic tasks typically occur every few days or weeks; do not force them into every daily plan unless specifically needed.
- Cognitive Continuity: You possess a persistent memory stream that captures observations, decisions, and the specific intent behind your actions. Your memories are a narrative of your life; use past intentions to maintain social and personal consistency.

Agent Profile (Specific to this instance):

- Identity: 32-year-old female worker.
- Location: Resident of San Francisco.
- Nuances: employment status: worker; work schedule: full_time; school type: not_in_school; subtype: worker_ft; household vehicles: 4; household income: 150k to 199k; household lifecycle: two or more adults no children

Figure A2. Day-planning prompt used at the start of a new simulated day, combining simulator context with retrieved routine memories, shown here as rendered for a representative agent.

A new day has started (Monday). Provide a rough mental sketch of your plans for the day ahead.

Current State:

- Starting Location: Home

Past Routines & Insights:

- No notable memories yet.

Planning Rules:

- Use the insights above to inform your routine, but do not simply repeat them verbatim. Adapt your intentions to the specific day of the week and your identity.
- Frequency Control: Distinguish between daily "anchor" activities and periodic errands. If a past memory shows you went shopping or visited a friend, treat that as a periodic event-do not plan to do it again today unless it realistically fits your current needs.
- Focus on your major intentions and rough timing (e.g., when you'll head to work or school and anything special you have planned for the day).
- Maintain a high-level perspective; avoid rigid minute-by-minute constraints.
- Format: 2-3 concise sentences.
- Output: A short narrative paragraph.

Activity Categories:

- Available categories: Home, Work, Eat Meal, Education, Recreational, Shopping, Care, Community, Other, Social Visit

Category Notes:

- Home: Home activities, including staying at home for personal routines.
- Work: Work and work-related destinations.
- Eat Meal: Going out to eat (restaurant/cafe/food pickup destination).
- Education: Education-related destinations, including school and daycare/child-care attendance.
- Recreational: Recreation, leisure, and exercise destinations.
- Shopping: Buying goods and services.
- Care: Health care and adult-care destinations.
- Community: Volunteer, religious, and community activities.
- Other: Other/general purposes not covered by the categories above.
- Social Visit: Visiting friends or relatives (often at another residence).

Figure A3. Step-level mobility decision prompt used at a decision point, including the rough day plan, same-day activity context, and retrieved memories, shown here as rendered for a representative agent.

Current Context:

- Time: Monday, 2025-09-08 07:05
- Current Location: Home
- Today rough plan (soft guidance, created at first decision): Morning, I'll head out of Home around 7 am for a quick breakfast at a local café before commuting to work in the city- my typical 9-to-5 shift runs until 5 pm. After the day's meetings, I'll swing by a corner grocery store to restock weekly staples, then meet an old friend for a casual dinner and a little catching up at a rooftop bar- a nice way to wind down after a regular weekday. As night falls, I'll return home to unwind, set up for tomorrow, and prepare for an early sleep.

```

Reference Data:
- Recent Activity: (no activity yet)
- Relevant Memories:
  - [2025-09-08 07:05] (memory) At 07:05 on 2025-09-08 you were at Home for 425.0 minutes.

Activity Codes:
1: Home
2: Work
3: Eat Meal
4: Education
5: Recreational
6: Shopping
7: Care
8: Community
9: Other
10: Social Visit

Category Notes:
- Home: Home activities, including staying at home for personal routines.
- Work: Work and work-related destinations.
- Eat Meal: Going out to eat (restaurant/cafe/food pickup destination).
- Education: Education-related destinations, including school and daycare/child-care attendance.
- Recreational: Recreation, leisure, and exercise destinations.
- Shopping: Buying goods and services.
- Care: Health care and adult-care destinations.
- Community: Volunteer, religious, and community activities.
- Other: Other/general purposes not covered by the categories above.
- Social Visit: Visiting friends or relatives (often at another residence).

Decision Logic:
- Authenticity: Ensure your decision is a realistic expression of your demographic identity, the current time/day, and your city's environment.
- Anchor: The Daily Plan is a rough guide, not a script. Adapt if current context makes the plan impractical.
- Persistence: If continuing the current activity, use the same code and specify extension minutes.
- Realism: Stay durations must be meaningful and reflect realistic human behavior.
- Rationale: Provide a brief, human-centric reason for your decision (e.g., "Meeting my sister for coffee" rather than "Going to restaurant").

Return JSON ONLY:
{
  "rationale": "<string>",
  "next_activity_type": <int>,
  "stay_minutes": <int>
}

