# GenWorld: An LLM-Ready Urban Simulation Platform with Empirically-Grounded Synthetic Populations

Gen Li[1], Jieyuan Lan[1], Pengcheng Xu[2], Zongyuan Wu[3], Masaki Ogura[1], and Tao Feng*[1]

[1]Graduate School of Advanced Science and Engineering, Hiroshima University, Higashi-Hiroshima, Japan
[2]School of Civil Engineering, Chang'an University, Xi'an, China
[3]North China University of Water Resources and Electric Power, China

January 30, 2026

## Abstract

Large Language Models (LLMs) are increasingly used to model agent behavior in simulation, yet existing platforms lack empirically grounded, city-scale environments with building-level spatial resolution. We present **GenWorld**, an urban simulation platform that grounds agent populations in real-world census and geospatial data at building-level resolution. The platform provides a structured agent–environment interface with machine-readable decision traces, and supports offline compilation of LLM signals into decision priors for city-scale rollout. We instantiate **GenWorld** in Higashihiroshima, Japan (196,608 synthetic residents), validate demographics against census tabulations, and use YJMob100K mobile-phone data as a commuting-distance diagnostic.

**Keywords:** LLM agents, Urban simulation, Synthetic population, Building-level assignment, Multi-agent systems, Knowledge distillation, Empirical validation

## 1 Introduction

### 1.1 The Need for Realistic Urban Environments for LLM Agents

Large Language Models (LLMs) have shown strong capabilities in reasoning, planning, and decision-making, leading to growing interest in their application as autonomous agents [32, 39]. Recent works show that LLM agents can engage in complex social interactions [32], solve multi-step reasoning tasks [42], and collaborate in team environments [15]. As these agents move toward real-world use in domains such as urban planning, transportation management, and disaster response, a key question is: **how can we deploy and study LLM agents in realistic, complex environments that mirror real-world constraints?** However, despite these advances, existing platforms lack empirically grounded, city-scale environments with building-level spatial resolution and validated synthetic populations. Our goal is to provide simulation infrastructure that enables LLM-driven agent research in realistic urban settings.

Urban environments provide a demanding testbed for such research. Cities exhibit **spatial constraints** where physical distance and infrastructure topology shape feasible actions, **resource competition** where multiple agents must coordinate access to limited facilities, **heterogeneous populations** with diverse demographics and capabilities, and **emergent dynamics** where individual decisions aggregate into system-level phenomena like traffic congestion or supply chain disruptions. These characteristics make urban simulation well-suited for studying **situated intelligence**—the ability of agents to make effective decisions grounded in realistic spatial, social, and temporal contexts. Beyond agent research, high-fidelity urban models can support applications such as flood risk assessment and evacuation planning (via building-level populations and elevation), and disruption analyses using social networks and infrastructure.

Recent efforts to apply LLM agents to urban contexts fall into two categories, both with important limitations. **Abstract text-based environments** make decisions without grounded spatial contexts [28], while **POI-based urban simulations** [32, 33, 38] rely on coarse spatial aggregations (e.g.,

*Corresponding author: taofeng@hiroshima-u.ac.jp

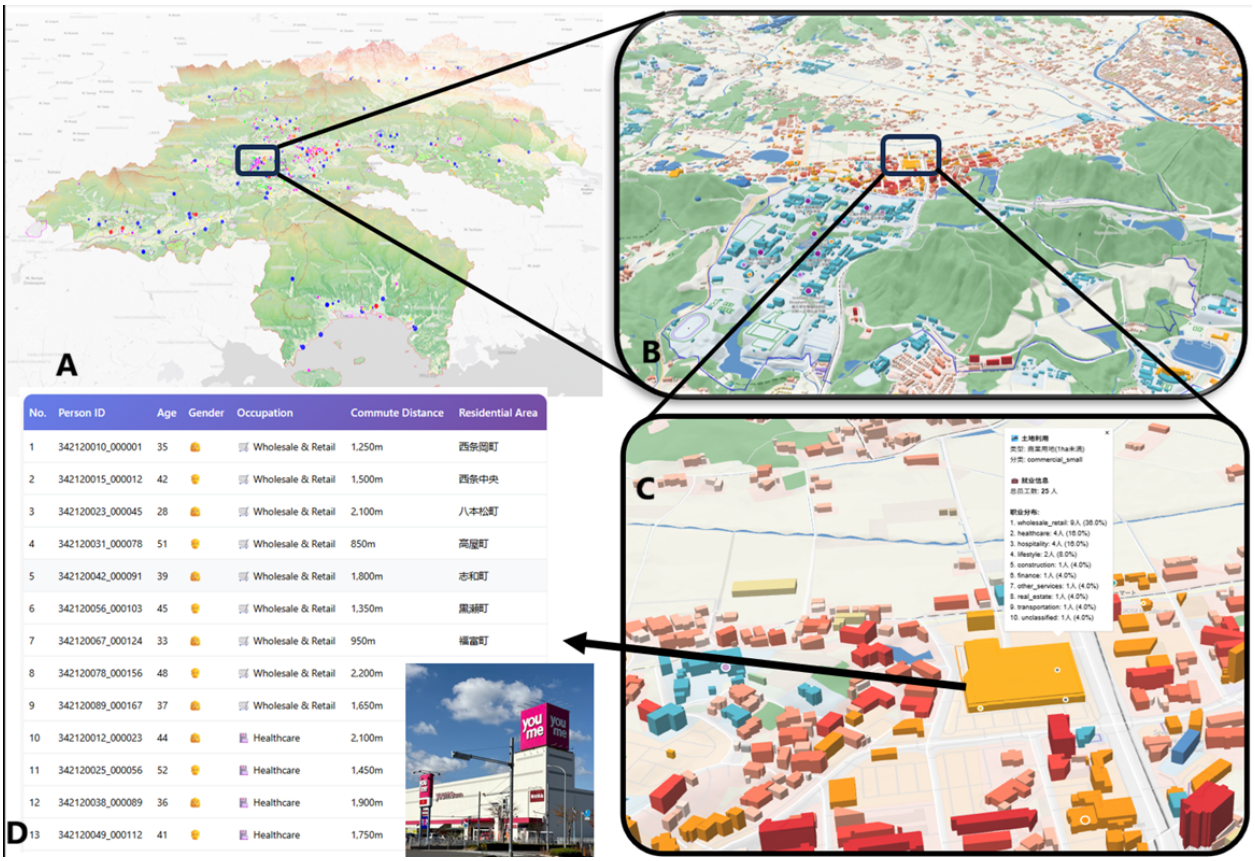



Figure 1: Multi-scale spatial granularity of **GenWorld**'s building-level population synthesis in Higashihiroshima, Hiroshima, Japan. **(A)** City-level view showing 196,608 individuals distributed across georeferenced buildings, validated against census data. **(B)** District-level view near Hiroshima University, revealing diverse building types (residential, commercial, educational) with topographic context and elevation data. **(C)** Building-level view of a Youme Town supermarket area with 47 employees spatially assigned to a corresponding commercial land-use parcel; residential buildings are rendered in red with color intensity proportional to resident counts (darker indicates more residents). **(D)** Individual-level details showing employee household origins, occupations, commuting distances, and residential neighborhoods (cho/town). This fine-grained spatial allocation supports realistic social network formation and environment-aware agent cognition, which are typically difficult to capture in TAZ-based or POI-list approaches.

TAZ-level zones) rather than building-level assignments, and often lack empirical validation against census and mobility data. As a result, current platforms may not capture the spatial and social constraints that determine what actions are feasible in real cities.

However, creating such a platform requires overcoming a fundamental challenge: **population realism**. Unlike abstract environments where agent diversity can be arbitrarily defined, realistic urban simulation demands that synthetic populations accurately reflect the demographic composition, spatial distribution, and employment patterns of real cities. Without this foundation, agent behaviors operate in an unrealistic vacuum, limiting the validity and generalizability of research findings [25].

## 1.2 Challenges in Building LLM-Ready Urban Simulation Platforms

Building a high-fidelity urban simulation platform suitable for LLM agent research raises several practical challenges:

**Challenge 1: Synthetic Population Generation and Validation** Traditional agent-based models often rely on simplified or ad-hoc population generation methods that fail to capture the full het-

erogeneity of real urban populations [47]. Common approaches either infer home locations from mobile phone data (missing non-users) or randomly generate populations within TAZ zones, along roadways, or at POI coordinates without georeferenced building assignments. Recent LLM-driven simulations [33, 38] have made progress in agent cognition but inherit these spatial limitations, operating with populations assigned to **abstract spatial units** rather than specific georeferenced buildings. This coarse spatial granularity prevents the emergence of realistic neighborhood social networks and limits agents' ability to reason about fine-grained spatial contexts (e.g., "my neighbor two buildings away"). While Iterative Proportional Fitting (IPF) and related techniques exist for population synthesis [18], existing pipelines struggle to integrate multi-source geospatial data (census, land use, building footprints), implement building-level spatial assignment algorithms that match realistic commuting patterns, and validate against ground-truth data such as census statistics and mobile phone records. This limitation is particularly problematic for LLM agents, whose reasoning and planning depend on fine-grained, relational, and language-expressible spatial contexts.

**Challenge 2: Computational Scalability of LLM-Driven Agents** LLMs are computationally expensive. A single inference call can take hundreds of milliseconds and cost significant resources. In a city-scale simulation with hundreds of thousands of agents, directly querying LLMs for each agent's decision at each time step is computationally expensive in practice. For instance, simulating 200,000 agents over a 24-hour period with 15-minute time steps would require 19.2 million LLM calls—even under optimistic latency assumptions, this is a substantial computational burden. This bottleneck has limited the use of LLM agents in large-scale, realistic simulations, and has often constrained evaluation to small-scale scenarios that may not reveal important emergent phenomena.

**Challenge 3: Integration and Accessibility for AI Researchers** Most existing urban simulation platforms were designed for domain experts in transportation or urban planning, not for AI researchers. They often require deep knowledge of urban modeling conventions and do not provide LLM-native interaction paradigms such as natural language observation spaces and JSON-structured action interfaces. This creates a barrier to entry for AI researchers seeking to deploy their agents in realistic urban contexts. We survey these platforms in Section 2.2.

## 1.3 Contributions

To address these challenges, we present **GenWorld**, an LLM-ready urban simulation platform grounded in empirical data and building-level spatial representation (Figure 1). This paper makes the following contributions:

1. **Empirically-Grounded Urban World at Building Level**: We develop a building-level population synthesis and grounding pipeline validated against multi-source data, generating 196,608 individuals with tract-level census totals enforced as hard constraints (reported for completeness) and distributional agreement on demographic variables not enforced as exact constraints (e.g., male-ratio MAE $\approx 0.016$, mean age KS $\approx 0.030$). We ground households to georeferenced buildings and assign schools and workplaces through rule-based and quota-constrained spatial allocation. We use anonymized mobile phone mobility data (YJMob100K) [40] as a commuting-distance diagnostic, with appropriate limitations due to anonymization and manual registration. On top of this population foundation, we construct an urban environment integrating POIs, roads, elevation, and building-level spatial representation, enabling situated decision-making under physical and infrastructure constraints.

2. **City-Scale LLM-Agent Simulation via Knowledge Distillation**: We develop a distillation pipeline that estimates context-dependent teacher decision distributions through repeated sampling and compiles them into efficient probabilistic policies (lookup tables) for simulation-time inference. This design shifts LLM calls out of the simulation loop and can yield large speedups in typical settings, enabling large-scale rollouts in our reference instantiation. Unlike standard policy distillation in reinforcement learning, our target is language-conditioned, context-dependent decision distributions under a fixed, executable candidate set.

3. **LLM-Ready Integration Interface**: We provide a standardized tool-based interface that bridges traditional urban simulation with modern LLM agents, exposing natural language observation spaces and structured action specifications. This LLM-native interaction layer lowers the barrier for AI researchers to deploy and study LLM agents in realistic urban settings.

### 1.4 Paper Organization

The remainder of this paper is organized as follows. Section 2 reviews related work in LLM agent simulation, urban simulation platforms, synthetic population generation, and distillation for agent simulation. Section 3 describes the LLM agent interface. Section 4 presents the distillation pipeline for city-scale simulation. Section 5 presents the empirically grounded urban world construction, including data sources, population synthesis and spatial grounding, and multi-source validation. Section 6 introduces the platform architecture and simulation engine. Section 7 presents demonstration results and scalability analysis. Finally, Sections 8 and 9 discuss limitations and conclude.