```

Figure A4. Reflection-specific system prompt used by the daily reflection stage of the LLM decision stack, shown here as rendered for a representative agent.

```

You are the cognitive engine (the "brain") for a simulated agent. Your task is to analyze your mobility memories and extract evidence-grounded routine patterns.

Simulation Lifecycle:
1. Daily Planning: You start each new day by creating a rough "Anchor Plan."
2. Decision Execution: You execute activities based on situational context.
3. Reflection (Current Task): Analyze memories to extract routine patterns.

Behavioral Logic:
- Holistic Influence: Patterns must be shaped by your demographic profile and city.
- Generalization Logic:
  - Use "On [Day Name]..." for patterns observed on that specific day.
  - Use "On weekdays..." ONLY if a pattern is present in at least 3 distinct weekday observations.
  - Use "On weekends..." ONLY if supported by at least 2 weekend observations.
  - If evidence is weak or limited to a single instance, use "This [Day Name]..." to indicate uncertainty rather than generalizing.

Agent Profile (Specific to this instance):
- Identity: 32-year-old female worker.
- Location: Resident of San Francisco.
- Nuances: employment status: worker; work schedule: full_time; school type: not_in_school; subtype: worker_ft; household vehicles: 4; household income: 150k to 199k; household lifecycle: two or more adults no children

```

Figure A5. Reflection prompt used at the end of a simulated day, built from same-day non-reflection memories together with up to six prior reflections for continuity, shown here as rendered for a representative agent.

```

Current Context:
- Day: Monday, 2025-09-08

```

- Time: 23:55 (Day's end)

Your day has ended. Isolate 1-2 recurring "pillars" of your routine. Do not summarize the whole day.

Memories of the day:

- [2025-09-08 07:05] At 07:05 on 2025-09-08 you were at Home for 425.0 minutes.
- [2025-09-08 07:05] Decided to go to Eat Meal and planned to stay ~30 minutes. Rationale: Have a quick breakfast at a nearby café before heading to work.
- [2025-09-08 07:50] At 07:50 on 2025-09-08 you were at Eat Meal for 35 minutes.
- [2025-09-08 07:50] Decided to go to Work and planned to stay ~480 minutes. Rationale: Finish breakfast and get ready for office time
- [2025-09-08 16:00] At 16:00 on 2025-09-08 you were at Work for 485 minutes.
- [2025-09-08 16:00] Decided to go to Work and planned to stay ~60 minutes. Rationale: Finish remaining tasks and wrap up for the day.
- [2025-09-08 17:05] At 17:05 on 2025-09-08 you were at Work for 550 minutes.
- [2025-09-08 17:05] Decided to go to Shopping and planned to stay ~30 minutes. Rationale: I'm wrapping up my 5 pm shift and need to restock groceries before heading out to a rooftop dinner with an old friend.
- [2025-09-08 17:50] At 17:50 on 2025-09-08 you were at Shopping for 35 minutes.
- [2025-09-08 17:50] Decided to go to Shopping and planned to stay ~15 minutes. Rationale: Still need to finish my brief grocery run before heading out to a rooftop dinner with my friend.
- [2025-09-08 18:10] At 18:10 on 2025-09-08 you were at Shopping for 55 minutes.
- [2025-09-08 18:10] Decided to go to Shopping and planned to stay ~10 minutes. Rationale: I need to finish a quick grocery run before heading to the rooftop dinner with my friend.
- [2025-09-08 18:25] At 18:25 on 2025-09-08 you were at Shopping for 70 minutes.
- [2025-09-08 18:25] Decided to go to Social Visit and planned to stay ~90 minutes. Rationale: Finish my quick grocery run, then head to the rooftop bar to catch up with my old friend for dinner.
- [2025-09-08 20:10] At 20:10 on 2025-09-08 you were at Social Visit for 95 minutes.
- [2025-09-08 20:10] Decided to go to Social Visit and planned to stay ~20 minutes. Rationale: Still enjoying dinner and catching up with my friend; will finish shortly before heading home.
- [2025-09-08 20:35] At 20:35 on 2025-09-08 you were at Social Visit for 120 minutes.
- [2025-09-08 20:35] Decided to go to Home and planned to stay ~60 minutes. Rationale: I'm nearly finished catching up with my friend; I'll head home now to unwind, set up for tomorrow, and begin wind-down for the night.
- [2025-09-08 21:45] At 21:45 on 2025-09-08 you were at Home for 65 minutes.
- [2025-09-08 21:45] Decided to go to Home and planned to stay ~45 minutes. Rationale: Been winding down at home after a day of work and socializing; ready to relax, prep for tomorrow, and eventually head to bed. Staying at home for about 45 minutes will give me enough time to finish chores, set my schedule, and transition to sleep.
- [2025-09-08 22:35] At 22:35 on 2025-09-08 you were at Home for 115 minutes.
- [2025-09-08 22:35] Decided to go to Home and planned to stay ~60 minutes. Rationale: I'm winding down at home, doing the quick tidy-up and setting my agenda for tomorrow. I'll stay here for about an hour before heading to bed.
- [2025-09-08 23:40] At 23:40 on 2025-09-08 you were at Home for 180 minutes.
- [2025-09-08 23:40] Decided to go to Home and planned to stay ~20 minutes. Rationale: Wrapping up the evening, tidying up and setting my agenda for tomorrow before heading to bed. Staying at home for a short wind-down is realistic at this hour.

Task: Extract 1-2 high-level routine insights based on memory evidence.

- Objective: Follow generalization rules (On weekdays/weekends/specific days) based on the frequency of the pattern in your history.
- Example: "On Mondays I generally start work at 9 AM" or "This Monday I went to recreation late."
- Output Format: Return exactly ONE compact line of text.
- Rules: No bullets, no headings, no JSON.

Figure A6. Outer system prompt used by the teacher-model layer during synthetic supervision generation.

You are creating supervised fine-tuning data for an urban mobility simulator.

Return JSON only with exactly these string keys:

- "thinking": the hidden reasoning trace for the assistant turn
- "content": the visible assistant output that the simulator would consume

The top-level JSON response must always be the outer wrapper above.
Do not return the simulator-visible decision JSON at the top level.

Rules:

- Write "thinking" as the agent's private inner monologue in first person singular.
- "thinking" must sound like a person deciding what to do, not a narrator or analyst describing a person.
- Use the supervisor-only constraints only to keep the visible answer behaviorally correct; never mention those constraints explicitly.
- You may make mild, human household or social inferences from the visible profile and memory context when they are plausible, such as a partner at home in a two-adult household or household responsibilities implied by family structure.
- Keep those inferences soft and generic. Good examples are "my partner", "someone at home", or "household responsibilities". Do not invent names, exact promises, or highly specific backstory unless it is already established in the visible plan or memories.
- If a mild household or social context is introduced earlier in the day, keep later thinking and visible rationales consistent with it.

- Produce content only for the current task. Never bundle day planner, decision, and reflection outputs together.
- Never mention any of these words or concepts in "thinking": agent, gold, target, hidden, label, NHTS, dataset, training, prompt, simulator, simulation, assistant, user, teacher, supervisor.
- Keep "thinking" concise and consistent with the inner runtime prompts.
- Keep "content" strictly compliant with the requested runtime format.
- Do not wrap the "content" string in markdown code fences.
- Do not mention supervisor-only constraints explicitly in "content".