## 2 Related Work

Table 1 provides an overview of how GenWorld compares to existing platforms across three categories: LLM agent simulation platforms, LLM-based urban mobility platforms, and population synthesis platforms. We detail these comparisons in the following subsections.

### 2.1 LLM Agents and Simulation Platforms

The emergence of Large Language Models has driven rapid progress in autonomous agent systems. Recent works demonstrate LLM agents across a range of settings, from social simulation [32] to tool use [34] and multi-agent collaboration [15]. This progress motivates the need for realistic simulation environments that can support LLM agent research under real-world constraints.

**Existing Agent Platforms.** Existing platforms and benchmarks span multiple levels of realism.

**Abstract environments** (e.g., GridWorld/TextWorld-style tasks) [28] are useful for isolating reasoning and planning, but they abstract away geography, resource constraints, and social interactions.

**Task-specific platforms** such as SWE-bench [19] (software engineering) and WebArena [46] (web navigation) provide grounded objectives and measurable success criteria, but they typically focus on single-agent, non-spatial settings.

**Social simulation platforms** such as Generative Agents [32] explore emergent interactions, yet the environments are simplified and the scale (e.g., 25 agents) is insufficient for studying city-scale phenomena and computational scalability. CityBench [8] evaluates LLM world-modeling capabilities for urban tasks but does not provide building-level population grounding.

**LLM Agents in Transportation and Mobility.** Beyond interactive simulacra, LLMs have been explored as simulated economic agents [17] and integrated into mobility and transportation settings. LLMob [38] uses self-consistency and retrieval-augmented strategies for individual mobility generation with GPS-based validation. Liu et al. [27] outline an LLM-agent-based transportation modeling framework with a small proof-of-concept. TrajLLM [20] combines LLM-based persona generation with hybrid destination choice (LLM + physical models), but focuses on POI-level trajectories. GATSim [26] and MobileCity [43] target larger-scale mobility simulation; MobileCity achieves efficiency partly by disabling LLM modules at scale, trading behavioral fidelity for speed. OpenCity [41] proposes a "group-and-distill" prompt optimization strategy that clusters agents with similar attributes and distills shared reasoning patterns, achieving 600× acceleration in simulation time; however, it focuses on prompt-level efficiency rather than building-level spatial grounding. Overall, these efforts primarily emphasize individual trajectory generation or engineering efficiency. They often do not provide city-scale population synthesis with jointly validated demographics and spatial assignments (e.g., building-level placement) or thorough empirical validation.

Existing platforms often do not jointly provide realistic population foundations supported by empirical data, spatial complexity with infrastructure constraints, computational scalability to city-scale (100,000+ agents), and LLM-compatible interfaces. GenWorld provides an empirically grounded urban environment with 200,000-agent scalability based on data from Higashihiroshima, Hiroshima, Japan.

### 2.2 Urban Simulation Platforms

Agent-based modeling has a rich history in urban and transportation research [6, 23], with several established platforms:

**Traditional ABM Platforms. GAMA** [35], **MASON** [29], and **NetLogo** [37] are widely used for urban simulation. These platforms provide powerful modeling capabilities but were designed for domain experts rather than AI researchers, and they do not provide standardized LLM integration interfaces or natural language observation spaces.

**Transportation Simulation Tools.** MATSim [16], SUMO [22], and similar tools focus on traffic simulation with detailed traffic modeling. However, they

Table 1: Comparison of GenWorld with Related Platforms

| Platform | Population Realism | Empirical Validation | Scale (Agents) | Real Geography | Spatial Detail | Social Networks |
|---|---|---|---|---|---|---|
| *LLM Agent Simulation Platforms* | | | | | | |
| GridWorld/TextWorld | Low | No | $< 100$ | No | No | No |
| Generative Agents [32] | Low | No | $< 100$ | No | Limited | Limited |
| WebArena [46] | N/A | N/A | Individual | No | No | No |
| *LLM-Based Urban Mobility Platforms* | | | | | | |
| LLM-ABM Framework [27] | Low | No | $< 100$ | No | Low | No |
| LLMob [38] | Medium | GPS | Individual | Yes | POI-level | No |
| TrajLLM [20] | Medium | Qualitative | $< 100$ | No | POI-level | No |
| MobAgent [24] | Medium | Survey | Individual | Yes | POI-level | No |
| GATSim [26] | Medium | No | 1K–10K | No | Medium | Limited |
| MobileCity [43] | Medium | No | 1K–10K | No | Medium | Limited |
| OpenCity [41] | Low | GPS | 1K–10K | Yes | POI-level | No |
| *Population Synthesis Platforms* | | | | | | |
| Jiang et al. [18] | High | Census | 100K+ | Yes | Road-based | Multi-layer |
| Pseudo-PFLOW [21] | High | Census | 100K+ | Yes | Building | No |
| **GenWorld (Ours)** | **High** | **Multi-source** | **100K+** | **Yes** | **Building** | Multi-layer[†] |

[†] Social networks are generated from spatial co-location but not used in current experiments.

typically use simplified behavioral models and do not incorporate the cognitive realism enabled by LLM-driven agents.

**Commercial Platforms.** AnyLogic, Citilabs, and other commercial tools offer sophisticated urban modeling but are closed-source, expensive, and not designed for AI research integration.

Recent open-source efforts such as VoxCity [9] provide seamless 3D urban environment generation, while Biljecki and Chow [3] establish global building morphology indicators for standardized urban analysis. However, existing platforms were not designed with LLM agents in mind. GenWorld aims to address these gaps by providing natural language observation spaces, flexible action specifications, validated population foundations, and computational scalability through knowledge distillation.

## 2.3 Synthetic Population Generation

Generating realistic synthetic populations is fundamental to valid agent-based modeling [25].

**Population Synthesis Methods. Iterative Proportional Fitting (IPF)** [5] and its variants are commonly used methods, adjusting cell weights to match marginal distributions from census data. Beyond IPF, prior work also explores alternative formulations such as combinatorial optimization, Bayesian approaches, and deep generative models (DGMs). While DGMs can generate diverse populations beyond observed samples, they often struggle to balance *sampling zeros* (valid but unobserved combinations) with *structural zeros* (implausible combinations) [25]. Recent work explores LLM-based approaches: Li et al. [24] proposed MobAgent, using LLMs to extract fine-grained mobility patterns from individual profiles through self-evaluation and recursive reasoning, validated on 0.2M travel surveys. Ma et al. [30] developed a foundation model using LLMs for semantic enrichment of GPS trajectories, demonstrating transfer learning across regions (LA to Egypt) for mobility pattern synthesis. While these LLM-based methods have been explored for individual trajectory generation, they focus on personal mobility modeling rather than city-scale population synthesis with validated demographic distributions and spatial assignments.

**Spatial Assignment and Social Networks.** Assigning synthetic individuals to geographic locations is important for spatial realism. Common approaches include: **gravity models** [1] for workplace assignment, **distance-based allocation** for household placement, and **constraint satisfaction** for student-to-school assignment. Jiang et al. [18] developed a large-scale method generating 23 million geographically-explicit individuals for New York Metro Area with multi-layer social networks (household, work, school, daycare) emergent from spatial co-location, highlighting the importance of social networks for urban simulations. Kashiyama et al. [21]

developed Pseudo-PFLOW, an agent-based framework that downscales census data to building-level assignments using Markov chain models for activity generation, covering Japan's 130 million population. While achieving strong validation results ($R^2$=0.61–0.98 for population distribution), these approaches rely on traditional statistical models rather than LLM-driven behavioral realism and lack integration with modern LLM agent frameworks.

**Validation Approaches.** Traditional validation relies primarily on census data comparison. Recent work has begun incorporating **mobile phone data** [40] for validating commuting patterns, building on foundational studies of human mobility patterns [11, 31, 13]. Ma et al. [30] demonstrated multi-level validation through traffic simulation, achieving MAPE $< 6\%$ for traffic volumes. However, systematic validation combining demographic distributions, spatial assignments, and mobility patterns against real-world data remains rare.

Most synthetic population studies focus on demographic accuracy but neglect spatial validation with real mobility data, social network construction, daily activity schedules, and integration with LLM agent frameworks. GenWorld provides an end-to-end pipeline that covers these aspects.

### 2.4 Knowledge Distillation for Agent Simulation

Knowledge distillation [14] has been widely applied in machine learning to compress large models into efficient ones. Recent applications include:

Beyond model compression, recent work explores abstraction and software architecture to scale LLM-agent simulations. Chopra et al. [4] introduce *LLM archetypes*, where many agents share an archetypal LLM policy to increase throughput at scale, but this can reduce individual-level heterogeneity and online adaptivity. SocioVerse [45] targets population-scale social simulation by aligning LLM agents to a large pool of real users and standardizing simulation procedures; however, it relies on large external datasets and its alignment pipeline can be costly to reproduce or transfer. For influence diffusion in social networks, LLM-AIDSim [44] integrates LLM-enhanced agents into classical diffusion simulation pipelines, but the approach is task-specific and may not directly generalize to open-ended urban decision spaces. From a systems perspective, SALLMA [2] proposes a layered multi-agent architecture with orchestration and containerized deployment; while improving modularity and scalability, it does not inherently remove per-decision LLM inference costs and can require substantial engineering infrastructure.

**LLM Distillation.** Distilling large language models into smaller, faster models while maintaining performance is an active area of research. However, most work focuses on natural language tasks, not agent decision-making in complex environments.

**Agent Behavior Cloning.** Imitation learning and behavior cloning train efficient policies from expert demonstrations. GenWorld extends this paradigm by using LLMs as "expert demonstrators" to generate training data for efficient student models.

We apply knowledge distillation to enable city-scale LLM agent simulation. Our approach estimates the teacher's discrete decision distribution via repeated Monte Carlo sampling and compiles the resulting probabilistic policy into efficient lookup tables, shifting expensive inference out of the simulation loop and enabling large speedups in typical settings for large-scale simulations.

As summarized in Table 1, GenWorld combines **building-level population grounding** with census-validated demographics, **city-scale scalability** via offline knowledge distillation (200,000+ agents), **multi-layer social networks** derived from spatial co-location, and **schema-validated LLM-ready interfaces** that produce machine-readable behavioral traces in a real-city instantiation.

## 3 Agent Interface

GenWorld exposes a lightweight decision interface for LLM agents and records each decision as a structured log entry. This interface is designed to enable post-hoc qualitative inspection of agent routines and failure modes and provide machine-readable decision traces for offline distillation. Concretely, each decision consumes a binned observation $\tilde{o}_{i,t}$ and a finite candidate set $\mathcal{A}_{i,t}$, and produces a schema-conformant JSON action, a validator bit, and (if needed) a deterministic fallback outcome, all recorded as a log entry.

**Observation and Action Schema** At each decision point for agent $i$ at time $t$, the simulator constructs a decision context from the city state $x_t$ (time, environment signals, and infrastructure states), a synthesized persona $u_i$ produced by the population instantiation pipeline (core demographics and spatial anchors such as home/work/school when available, with optional household and social features), and optionally short-term memory summaries $m_{i,t}$ distilled from recent logs. This context is denoted as $c_{i,t} = (x_t, u_i, m_{i,t})$. Given a decision query $q_t$, the

environment deterministically produces a binned observation and a finite candidate action set:

$$\tilde{o}_{i,t} = \phi(c_{i,t}; q_t),$$
$$\mathcal{A}_{i,t} = \kappa(q_t, \tilde{o}_{i,t}).$$

The function $\phi$ is implemented as a deterministic encoder stack that includes coarse binning and query-specific formatting. A prompt composer $g(\tilde{o}_{i,t}, q_t)$ assembles a stable template with question-specific slots. The agent then outputs a structured JSON action $a_{i,t} \in \mathcal{A}_{i,t}$ following a fixed schema (e.g., activity type). A deterministic validator $v(\tilde{o}_{i,t}, a_{i,t}) \in \{0, 1\}$ enforces schema and feasibility constraints; invalid actions trigger a deterministic safe fallback before execution, and all artifacts are logged. Figure 2 illustrates a representative query where raw persona/state fields are deterministically mapped into coarse bins before being passed to the LLM. Figure 3 summarizes how the resulting structured outputs are executed into full-day trajectories by lightweight deterministic rules. In this implementation, persona slices are intentionally sparse, while richer preference/trait slices can be added as optional extensions or treated as latent variables depending on the target application.