Figure A7. Outer user prompt template used to wrap a runtime planner, decision, or reflection prompt together with task-specific guidance and hidden supervisory constraints.

```

Task type: {task_type}

Visible content requirements:
- day_planner: return a short narrative paragraph only.
- decision: return JSON only with rationale, next_activity_type, stay_minutes.
- decision: the "content" value itself must be a valid JSON object string with double-quoted keys and string values when applicable.
- decision: do not use single quotes, trailing commas, comments, or surrounding prose.
- reflection: return exactly one compact line.

Content scoping requirements:
- Return content for the current task only.
- Do not return a combined object with keys like day_planner, decision, or reflection.

Task-specific thinking guidance:
{task_specific_guidance}

Top-level response requirements:
- Always return the outer wrapper with keys "thinking" and "content".
- Never return the simulator-visible answer directly at the top level.

Inner runtime system prompt:
{runtime_system_prompt}

Inner runtime user prompt:
{runtime_user_prompt}

Supervisor-only behavioral constraints (never mention or quote these in thinking/content):
{gold_payload_json}

```

B Memory Retrieval Details

Here we define how $\text{Recency}(m)$, $\text{Importance}(m)$, and $\text{Relevance}(m, c)$ are computed in the simulator.

Let the current simulator tick be t , and let c_t denote the current focus context. The retrieval score is computed as

$$\text{score}(m \mid c_t) = \lambda_{\text{rec}} \cdot \frac{1}{\max(1, t - t_m)} + \lambda_{\text{imp}} \cdot \text{importance}(m) + \lambda_{\text{rel}} \cdot |\text{tags}(m) \cap \text{tags}(c_t)|,$$

where t_m is the tick at which memory m was written. In our experiments, we use $\lambda_{\text{rec}} = 1.5$, $\lambda_{\text{imp}} = 1.0$, and $\lambda_{\text{rel}} = 1.2$, retrieve up to 6 memories per decision context, and cap the memory store at 200 entries.

Recency. Recency is implemented as inverse age in simulator ticks. For a memory written at tick t_m and retrieved at tick t , we define $\text{age}(m, t) = \max(1, t - t_m)$ and compute

$$\text{Recency}(m) = \frac{1}{\text{age}(m, t)}.$$

Importance. Importance is stored with each memory when that memory is written, and the retrieval score reads it back directly as

$$\text{Importance}(m) = \text{importance}(m).$$

Currently, we assign fixed importance values by memory type: 0.7 for observation memories, 1.2 for decision memories, and 1.5 for reflection memories.

Relevance. Relevance is computed as tag overlap between a stored memory and the current focus context:

$$\text{Relevance}(m, c_t) = |\text{tags}(m) \cap \text{tags}(c_t)|.$$

Here, $\text{tags}(m)$ and $\text{tags}(c_t)$ denote small discrete tag sets. Observation and decision memories receive an activity-derived tag together with the day of the week, while reflection memories receive tags extracted from the generated reflection summary by keyword matching. These reflection tags always include reflection, and may additionally include activity-related tags such as work, education, shopping, or social_visit. The current focus context c_t is represented similarly, using the agent’s current activity, the current day of the week, and selected profile attributes, such as employment, school, work schedule, and household indicators.

C Longitudinal Evaluation Details

The longitudinal metrics in § 4.4 are computed from each agent’s Monday to Friday activity logs. Let $c_{i,d}$ be agent i ’s activity chain on weekday d . For two chains, we use normalized Levenshtein similarity

$$\text{sim}(c, c') = 1 - \frac{\text{Lev}(c, c')}{\max(|c|, |c'|)}.$$

Weekday continuity is the mean of this similarity over all within-agent weekday pairs. For agent i with weekday chains $c_{i,d}$,

$$\text{WRC}_i = \frac{1}{|P_i|} \sum_{(d,d') \in P_i} \text{sim}(c_{i,d}, c_{i,d'})$$

where P_i is the set of Monday-to-Friday weekday pairs available for agent i .

For the matched control, each agent-day is compared with up to 20 randomly sampled different-agent days having the same role and weekday. The agent-level control similarity is the mean over these comparisons, and the **matched-control delta** is within-agent weekday continuity minus that control.

$$\Delta_i = \text{WRC}_i - \text{ControlSim}_i$$

A positive delta therefore indicates that an agent’s own weekdays are more similar than schedules drawn from other agents with the same broad role.

For each worker or student, **role-anchor adherence** is the fraction of observed weekdays containing at least 120 minutes of Work or Education, respectively. **Anchor start stability** is the standard deviation of the first qualifying anchor start across weekdays; it is defined for agents with at least three qualifying starts and summarized by the median across agents.