**Two-Tier Decision Queries for Long-Horizon Rollout** Decision-making is separated into two structured outputs with different time scales. `ActivityPreference` is a per-agent, persona-conditioned preference profile that is initialized once (and optionally refreshed) and defines propensities over activity types for each high-level intention. `DayPlan` is a per-day (or per-checkpoint) plan that specifies a small mixture of intention-chain templates together with discretized POI-selection weights. The plan sampling index is denoted by $k$ (day-start or checkpoint), which is much sparser than the simulator time step $t$ used for execution, and the city state at plan sampling time is written as $x_k$. The intention space is fixed to {`home`, `duty`, `leisure`, `maintenance`}.

This two-tier abstraction is grounded in time-geography theory [12]: daily mobility is constrained by capability (physical limits), coupling (coordination with others), and authority (institutional schedules). Our intention hierarchy (home/duty/leisure/maintenance) captures these canonical constraint classes, while the activity vocabulary covers the primary purposes observed in national time-use surveys. The fixed ontology trades open-ended expressiveness for tractability and repeatability; extending the vocabulary is straightforward within the same interface contract.

Critically, the day-level query is not conditioned on a single intention; instead, the simulator provides a small, day-type-specific candidate set of intention-chain templates (e.g., weekday vs. weekend variants) and includes this candidate set as part of the binned context. During rollout, agents sample a `DayPlan` at day start, and the simulator consumes it through a lightweight executor (as shown in Figure 3) to produce an explicit trajectory of simulator actions. Concretely, an intention-chain template is sampled from the day-type-specific candidate set, expanded into activity types by sampling `ActivityPreference`, and grounded into concrete destinations via a fixed activity-to-place ontology and feasibility checks. Overrides may be requested by the agent or forced by the simulator when feasibility checks fail or exogenous events invalidate the plan. In both cases, a deterministic return-home fallback is applied and the agent stays at home until the next plan sampling time (day-start or checkpoint). Section 4 describes how decision traces are collected under binned contexts and compiled into scalable student policies.

**Formal Contract Summary** The formal contract (Figure 2) is summarized as follows. The simulator deterministically maps raw persona and state fields into coarse bins via encoders $b_u$ and $b_x$:

$$\begin{aligned}
\mathcal{I} &= \{\texttt{home}, \texttt{duty}, \texttt{leisure}, \texttt{maintenance}\},\\
\tilde{u}_i &= b_u(u_i), \quad \tilde{x}_k = b_x(x_k),\\
\tau &\in \{\texttt{weekday}, \texttt{weekend}\},\\
\mathcal{C}_\tau &\subseteq \mathcal{I}^*.
\end{aligned}$$

where $\tau$ is a coarse day-type label and $\mathcal{C}_\tau$ is a small predefined candidate set of intention-chain templates. The per-agent query defines a conditional categorical distribution over activity types given intention $z$:

$$\begin{aligned}
A_i(z) &:= \texttt{ActivityPreference}_i(z),\\
A_i(z) &= \{(a, p(a \mid z))\}_{a \in \mathcal{A}_z}, \quad z \in \mathcal{I},\\
a &\in \mathcal{A}_z, \quad \sum_{a \in \mathcal{A}_z} p(a \mid z) = 1,
\end{aligned}$$

where $\mathcal{A}_z$ is a small predefined set of activity types allowed under intention $z$ (Appendix A.3). The per-

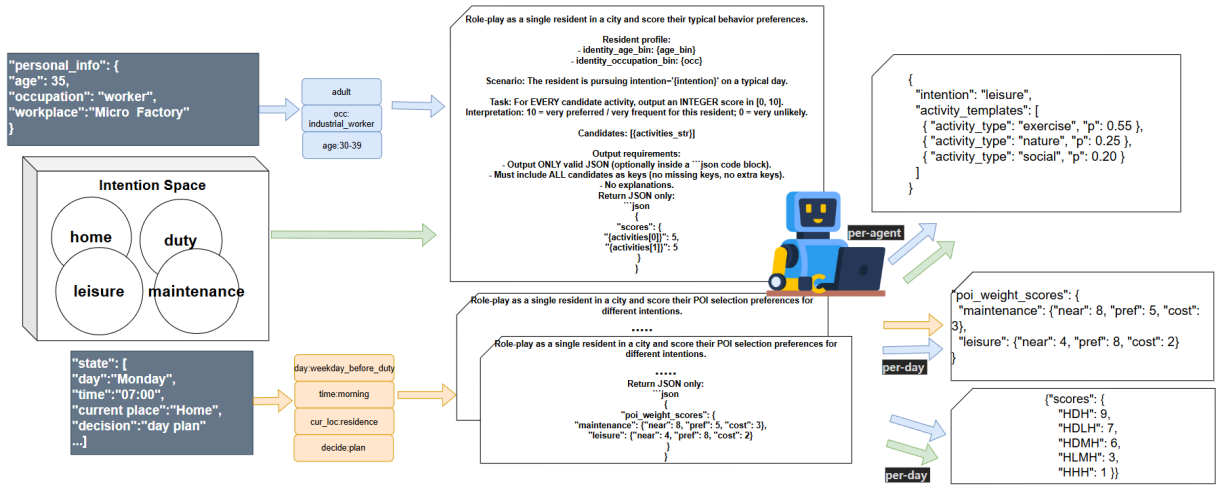



Figure 2: Query-conditioned prompt construction for our structured decision interface. Raw persona/state fields are deterministically mapped into coarse bins and are not included verbatim in the prompt. The figure schematically illustrates prompt variants used in this instantiation: a per-agent `ActivityPreference` query over a fixed candidate set under a given intention, and day-level prompts that score POI-selection preferences over `near/pref/cost` weights and intention-chain templates over a predefined chain candidate set. In this default instantiation, POI-weight scoring and intention-chain scoring are issued jointly as a single `DayPlan` query, but they can also be queried separately. Input features are represented using coarse discrete bins, while candidate scores returned by the teacher are integers in $[0, 10]$ over a predefined option set. Section 4 describes how these structured traces are aggregated and compiled for scalable rollout.

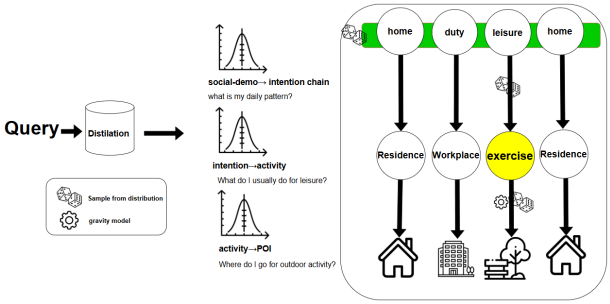



Figure 3: Plan-to-trajectory execution with a two-tier decision structure. `ActivityPreference` provides persona-conditioned activity propensities, while `DayPlan` specifies intention-chain templates and POI-selection weights. A lightweight executor produces explicit trajectories through fixed ontologies and feasibility checks.

day (or per-checkpoint) query returns:

$$
\begin{aligned}
D_{i,k} &= \texttt{DayPlan}_{i,k}(\tilde{x}_k, \tilde{u}_i, \mathcal{C}_\tau), \\
D_{i,k} &= (r_{i,k}, C_{i,k}, w_{i,k}), \\
C_{i,k} &= \{(c_j, \pi_j)\}_{j=1}^{|\mathcal{C}_\tau|}, \\
r_{i,k} &\in \{0,1\}, \quad \sum_{j=1}^{|\mathcal{C}_\tau|} \pi_j = 1, \\
c_j &\in \mathcal{C}_\tau \subseteq \mathcal{I}^*, \\
w_{i,k}(z) &= (\ell_{i,k}^{\texttt{near}}(z), \ell_{i,k}^{\texttt{pref}}(z), \ell_{i,k}^{\texttt{cost}}(z)), \\
\ell_{i,k}^{*}(z) &\in \{0, \ldots, 10\}, \quad z \in \mathcal{I},
\end{aligned}
$$

where $\mathcal{C}_\tau$ is a small predefined candidate set of intention-chain templates (in our instantiation, $|\mathcal{C}_\tau| = 6$ per day type). Here $r_{i,k}$ is an override request flag, each $c_j$ is an intention-chain template, and $w_{i,k}(z)$ specifies discretized POI-selection weights for intention $z$. Here $k$ denotes the plan sampling index (day-start or checkpoint), which is much sparser than the execution time step. Overrides may also be forced by the simulator when feasibility checks fail; in either case, a deterministic return-home fallback is applied.

**Tool-Oriented Interface, Robustness, and Traceability** The interface is realized as stable prompt templates with strict JSON schemas that are validated and logged by the simulator, and can be wrapped by standard tool-calling middleware when needed. A fixed, query-conditioned observation schema, a discrete and bounded action space with strict validation, and deterministic execution semantics are enforced. At city scale, even rare formatting or parsing failures can derail long simulations. LLM decisions are therefore constrained to a small discrete action set with a fixed schema, and strict validation and deterministic fallback rules are enforced in the decision logger. This design makes decision traces di-

rectly machine-readable and suitable for downstream analysis and policy compilation (Section 4).

# 4 Distillation and Scaling

To scale LLM-driven decision-making to city-scale simulations, the teacher's stochastic decision behavior is distilled into empirical score vectors and sampling distributions under discretized contexts by repeatedly querying the LLM under identical context keys and aggregating its scores over a fixed candidate set (e.g., intention-chain templates or intention-conditioned activity templates). Because the interface bins raw contexts into discrete keys and restricts each query to a finite candidate set with strict validation, the teacher can be repeatedly queried under identical keys and its scores can be aggregated. The key idea is to shift expensive inference out of the simulation loop: a one-time offline cost is paid to estimate these distributions, and the resulting compiled tables are executed via amortized constant-time lookup and sampling given bounded candidate sets per query, with respect to the number of agents and decision steps.

In a micro-benchmark on the compiled `ActivityPreference` table, Python lookup achieves 1.85M queries/s ($0.54\,\mu$s per query) over 200,000 randomized context keys. While absolute throughput depends on hardware and implementation details, this benchmark highlights the potential for large speedups relative to online LLM inference in typical settings. End-to-end wall-clock time per simulator step also includes environment updates, routing, and execution overheads. Prompt templates used for distillation are listed in Appendix A.4.

**Action Primitives and Context Discretization** Repeated sampling requires that the teacher be queried under identical contexts. Following the interface contract in Section 3, raw persona and state are discretized into bins (e.g., $\tilde{u}_i = b_u(u_i)$, $\tilde{x} = b_x(x)$) and each decision query $q_t$ is treated as defining its own finite action space. Concretely, for each query type $q_t$ (e.g., `ActivityPreference` or `DayPlan`), an executable discrete action set $\mathcal{A}_{q_t}$ is defined that matches the simulator's structured schema and validation rules. A finite context key $s = (\tilde{u}_i, \tilde{x}, q_t, \tau)$ is then formed, where $\tau$ indexes the day-type-specific candidate template set used by `DayPlan`. This makes repeated offline teacher aggregation well-defined and enables compilation into amortized constant-time lookup policies. The day-type indicator $\tau$ is included explicitly because the `DayPlan` candidate set differs across day types (e.g., weekday vs. weekend).

**Computational Motivation** At city scale, a direct teacher-driven simulation requires $O(NT)$ LLM calls, where $N$ is the number of agents and $T$ is the number of decision points per simulated day. For example, $N = 200{,}000$ agents with 15-minute time steps over 24 hours yields $T = 96$ and thus $1.92 \times 10^7$ calls for a single day, which is computationally expensive in practice. Distillation reduces simulation-time inference to amortized constant-time table lookup and sampling with respect to the number of agents and decision steps.

**Repeated Teacher Query Aggregation** For a fixed candidate set $\mathcal{A}_{q_t}$ and context key $s$, $K$ teacher score vectors $\{r^{(k)}(\cdot)\}_{k=1}^{K}$ are sampled, where each query returns an integer score $r^{(k)}(a) \in [0, 10]$ for every candidate $a \in \mathcal{A}_{q_t}$. The mean score and consistency statistics are aggregated:

$$\mu(a \mid s) = \frac{1}{K} \sum_{k=1}^{K} r^{(k)}(a), \tag{1}$$

$$\sigma(a \mid s) = \sqrt{K^{-1} \sum_{k=1}^{K} \big(r^{(k)}(a) - \mu(a \mid s)\big)^2}. \tag{2}$$

The aggregated mean scores are normalized across candidates into a categorical sampling distribution $\pi(\cdot \mid s)$, which is used for simulation-time sampling. Since scores are in $[0, 10]$, an executable sampling distribution is constructed by normalizing the mean scores:

$$\pi(a \mid s) = \text{Normalize}\left(\mu(a \mid s)\right), \quad a \in \mathcal{A}_{q_t}. \tag{3}$$

The score variability $\sigma(a \mid s)$ is reported to quantify teacher consistency across repeated queries (and to diagnose context regions with high variability).

**Policy Compilation and Simulation-Time Inference** The aggregated scores and sampling distributions (e.g., $\mu(\cdot \mid s)$ and $\pi(\cdot \mid s)$) are compiled into per-query lookup tables keyed by discretized context features (persona bins, time bins, coarse location types, scenario indicators, and day-type indicators). During simulation, agents sample an intention-chain template or activity template according to the distilled distribution rather than querying the LLM, and execute the sampled schema through the same validator/executor as the teacher outputs. Scoring and sampling over intention-chain templates enables

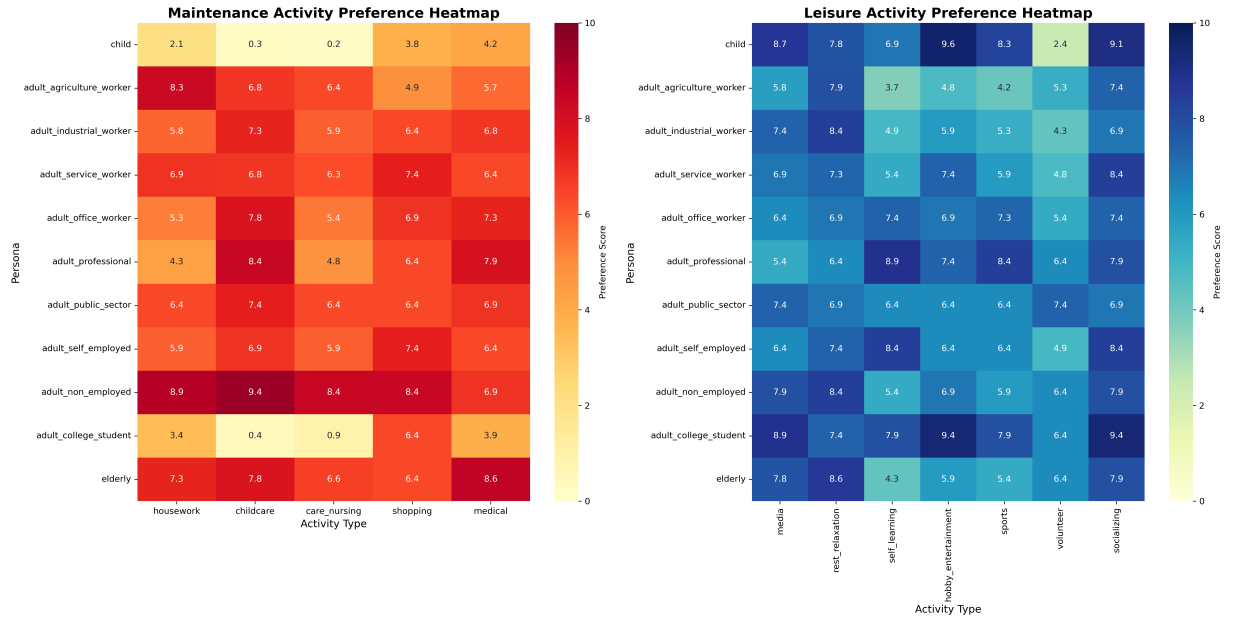
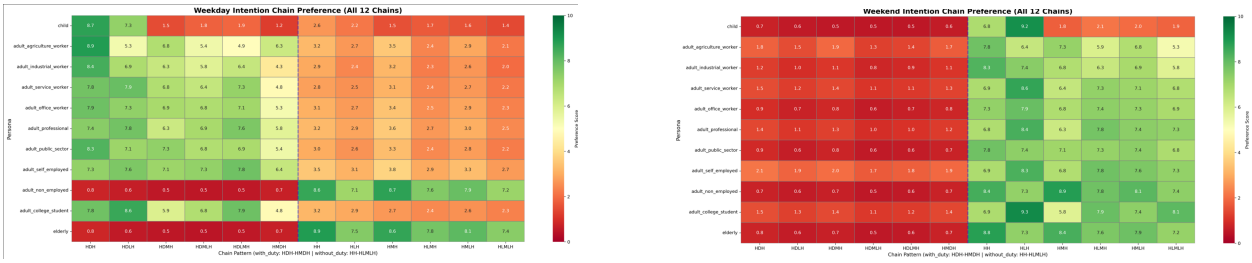



Figure 4: Teacher preference scores (0–10) for `ActivityPreference` across persona categories (rows) and candidate activity types (columns), shown separately for maintenance (left) and leisure (right). The scores define the simulation-time sampling distribution used by the compiled policy.




(a) Weekday intention-chain template preference.

(b) Weekend intention-chain template preference.

Figure 5: Distilled teacher scores for `DayPlan` intention-chain templates, shown separately for weekday and weekend candidate sets.

long-horizon diversity while keeping the execution interface lightweight. This compilation separates two concerns:

- **Teacher inference (offline)**: generate multiple samples per context to estimate $\mu(\cdot \mid s)$ (and $\sigma(\cdot \mid s)$), then derive $\pi(\cdot \mid s)$.
- **Agent rollout (online)**: execute a lightweight stochastic decision rule by table lookup and sampling.

**Context Design and Coverage** To make compilation feasible, contexts are discretized into a finite key space (e.g., persona bins, coarse location types, and time bins) and representative contexts are sampled according to the instantiated population distribution. This allows the offline sampling budget to be allocated where it matters most while keeping simulation-time inference amortized constant-time with respect to the number of agents and decision steps. This discretization trades off fidelity for tractability: behavior matching depends on context key design and coverage, and unseen keys may require backing off to coarser keys or a conservative default.

# 5 Empirical Grounding of the Urban World

Our reference instantiation integrates multi-source empirical datasets, including official census statistics and administrative boundaries, building footprints

and POIs, parcel-level land-use labels, a complete road network with elevation, and anonymized mobility data for commuting diagnostics. These inputs provide constraints for population synthesis and spatial grounding, and also provide independent signals for validation.

Detailed data sources and processing steps are provided in Appendix A.2 (Table 2).

## 5.1 Population and Environment Foundation

This section describes the empirically-grounded population and environment foundation used in our Higashihiroshima reference instantiation, grounded in publicly available census tabulations and geospatial layers (buildings, land use parcels, school districts, POIs, and roads), synthesizing 196,608 individuals across 90,093 households. The formulation combines demographic micro-synthesis under tract-level census constraints with spatial grounding of home, school, and work locations under capacity and distance constraints.

### 5.1.1 Tract-Level Micro-Synthesis and Attribute Assignment

For each tract $t$, the total population $N_t$ is treated as a hard constraint and an age–gender joint distribution is estimated whose marginals match census age counts and gender totals. A 2D IPF procedure is adopted on an age $\times$ gender matrix $M^{(t)}$:

$$M_{a,g}^{(k+\frac{1}{2})} = M_{a,g}^{(k)} \cdot \frac{n_{t,a}}{\sum_{g'} M_{a,g'}^{(k)}} \tag{4}$$

$$M_{a,g}^{(k+1)} = M_{a,g}^{(k+\frac{1}{2})} \cdot \frac{n_{t,g}}{\sum_{a'} M_{a',g}^{(k+\frac{1}{2})}} \tag{5}$$

where $n_{t,a}$ is the census count of age bin $a$ in tract $t$, and $n_{t,g}$ is the census total of gender $g \in \{\text{male}, \text{female}\}$. $M^{(t)}$ is initialized with a strictly positive prior (e.g., uniform or tract-independent) and Eq. (5) is iterated until marginal errors fall below $\epsilon$ or for a fixed number of rounds. Individuals are then sampled from the normalized joint distribution, and a concrete integer age is sampled uniformly within the selected age bin.

Given the sampled individuals, households are formed using the tract household-size histogram (1,2,3,4,5,6+) with a lightweight plausibility heuristic (e.g., capping household size at 6). The census household count target $H_t$ is enforced and household sizes are sampled to match the tract histogram. Employment status and occupation categories are then assigned for working-age individuals so that tract-level employed totals and occupational marginals match the census. Let $\mathcal{I}_t$ be individuals in tract $t$, and $\mathcal{W}_t \subset \mathcal{I}_t$ be working-age individuals. Denote the census employed target as $E_t$ and the census occupation target counts as $C_{t,o}$ for occupation category $o \in \mathcal{O}$. Let $E'_t = \min(E_t, |\mathcal{W}_t|)$ and let $C'_{t,o}$ be adjusted occupation targets derived from $\{C_{t,o}\}_{o\in\mathcal{O}}$ by padding/truncation so that $\sum_{o\in\mathcal{O}} C'_{t,o} = E'_t$. The following constraints are enforced:

$$\sum_{i\in\mathcal{W}_t} \mathbb{I}[\text{employed}_i] = E'_t \tag{6}$$

$$\sum_{i\in\mathcal{W}_t} \mathbb{I}[\text{employed}_i \wedge \text{occ}_i = o] = C'_{t,o}, \quad \forall o \in \mathcal{O} \tag{7}$$

Eq. (6)–(7) are realized via seeded sampling: an employed subset of size $E'_t$ is drawn and an occupation multiset with counts $C'_{t,o}$ is assigned, followed by a tract-seeded random permutation.

### 5.1.2 Spatial Grounding of Home, School, and Work

Households are assigned to residential buildings within each tract using a capacity-aware allocation; students are assigned to schools using district polygons when available with nearest-school fallback; university assignment uses a distance-based stochastic choice with weights proportional to $1/d^2$. For workplace allocation, employed individuals are mapped to *landuse* parcels (not building IDs) using an occupation-conditioned landuse prior (occupation$\rightarrow$landuse mapping with ratios $r_{o,l}$) together with a maximum commute-distance constraint $d_{\max}$.

*Capacity inference by area (quotas).* For occupation $o$, let total employees be $N_o$, eligible landuse categories be $\mathcal{L}_o$, and the configured landuse ratio be $r_{o,l}$ for $l \in \mathcal{L}_o$ with $\sum_{l\in\mathcal{L}_o} r_{o,l} = 1$. For each landuse parcel $j$ of category $l$ with area $A_j$, an occupation-specific quota is defined:

$$q_{j,o} = \frac{N_o\, r_{o,l}\, A_j}{\sum_{k\in\mathcal{P}_l} A_k}, \quad j \in \mathcal{P}_l,\ l \in \mathcal{L}_o \tag{8}$$

where $\mathcal{P}_l$ is the set of parcels with landuse category $l$. Fractional quotas are converted into integer capacities $\hat{q}_{j,o}$ (e.g., via floor with remainder redistribution or stochastic rounding) to preserve total capacity per occupation.

*Gravity-based allocation.* We employ a gravity model to assign workplaces, balancing employment opportunities with distance decay. Let $d_{ij}$ be the haversine distance between employed individual $i$'s home and landuse parcel $j$. The probability $P_{ij}$ of

individual $i$ choosing workplace $j$ is proportional to the parcel's destination attractiveness (capacity) and inversely proportional to commute distance:

$$P_{ij} \propto A_j^{\alpha} \cdot f(d_{ij}) \cdot M_{ij} \tag{9}$$

where $A_j$ is the capacity (attractiveness) of parcel $j$, $f(d) = d^{-\beta}$ is the distance decay function with friction parameter $\beta$, and $M_{ij}$ is a binary mask enforcing occupation compatibility ($M_{ij} = 1$ if parcel $j$ supports individual $i$'s occupation $o_i$ and $j$ has remaining capacity, else 0). We set $\alpha = 1$ and calibrate $\beta$ against empirical mobility data. The assignment is performed stochastically:

$$j^* \sim \text{Categorical}(\{P_{ij}\}_j) \tag{10}$$

This probabilistic approach allows for a realistic distribution of commute distances, including long-tail commutes, unlike strict distance minimization.