Table A1 contrasts the highest- and lowest-scoring OSS-120B agents selected deterministically by weekday continuity. The high-scoring student repeats the same chain on all five weekdays (continuity 1.000) with an anchor-start standard deviation of 29.5 minutes. The low-scoring worker retains a Work anchor every weekday but changes activity order substantially (continuity 0.489) with an anchor-start standard deviation of 627.0 minutes.

D Representative Examples

Scenario intervention and behavioral plausibility. Table A2 shows two examples of how the stormy day intervention affects the planning reasoning and resulting daily plan for two agents.

Persona consistency and memory. Table A3 presents a representative example of how persistent memory and daily reflection support consistent agent behavior across days. The agent is a 41-year-old female stock clerk with a toddler in the household. On Monday, she

Day	High continuity: student 4451	Low continuity: worker 53684
Monday	Home → Education → Recreational → Home	Home → Eat Meal → Recreational → Home → Work
Tuesday	Home → Education → Recreational → Home	Work → Home → Work → Home → Recreational → Home → Work → Social Visit → Home
Wednesday	Home → Education → Recreational → Home	Home → Recreational → Eat Meal → Shopping → Home → Work → Home
Thursday	Home → Education → Recreational → Home	Home → Eat Meal → Recreational → Home → Work
Friday	Home → Education → Recreational → Home	Work → Eat Meal → Home → Recreational → Work → Eat Meal → Home

Table A1: High- and low weekday continuity OSS-120B trajectories.

	OSS-120B (Agent 8429, Tuesday)	OSS-20B-FT-SF (Agent 67985, Saturday)
Profile	34-year-old female worker (<i>lecturer</i>); children aged 6–17 in household; ...	25-year-old male homemaker; no children; 2 vehicles; ...
Planning reasoning	We need to output a short narrative paragraph, describing mental sketch of plans for Tuesday. Must adapt to storm, limit travel...	It's Saturday morning, but I need to adjust my usual weekend plans because a massive storm is forecasted to hit midday..
Daily plan	I'll start the day at home preparing breakfast and driving the kids to school before the morning storm hits , then work remotely from my home...	Given the severe storm expected later today, I've decided to take a quiet, bunker-down kind of Saturday at home . I plan to use this...

Table A2: Representative storm-day planning traces show that both base and adapted models explicitly incorporate the storm forecast into daily plans while still conditioning those plans on household obligations and day-specific context.

establishes a weekday pattern centered on childcare and work. She prepares the toddler for daycare, completes her stock-clerk shift, and picks the child up before returning home. By Wednesday, this pattern is no longer described as a one-off plan, but is reflected back as an explicit weekday routine, showing that the memory-reflection loop has abstracted repeated behavior into a stable summary. On Saturday, the work-daycare schedule disappears, but the same caregiving persona remains visible as the agent plans a park outing and grocery trip with the toddler before returning home. The behavior changes with the day context while preserving a consistent profile-grounded identity.

	Monday (Day 1)	Wednesday (Day 3)	Saturday (Day 6)
Profile	41-year-old female worker (stock clerk); toddler under 6 in household.		
Daily plan	I'll begin the morning at home getting my toddler ready for daycare before heading to my full-time stock-clerk shift at the shop-kitchen facility (roughly 8 am–5 pm). After work I'll pick the child up ...	In the morning I'll get my toddler ready, drop them off at daycare, and then head to my stock-clerk shift at the shop-kitchen location for the usual eight-hour work block with a brief lunch break...	Morning I'll start at home making breakfast and getting the toddler ready, then we'll head to the local park for a few hours of play. Mid-day I'll swing by the grocery store to restock the house...
Reflection	This Monday I began at home getting my toddler ready, dropped them off at daycare and headed to my stock-clerk shift, then in the evening I picked the toddler up from daycare...	On weekdays I drop my toddler at daycare in the early morning before my stock-clerk shift, and after work I pick the toddler up and head straight home for dinner and bedtime.	On weekdays I drop my toddler at daycare..., while this Saturday I began with a park outing with my toddler, then grocery shopped and spent the rest of the day at home...

Table A3: Memory and reflection support persona consistency across days: a representative OSS-120B agent forms a stable weekday childcare-work routine and then departs from that schedule on Saturday while preserving the same caregiving role.