#### 5.1.3 Derived Social Networks

Multi-layer networks are a deterministic byproduct of the assigned home/school/work locations and institutional membership. While not used by the agent interface or the experiments in this paper, they are retained as an optional artifact for internal consistency checks and future extensions:

$$G = (V, E), \tag{11}$$

$$E = E_{\text{household}} \cup E_{\text{home}} \cup E_{\text{school}} \cup E_{\text{work}} \cup E_{\text{neighborhood}} \tag{12}$$

where edges represent interaction opportunities induced by shared households, shared residential buildings, shared schools, shared workplace landuse, and neighborhood proximity. To keep graphs sparse at scale, degrees are capped or edges are sampled within large buildings/institutions and edges can optionally be weighted by co-location frequency.

#### 5.1.4 Urban Environment Integration

The platform integrates multiple layers of urban infrastructure:

$$\mathcal{E} = \{\mathcal{P}, \mathcal{R}, \mathcal{B}, \mathcal{A}\} \tag{13}$$

where:

- $\mathcal{P}$: POI catalog with categorical attributes $\mathcal{P} = \{(p_i, \text{type}_i, \text{capacity}_i, \text{hours}_i)\}$
- $\mathcal{R}$: Road network graph $\mathcal{R} = (V_r, E_r, w_r)$ with edge weights (distance, speed, capacity)
- $\mathcal{B}$: Building set with spatial footprints and land use $\mathcal{B} = \{(b_i, \text{geom}_i, \text{use}_i, C_i)\}$
- $\mathcal{A}$: Administrative hierarchy (census blocks, districts, city) for spatial aggregation

When explicit capacities, opening hours, or road-capacity attributes are missing in the source layers, the implementation uses conservative defaults or simple rule-based proxies (e.g., POI-type-specific heuristics and road-class-based speed/capacity settings) to support feasibility checks.

### 5.2 Activity Generation and Temporal Grounding

We implement a **hybrid generative mechanism** to ensure both behavioral realism and temporal fidelity. While the **sequence and semantics** of daily activities (e.g., the decision to visit a gym after work) are generated by the LLM-distilled policy to capture heterogeneous preferences, the **temporal attributes** (start time and duration) are grounded in the **National Time Use Survey**. Specifically, once an activity type is selected by the agent, its timing is sampled from the corresponding empirical distribution (e.g., 'Sports' duration distribution for a 'Gym' visit), thereby preventing unrealistic hallucinations common in pure LLM scheduling.

We utilize the **action initialization probability** (derived from activity start-time statistics) rather than the raw **action participation rate** (occupancy). Using raw occupancy rates as sampling probabilities—a common pitfall—would incorrectly bias the duration of activities. Our pipeline explicitly separates the *decision to start* an activity from the *duration* of the activity, ensuring that the generated temporal dynamics mathematically align with the aggregate census observations.

### 5.3 Population Distribution Validation

We validate our synthetic population against census data at the tract level to ensure demographic accuracy.

#### 5.3.1 Census Data Validation

Our population synthesis method generates 196,608 individuals across 90,093 households in Higashihiroshima. We validate the synthetic population against 2020 Japanese Census tabulations at census-tract granularity across multiple demographic dimensions.

For household size statistics, the census reports *general household* counts, while some tracts include non-household residents (e.g., dormitories or institutional facilities). We therefore evaluate household size distributions on tracts where total population equals general-household persons (see Appendix A for details).

**Distributional Fit Metrics** We distinguish hard constraints from soft-fit metrics. Tract-level total population is constrained to match census totals exactly, yielding very close agreement with census totals by construction. We therefore emphasize distributional similarity for variables not enforced as exact constraints.

After restricting census tabulations to the instantiated study area, we obtain 198 finest-resolution census units (HYOSYO=2/4). We evaluate demographic fit on 185 tracts; 13 census units with zero population and zero households (e.g., industrial parks) are excluded. Gender ratios are well matched (male ratio MAE $< 0.02$). Age distributions achieve mean L1 = 0.1229 (median 0.10, max 0.31), mean KS = 0.0299 (max 0.12), and mean JS = 0.0047 (max 0.02), with 95% of tracts having L1 $< 0.20$. Household size distributions achieve mean L1 = 0.0547, mean KS = 0.0269, and mean JS = 0.0075. Employment counts (15+) show high tract-level agreement ($R^2 > 0.99$). Occupation distributions achieve mean L1 = 0.1945, mean KS = 0.0972, and mean JS = 0.0382. The tight distribution of per-tract errors reflects the effectiveness of the IPF constraints.

#### 5.3.2 Spatial Distribution Validation

Unlike TAZ-based methods that assign residents to abstract zones, our building-level approach assigns households to specific georeferenced buildings under tract-level and capacity constraints. Because building footprints and land-use labels may be incomplete in a small number of tracts (e.g., industrial parks), we report explicit assignment diagnostics rather than silently forcing fallback placements.

In our reference instantiation, 89,933 out of 89,988 tract-level census target households are successfully assigned to residential buildings. The remaining 55 target households belong to tracts with zero residential supply under our mixed residential-identification rules (e.g., industrial/logistics areas). We additionally report unmapped synthetic households due to definition mismatch between census targets and the sampled household list.

#### 5.3.3 School Assignment Validation

School assignment uses building-level home locations. Elementary and junior-high students are assigned by school-district polygons with nearest-school fallback. High-school assignment is nearest-school based with limited randomness among candidates within a distance threshold, and university assignment uses a gravity-style stochastic choice with weights proportional to $1/d^2$.

In our reference instantiation, 43,260 out of 43,557 students are assigned to a school (99.32%); 53 students are flagged as `no_location` due to missing home geolocation (caused by a small number of households not mapped to residential buildings under supply constraints). We report the assigned school enrollment distribution in Figure A2.

### 5.4 Mobility Pattern Validation

We compare commuting statistics against anonymized mobile phone mobility data from Yahoo Japan Mobility (YJMob100K) [40]. The dataset discretizes location pings into 500m × 500m grid cells and timestamps into 30-minute bins, with the metropolitan area undisclosed for privacy. For our case study, we extract a subregion consistent with the Higashihiroshima area by registering the released mesh grid via manual rigid alignment. The registration uses coastline landmarks and major terrain features as control points, with an estimated alignment error of <500m (one grid cell). A sensitivity analysis indicates that commute distance distributions are robust to registration errors within this range. The registration script and control point coordinates are provided in our repository (`data_prepare/step3_assign/yjm_registration.py`); the comparison is treated as a commuting-distance diagnostic rather than an OD-flow benchmark.

We infer each user's home mesh from nighttime records and work mesh from daytime records (fixed time windows), then derive a commuting distance distribution in the mesh space. Figure 6 summarizes the extracted commuting patterns for the selected subregion.

To address the distinction between parameter calibration and model validation, we temporally split the 75-day YJMob dataset into two disjoint subsets: a *calibration set* (days 0–48, approximately 7 weeks) used exclusively for parameter tuning, and a held-out *validation set* (days 49–74, approximately 3–4 weeks) reserved for out-of-sample evaluation. We identified the weekly periodicity in the data and restricted home/work inference to weekday records only

to avoid weekend mobility patterns confounding the commute signal.

On the calibration set, we extracted 5,422 commuters with a mean commute distance of 6.41 km (median 3.81 km). We calibrated the gravity model's friction parameter $\beta$ by performing a parameter sweep and selecting the configuration that minimizes the Kolmogorov-Smirnov (KS) statistic between the synthetic and observed commute distance distributions. The optimal parameter was found to be $\beta = 0.5$.

To validate this calibrated model, we evaluated against the *held-out validation set*, which contains 4,412 commuters with a mean commute distance of 5.98 km (median 3.50 km). Importantly, the calibration and validation sets show strong temporal consistency (KS = 0.032), confirming they are drawn from the same underlying distribution. Under the calibrated configuration, the synthetic population (90,394 employed individuals, mean commute 7.16 km, median 5.87 km) achieves a KS statistic of **0.162** against the held-out validation set and a Wasserstein distance of **1.60 km**. This out-of-sample evaluation provides a stricter assessment than using the same data for both calibration and validation.

Figure A4 compares the resulting distributions. We treat this comparison as a commuting-distance diagnostic rather than a strict OD-flow correlation, because the observed mesh space is anonymized and requires manual registration.

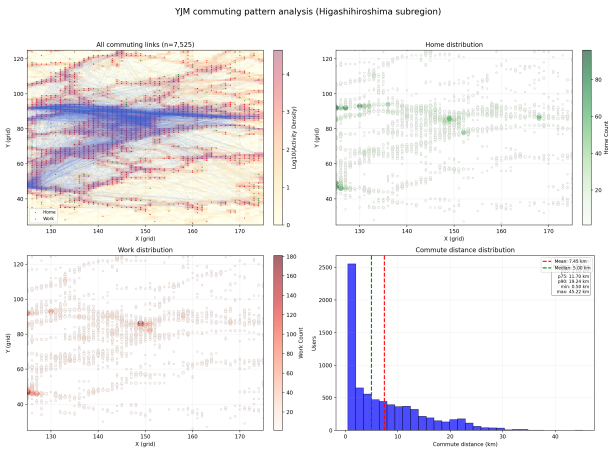



Figure 6: Commuting pattern extraction from YJ-Mob100K after registering the anonymized mesh grid to our study area. The figure visualizes inferred home/work points and commuting distance statistics for the extracted subregion.

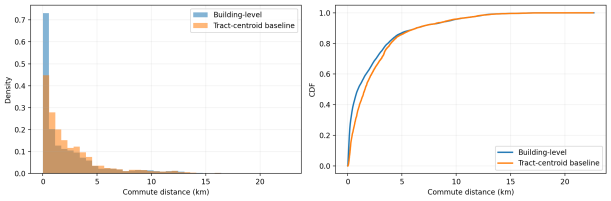



Figure 7: Commuting distance distributions under building-level grounding versus a tract-centroid baseline. The baseline collapses within-tract heterogeneity by placing all households at tract centroids, illustrating how coarse spatial grounding can distort short-range commuting structure even when workplace assignments are held fixed.

# 6 Platform Architecture

GenWorld emphasizes **modularity** (independent components for flexibility), **scalability** (efficient handling of 200,000+ agents in our reference instantiation), and **accessibility** (LLM-compatible interfaces for AI researchers). Figure 8 illustrates the detailed system architecture. Platform UI screenshots (Streamlit-based interface) are provided in Appendix Figure A5.

## 6.1 System Overview

The platform is organized into three layers:

**Layer 1: Population and Environment Foundation** Instantiates the georeferenced urban world and synthetic population under census constraints and reports validation diagnostics; see Section 5.

**Layer 2: Agent Decision Framework** Exposes a structured agent–environment interface with binned observations and finite JSON-validated action candidates, enabling rule-based, teacher-LLM, and distilled-student policies; see Sections 3 and 4.

**Layer 3: Simulation Engine** Orchestrates time-stepped multi-agent execution with feasibility checks, system-level consistency updates, and detailed logging; see Section 6.2.

The following subsections detail the simulation engine.

## 6.2 Simulation Engine

The simulation engine orchestrates time-stepped multi-agent execution, managing time progression,

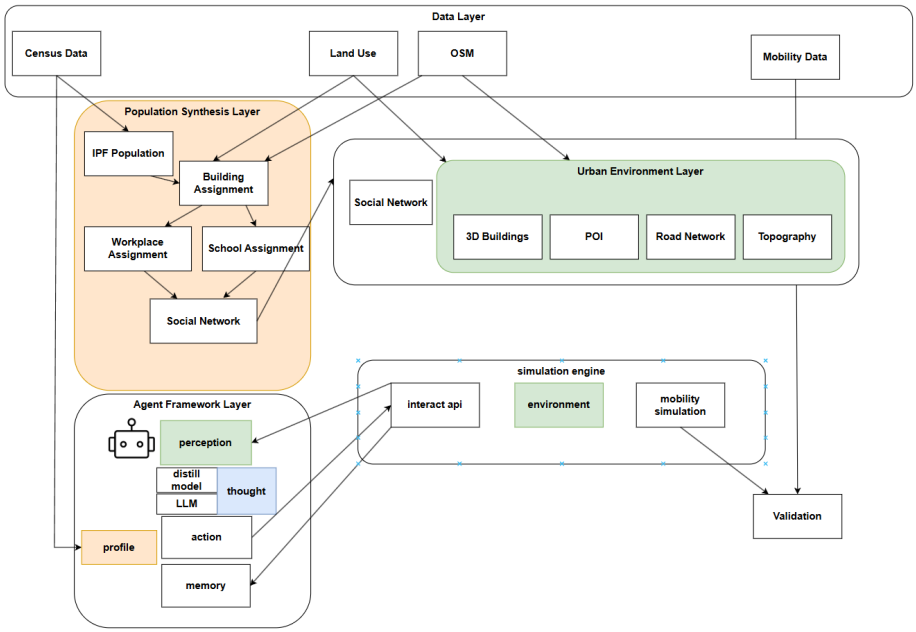



Figure 8: GenWorld System Architecture. The platform is organized into three layers: Population & Environment Foundation, Agent Decision Framework, and Simulation Engine. The architecture supports LLM integration and knowledge distillation for city-scale scalability.

spatial dynamics, and system-level feasibility constraints. The engine is designed to support both small-scale LLM experiments and large-scale distilled simulations.

**Time-stepped Execution (Pseudo-code)** The simulator advances in discrete time steps (typically 15-minute intervals) and executes validated actions under feasibility constraints, while recording structured decision traces for analysis and offline compilation.

This modular architecture supports repeatability through deterministic execution and configuration-based parameters, while enabling extensibility for new agent models, additional cities, and integration with external frameworks.

**Algorithm 1** Time-stepped simulation engine with structured decision interface

**for** each simulation step $t$ **do**
  determine active agents $\mathcal{S}_t$ from schedules
  **for** each agent $i \in \mathcal{S}_t$ **do**
    construct context $c_{i,t}$ from world state and persona
    $\tilde{o}_{i,t} \leftarrow \phi(c_{i,t}; q_t)$ ▷ binned observation
    $\mathcal{A}_{i,t} \leftarrow \kappa(q_t, \tilde{o}_{i,t})$ ▷ finite candidates
    $a_{i,t} \leftarrow \pi(\tilde{o}_{i,t}, \mathcal{A}_{i,t})$ ▷ rule/teacher/student
    **if** $v(\tilde{o}_{i,t}, a_{i,t}) = 0$ **then**
      $a_{i,t} \leftarrow f(\tilde{o}_{i,t})$ ▷ deterministic fallback
    **end if**
    execute $a_{i,t}$ and update agent/world states
    append decision record and trajectory log
  **end for**
  apply system-level consistency updates (e.g., travel-time feedback and POI capacity)
  record aggregate metrics (e.g., utilization and travel-time indicators)
**end for**

# 7 Results and Applications

## 7.1 Current Demonstrations

We demonstrate GenWorld's capabilities through baseline simulations and scalability tests in Higashihiroshima.

### 7.1.1 Baseline Simulation

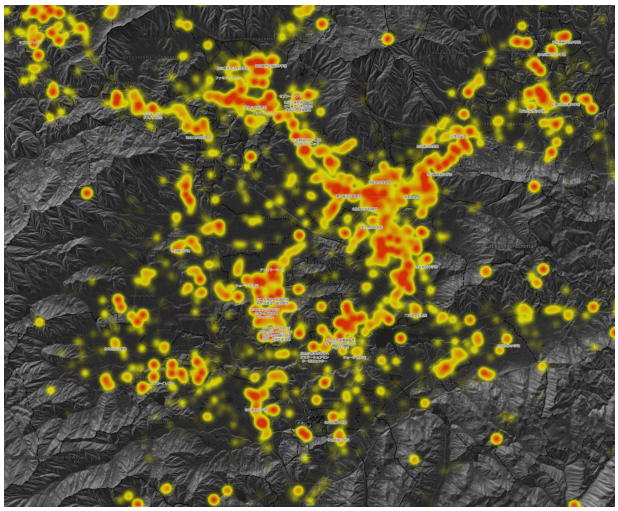
Our baseline simulation includes 196,608 agents (190,000 residents) distributed across 90,093 households in Higashihiroshima, with building-level home assignment, home/school/work anchors, and daily activity schedules executed under the structured interface.

We visualize the spatial distribution of agents and their daily commuting flows. The 3D visualization supports qualitative inspection of residential density gradients, commuting corridors, activity hotspots around commercial and institutional areas, and day–night population shifts. Figure 9 shows two snapshots of the visualized resident locations: during worktime the distribution exhibits strong clustering around activity centers (e.g., the Hiroshima University area), while at nighttime these daytime hotspots become sparse as residents return to their home neighborhoods.

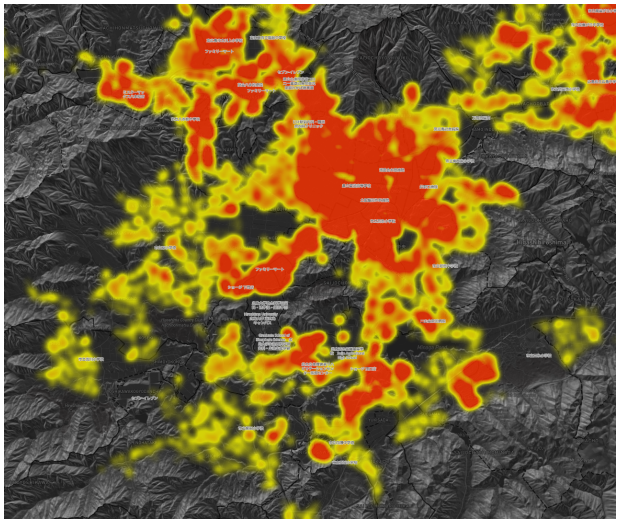
Additional weekday spatial heatmaps for representative activity types (shopping, socializing, and childcare) at multiple time windows are provided in the appendix (Figure A6).

We also summarize the city-scale diurnal rhythm by aggregating simulated activity occupancy over time. Figure 10 visualizes the 24-hour distribution of activity categories as a radial stacked plot, providing a compact view of time-of-day regularities in the baseline rollout.

We further visualize aggregate road-network traffic flow by routing simulated trips between consecutive activity locations. Figure 11 shows the all-day flow map computed from a 50,000-resident sample, where edge color intensity indicates higher accumulated volumes. Note that this is a *static shortest-path visualization* without dynamic congestion feedback; validating against real-time traffic counts and incorporating equilibrium assignment are left for future work.



(a) Worktime resident-location heatmap.



(b) Nighttime resident-location heatmap.

Figure 9: Day–night contrast of visualized resident locations in the baseline rollout. The worktime snapshot highlights dense daytime clustering around major institutional and employment centers (e.g., the Hiroshima University area), whereas the nighttime snapshot shows these areas becoming nearly empty as the population shifts back toward residential neighborhoods.

### 7.1.2 Scalability Analysis

Through offline compilation, simulation-time decision-making can be implemented as amortized constant-time table lookup and sampling under bounded candidate sets. The computational complexity comparison is as follows:

- **Online LLM**: $O(N \cdot T \cdot C_{\text{LLM}})$ per simulated day, where $N$ is agent count, $T$ is decision steps per day, and $C_{\text{LLM}}$ is per-query LLM inference cost (typically 0.5–2s for local 7B models).
- **Distilled policy**: $O(N \cdot T \cdot C_{\text{lookup}})$, where

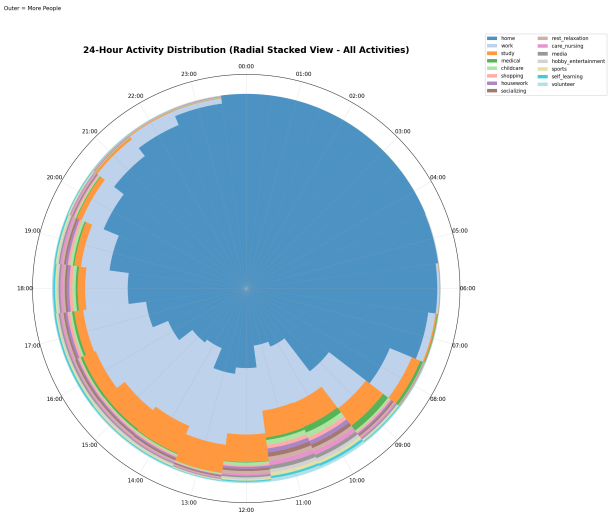



Figure 10: 24-hour activity occupancy distribution in the baseline rollout, shown as a radial stacked plot (outer radius indicates more people). The visualization highlights the expected day–night cycle: home/sleep dominates overnight, work and study increase during daytime hours, and leisure and other discretionary activities rise in the evening.

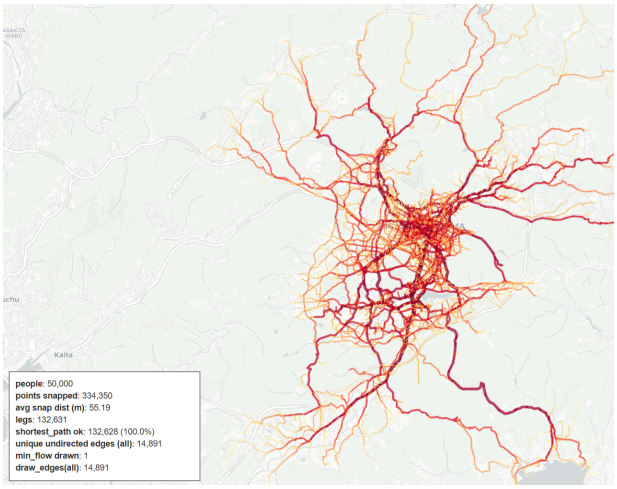



Figure 11: All-day road-network traffic flow aggregated from a 50,000-resident sample. Trips are routed via static shortest paths (no congestion feedback); edge intensity indicates accumulated volume. This is intended as a visualization of spatial demand patterns rather than a validated traffic simulation.

$C_{\text{lookup}} \approx 1\mu$s (hash table lookup + categorical sampling).

For $N = 200{,}000$ agents with $T = 96$ decision points per day (15-minute steps), online LLM simulation would require ∼19M inference calls per simulated day, which is computationally expensive in practice. Our distilled policy replaces these calls with table lookups, allowing city-scale rollout in our reference setup.

In a micro-test, Python lookup achieves 1.85M queries/s (0.54$\mu$s per query) over 200,000 randomized context keys on an Intel Core i5-14600K CPU. End-to-end wall-clock time per simulator step also includes environment updates, spatial queries, and activity execution; profiling under varying agent counts is ongoing work.

### 7.2 Summary

Our results demonstrate that GenWorld combines empirically grounded world instantiation, a structured agent interface, and scalable simulation-time rollout via offline compilation. While current validation focuses on census consistency and mobility-scale diagnostics, broader validation and calibrated policy evaluation would require additional datasets and is left for future work.

## 8 Discussion

**Limitations and Future Work** Several limitations remain in the current reference instantiation.

**Validation Scope** We validate synthetic populations against census tabulations, commuting distances against YJMob100K mobile phone data, and activity schedules against the Japanese National Time Use Survey (e-Stat). Our activity schedule validation shows good agreement for diurnal patterns (average correlation $r > 0.86$, RMSE $< 3\%$), though peak-time shifts for work/study activities suggest lunch-break modeling needs refinement. Broader validation, such as link-level traffic counts and full OD-flow correlation, would require additional calibrated datasets and is left for future work.

**Distillation Fidelity** Our distillation pipeline aggregates teacher-model responses into lookup tables, but the fidelity of this compilation is not fully validated. We use $K = 10$–$30$ samples per context key with a single teacher model (Gemma 3 27B); ablation of sampling count, temperature, and teacher model

choice is needed. We also do not quantitatively compare distilled outputs against fresh teacher queries (e.g., via KL divergence or decision agreement rate).

**Behavioral Modeling** The structured interface enables logging and analysis of LLM-driven decisions, but connecting these to human decision processes is not addressed here. Possible extensions include comparisons against human subjects or stated-preference surveys, sensitivity analyses of prompt design, and evaluation of emergent behaviors under scenario perturbations.

**Generalizability** The current implementation is instantiated in Higashihiroshima, a mid-sized Japanese city with approximately 200,000 residents. Higashihiroshima has a relatively dispersed urban form centered around Hiroshima University; scalability to denser metropolitan areas (Tokyo, Osaka) with more complex transit networks remains untested, and computational challenges may arise at 10× population scales.

Our data pipeline relies on Japan-specific sources (e-Stat census, YJMob100K mobility, Hiroshima DoBOX land use). Replication elsewhere requires equivalent data sources and adapted preprocessing; availability and format consistency vary across regions. Activity patterns and commuting behaviors also differ across urban contexts—US suburban sprawl, European compact cities, and Asian high-density development each have distinct characteristics. The distilled decision distributions may not transfer without local calibration.

**Potential Application Scenarios** Although the results reported in this paper focus on empirical grounding and scalable rollout, the same instantiation and structured decision traces also support qualitative what-if analyses. Example use cases include transportation planning (inspecting commuting-pattern shifts under hypothetical transit or land-use changes), disaster response and resilience (elevation-aware exposure inspection and evacuation accessibility under flood scenarios), and urban policy evaluation (routine or constraint modifications such as remote-work adoption and capacity policies). These scenarios are intended as illustrative demonstrations rather than calibrated forecasts.

# 9 Conclusion

GenWorld is an LLM-ready urban simulation platform that couples empirically grounded population-and-environment instantiation with a structured agent interface and scalable rollout. A reference instantiation in Higashihiroshima demonstrates end-to-end feasibility for deploying and studying LLM-driven agents in realistic urban settings.

GenWorld contributes an empirically grounded, building-level urban world together with a structured agent interface that yields machine-readable decision traces. The interface uses query-conditioned, binned observations and finite JSON-validated action candidates, enabling rule-based policies, teacher-LLM decision traces, and compiled student policies for scalable simulation-time inference. Current validation focuses on census consistency and anonymized mobile-phone mobility diagnostics; broader validation against additional datasets is left for future work. Code, configurations, and documentation will be released as open-source software upon publication, following the principles of reproducible urban research [7].

# Acknowledgments

We thank Xuesong (Simon) Zhou for his valuable suggestions.

# References

[1] James E Anderson. The gravity model. *Annu. Rev. Econ.*, 3(1):133–160, 2011.

[2] Marco Becattini, Roberto Verdecchia, and Enrico Vicario. Sallma: A software architecture for llm-based multi-agent systems. In *2025 IEEE/ACM International Workshop New Trends in Software Architecture (SATrends)*, pages 5–8. IEEE, 2025.

[3] Filip Biljecki and Yoong Shin Chow. Global building morphology indicators. *Computers, Environment and Urban Systems*, 95:101809, 2022.

[4] Ayush Chopra, Shashank Kumar, Nurullah Giray-Kuru, Ramesh Raskar, and Arnau Quera-Bofarull. On the limits of agency in agent-based models. *arXiv preprint arXiv:2409.10568*, 2024.

[5] Abdoul-Ahad Choupani and Amir Reza Mamdoohi. Population synthesis using iterative proportional fitting (ipf): A review and future research. *Transportation Research Procedia*, 17:223–233, 2016.

[6] Joshua M Epstein and Robert Axtell. *Growing artificial societies: social science from the bottom up*. Brookings Institution Press, 1996.

[7] Rosa Félix, Filipe Moura, and Robin Lovelace. Reproducible methods for modeling combined public transport and cycling trips and associated benefits: Evidence from the biclar tool. *Computers, Environment and Urban Systems*, 117:102230, 2025.

[8] Jie Feng, Jun Zhang, Junbo Yan, Xin Zhang, Tianjian Ouyang, Tianhui Liu, Yuwei Du, Siqi Guo, and Yong Li. Citybench: Evaluating the capabilities of large language model as world model. *arXiv e-prints*, pages arXiv–2406, 2024.

[9] Kunihiko Fujiwara, Ryuta Tsurumi, Tomoki Kiyono, Zicheng Fan, Xiucheng Liang, Binyu Lei, Winston Yap, Koichi Ito, and Filip Biljecki. Voxcity: A seamless framework for open geospatial data integration, grid-based semantic 3d city model generation, and urban environment simulation. *arXiv preprint arXiv:2504.13934*, 2025.

[10] Dawei Gao, Zitao Li, Xuchen Pan, Weirui Kuang, Zhijian Ma, Bingchen Qian, Fei Wei, Wenhao Zhang, Yuexiang Xie, Daoyuan Chen, et al. Agentscope: A flexible yet robust multi-agent platform. *arXiv preprint arXiv:2402.14034*, 2024.

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

[12] Torsten Hägerstrand. What about people in regional science. *Transport Sociology: Social aspects of transport planning*, pages 143–158, 1970.

[13] Samiul Hasan, Christian M Schneider, Satish V Ukkusuri, and Marta C González. Spatiotemporal patterns of urban human mobility. *Journal of Statistical Physics*, 151(1):304–318, 2013.

[14] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[15] Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, et al. Metagpt: Meta programming for a multi-agent collaborative framework. In *The twelfth international conference on learning representations*, 2023.

[16] Andreas Horni, Kai Nagel, and Kay W Axhausen. Introducing matsim. In *Multi-Agent Transport Simulation MATSim*. Ubiquity Press, 2016.

[17] John J Horton. Large language models as simulated economic agents: What can we learn from homo silicus? Technical report, National Bureau of Economic Research, 2023.

[18] Na Jiang, Andrew T Crooks, Hamdi Kavak, Annetta Burger, and William G Kennedy. A method to create a synthetic population with social networks for geographically-explicit agent-based models. *Computational Urban Science*, 2(1):7, 2022.

[19] Carlos E Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. Swe-bench: Can language models resolve real-world github issues? *arXiv preprint arXiv:2310.06770*, 2023.

[20] 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*, pages 2847–2850, 2025.

[21] Takehiro Kashiyama, Yanbo Pang, Yuya Shibuya, Takahiro Yabe, and Yoshihide Sekimoto. Nationwide synthetic human mobility dataset construction from limited travel surveys and open data. *Computer-Aided Civil and Infrastructure Engineering*, 39(21):3337–3353, 2024.

[22] Daniel Krajzewicz, Jakob Erdmann, Michael Behrisch, Laura Bieker, et al. Recent development and applications of sumo-simulation of urban mobility. *International journal on advances in systems and measurements*, 5(3&4):128–138, 2012.

[23] David Lazer, Alex Pentland, Lada Adamic, Sinan Aral, Albert-László Barabási, Devon Brewer, Nicholas Christakis, Noshir Contractor, James Fowler, Myron Gutmann, et al. Computational social science. *Science*, 323(5915):721–723, 2009.

[24] 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. *arXiv preprint arXiv:2407.18932*, 2024.

[25] Sung Yoo Lim, Hyunsoo Yun, Prateek Bansal, Dong-Kyu Kim, and Eui-Jin Kim. A large language model for feasible and diverse population synthesis. *arXiv preprint arXiv:2505.04196*, 2025.

[26] Qi Liu, Can Li, and Wanjing Ma. Gatsim: Urban mobility simulation with generative agents. *arXiv preprint arXiv:2506.23306*, 2025.

[27] Tianming Liu, Jirong Yang, and Yafeng Yin. Toward llm-agent-based modeling of transportation systems: A conceptual framework. *Artificial Intelligence for Transportation*, 1:100001, 2025.

[28] Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, et al. Agentbench: Evaluating llms as agents. *arXiv preprint arXiv:2308.03688*, 2023.

[29] Sean Luke, Claudio Cioffi-Revilla, Liviu Panait, Keith Sullivan, and Gabriel Balan. Mason: A multiagent simulation environment. *Simulation*, 81(7):517–527, 2005.

[30] Haoxuan Ma, Xishun Liao, Yifan Liu, Qinhua Jiang, Chris Stanford, Shangqing Cao, and Jiaqi Ma. Learning universal human mobility patterns with a foundation model for cross-domain data fusion. *Transportation Research Part C: Emerging Technologies*, 180:105311, 2025.

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

[32] 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*, pages 1–22, 2023.

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

[34] Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems*, 36:68539–68551, 2023.

[35] Patrick Taillandier, Benoit Gaudou, Arnaud Grignard, Quang-Nghi Huynh, Nicolas Marilleau, Philippe Caillou, Damien Philippon, and Alexis Drogoul. Building, composing and experimenting complex spatial models with the gama platform. *GeoInformatica*, 23(2):299–322, 2019.

[36] Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, et al. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025.

[37] Seth Tisue, Uri Wilensky, et al. Netlogo: A simple environment for modeling complexity. In *International conference on complex systems*, volume 21, pages 16–21. Boston, MA, 2004.

[38] 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.

[39] Zhiheng Xi, Wenxiang Chen, Xin Guo, Wei He, Yiwen Ding, Boyang Hong, Ming Zhang, Junzhe Wang, Senjie Jin, Enyu Zhou, et al. The rise and potential of large language model based agents: A survey. *Science China Information Sciences*, 68(2):121101, 2025.

[40] Takahiro Yabe, Kota Tsubouchi, Toru Shimizu, Yoshihide Sekimoto, Kaoru Sezaki, Esteban Moro, and Alex Pentland. Yjmob100k: City-scale and longitudinal dataset of anonymized human mobility trajectories. *Scientific Data*, 11(1):397, 2024.

[41] 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. *arXiv preprint arXiv:2410.21286*, 2024.

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

[43] Xiaotong Ye, Nicolas Bougie, Toshihiko Yamasaki, and Narimasa Watanabe. Mobilecity: An efficient framework for large-scale urban behavior simulation. *arXiv preprint arXiv:2504.16946*, 2025.

[44] Lan Zhang, Yuxuan Hu, Weihua Li, Quan Bai, and Parma Nand. Llm-aidsim: Llm-enhanced agent-based influence diffusion simulation in social networks. *Systems*, 13(1):29, 2025.

[45] Xinnong Zhang, Jiayu Lin, Xinyi Mou, Shiyue Yang, Xiawei Liu, Libo Sun, Hanjia Lyu, Yihang Yang, Weihong Qi, Yue Chen, et al. Socioverse: A world model for social simulation powered by llm agents and a pool of 10 million real-world users. *arXiv preprint arXiv:2504.10157*, 2025.

[46] Shuyan Zhou, Frank F Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, et al. Webarena: A realistic web environment for building autonomous agents. *arXiv preprint arXiv:2307.13854*, 2023.

[47] Lu Zhuo and Dawei Han. Agent-based modelling and flood risk management: A compendious literature review. *Journal of Hydrology*, 591:125600, 2020.

# A Supplementary Materials

## A.1 Additional Figures

## A.2 Data Sources


Figure A1: Census data summary showing age-gender-occupation distributions across the finest-resolution census units (level 2 + level 4) in Higashihiroshima. The tabulations are used as a reference for evaluating demographic accuracy of the synthetic population.

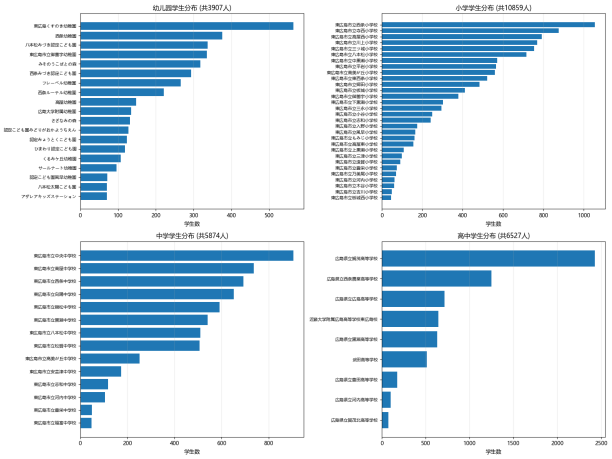



Figure A2: School enrollment distribution across 85 schools in Higashihiroshima, showing the number of students assigned to each educational level. The distribution is consistent with official enrollment statistics.

Table 2: Data sources used to instantiate and validate GenWorld in Higashihiroshima. Access column indicates availability: **Open** = publicly available for automatic download; **Reg** = requires free registration; **NR** = non-redistributable (requires user to obtain from original source).

| **Data Type** | **Source** | **Access** | **Description** |
|---|---|---|---|
| Census Data | e-Stat | Open | Age-gender, household, occupation statistics (198 census units) |
| Time Use Survey | e-Stat | Open | National time-use survey tabulations for activity distributions |
| Admin Boundaries | e-Stat | Open | Census tract boundaries for spatial aggregation |
| Buildings | OpenStreetMap | Open | Building footprints with height and area (45,000+ buildings) |
| POI Data | OpenStreetMap | Open | Points of interest (57,000 POIs) |
| Manufacturing POIs | Hiroshima High-Tech Assoc. | NR | Company locations and employee counts (215 facilities) |
| Land Use | Hiroshima DoBOX | Reg | Parcel-level land use classification |
| Elevation | GSI FGD DEM1A | Reg | 1m-mesh digital elevation model |
| Road Network | OpenStreetMap | Open | Road network with hierarchy (15,861 nodes) |
| School Districts | e-Stat | Open | School district boundaries (85 schools) |
| Mobile Phone Data | YJMob100K [40] | NR | Aggregated commuting patterns for validation |

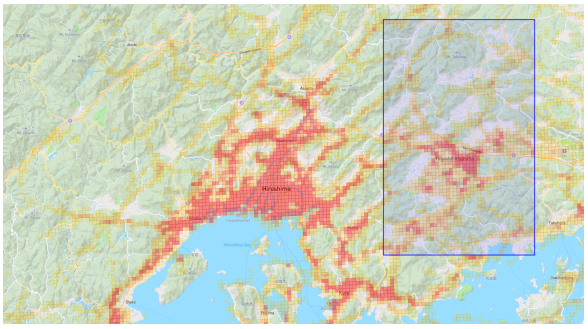


Figure A3: Example of YJMob100K data showing aggregated commuting flows after registering the anonymized mesh grid to our Higashihiroshima study area. The data provides mesh-level origin-destination patterns derived from anonymized mobile phone GPS trajectories, and is used as an external mobility reference.

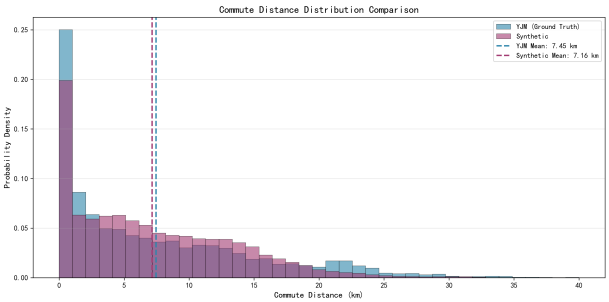
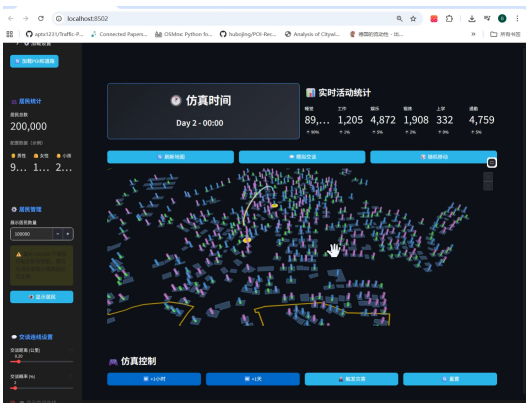
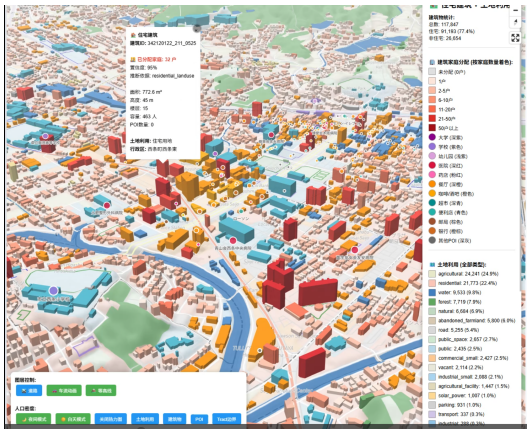



Figure A4: Commute distance distribution comparison between the synthetic population and YJM data, used as a diagnostic for commuting-distance scale.




(a) Simulation dashboard and real-time activity statistics in the Streamlit-based UI.



(b) Interactive building-level map view for inspecting the instantiated urban world (e.g., land use and assigned households). Residential buildings are rendered in red with color intensity proportional to resident counts (darker indicates more residents).

Figure A5: Platform UI screenshots of GenWorld, implemented with Streamlit for interactive inspection and monitoring of the simulation and instantiated urban world.

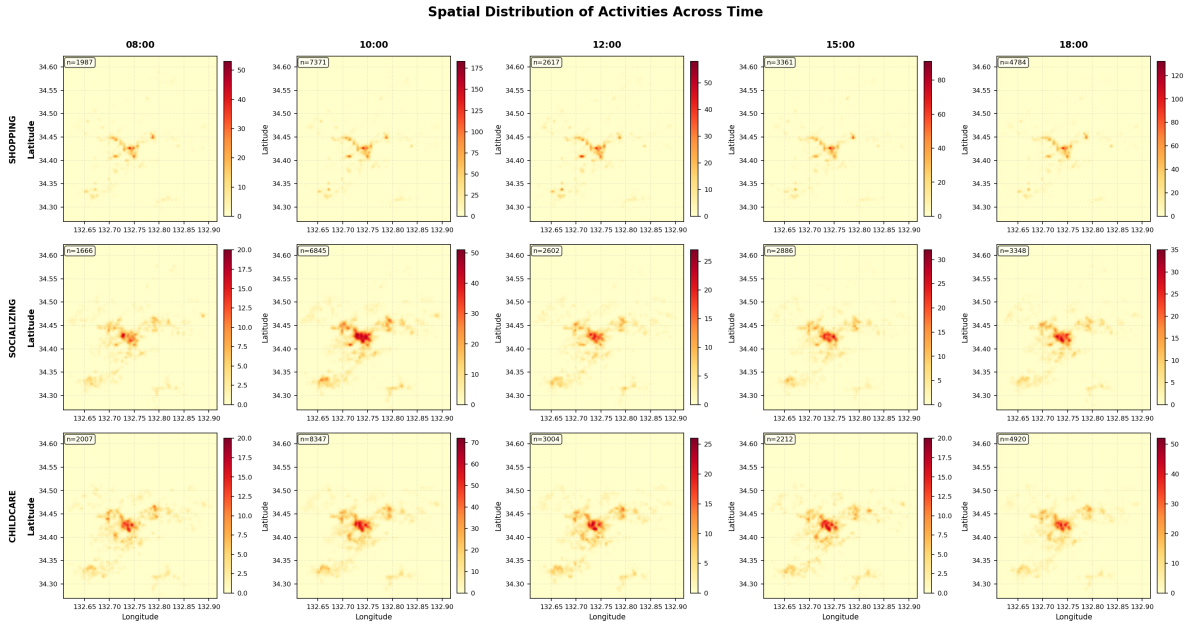



Figure A6: Weekday spatial heatmaps for three representative activity types (shopping, socializing, and childcare) at five time windows. Each row corresponds to an activity type and each column corresponds to a time window; color intensity indicates higher occupancy.

## A.3 Intention and Activity-Type Taxonomy

**Activity Type Vocabulary** We use a small, discrete activity-template vocabulary (configured in `data_prepare/step4_llm_distill/bins_activity_preference.json`) in our reference instantiation:

```
sleep_rest, work_task, study_class, daily_shopping,
↪ personal_service, solo_meal, social_meal,
↪ medical_care, admin_errand, social_visit,
↪ entertainment_activity, structured_exercise,
↪ casual_walk, outdoor_leisure
```

**Distillation Candidate Sets** The same configuration file specifies the intention set $\mathcal{I} = \{\texttt{home}, \texttt{duty}, \texttt{leisure}, \texttt{maintenance}\}$, weekday/weekend intention-chain candidates (`with_duty_intention_chain` and `without_duty_intention_chain`), and the legal mappings `activity`→`intention` and `activity`→`landuse`. These candidate sets define the finite action space used by offline distillation and simulation-time lookup.

Table 3: Reference intention set and allowed activity types used in our instantiation. For each intention $z \in \mathcal{I}$, the teacher scores the predefined candidate set $\mathcal{A}_z$ and we normalize the aggregated scores into a categorical distribution for simulation-time sampling.

| **Intention** $z$ | **Semantics** | **Allowed activity types** $\mathcal{A}_z$ |
|---|---|---|
| `home` | Stay at residence / rest | `sleep_rest` |
| `duty` | Obligations (work/school) | `work_task,study_class` |
| `maintenance` | Daily necessities and errands | `daily_shopping,personal_service,medical_care,admin_errand` |
| `leisure` | Discretionary activities | `solo_meal,social_meal,social_visit,entertainment_activity,structured_exercise,casual_walk,outdoor_leisure` |

## A.4 Distillation Prompt Templates

Below are representative prompt templates for offline distillation. Each query type uses a fixed template that includes resident profile fields and outputs structured JSON scores.

### Chain Scores Prompt

```
Role-play as a resident and score behavior preferences.

Resident: age_bin=<age>, occupation=<occ>
Scenario: typical <day_type>
Candidates: [<chain_1>, <chain_2>, ...]
(H=home, D=duty, L=leisure, M=maintenance)

Task: Score each chain [0-10]. Output JSON only:
{"scores": {"<chain_1>": 5, "<chain_2>": 5}}
```

### Activity Scores Prompt

```
Role-play as a resident and score activity preferences.

Resident: age_bin=<age>, occupation=<occ>
Scenario: pursuing intention='<intention>'
Candidates: [<activity_1>, <activity_2>, ...]

Task: Score each activity [0-10]. Output JSON only:
{"scores": {"<activity_1>": 5, "<activity_2>": 5}}
```

Full templates and configuration files are available in the repository at `data_prepare/step4_llm_distill/`.

## A.5 LLM Interface Schema

This section provides detailed repeatability notes for the LLM-ready interface, including discretization bins, activity–landuse mappings, and missing value handling.

**Context Discretization Bins** Agent context is discretized into coarse bins to enable efficient lookup-table compilation:

- **Age bins** (3 categories): `child` (0–17), `adult` (18–64), `elderly` (65+)
- **Occupation bins** (9 categories): `agriculture_worker`, `industrial_worker`, `service_worker`, `office_worker`, `professional`, `public_sector`, `self_employed`, `non_employed`, `college_student`
- **Day type** (2 categories): `weekday`, `weekend`

**Activity–Intention Mapping** Each activity type maps to exactly one intention category:

| **Activity** | **Intention** |
|---|---|
| `sleep_rest` | `home` |
| `work_task`, `study_class` | `duty` |
| `daily_shopping`, `personal_service`, `medical_care`, `admin_errand` | `maint.` |
| `solo_meal`, `social_meal`, `social_visit`, `entertainment_activity`, `structured_exercise`, `casual_walk`, `outdoor_leisure` | `leisure` |

**Activity–Landuse Mapping** Each activity type is constrained to specific landuse categories (abbreviations: C=commercial, I=industrial, P=public_facility, T=transport, O=open_space, R=residential, A=agriculture, N=nature):

| **Activity** | **Landuse** |
|---|---|
| `sleep_rest` | R |
| `work_task` | C, I, P, T, O, A |
| `study_class` | P |
| `daily_shopping`, `personal_service` | C |
| `medical_care`, `admin_errand` | P |
| `solo_meal` | C, P, T, O |
| `social_meal`, `entertainment` | C, O |
| `social_visit` | R, O |
| `structured_exercise` | O, P |
| `casual_walk` | O, road |
| `outdoor_leisure` | O, N |

**Missing Value Handling** When agent attributes are incomplete, the following defaults apply:

- **Missing occupation**: Mapped to `non_employed` bin
- **Missing age**: Mapped to `adult` bin (modal category)
- **Missing home location**: Agent excluded from spatial activity generation; flagged as `no_location`
- **No valid POI for activity**: Fallback to nearest POI of any compatible landuse type; if none available within search radius, activity skipped

The complete schema files are available in the repository at `data_prepare/step4_llm_distill/bins_*.json`.

## A.6 Distillation Setup

We perform offline distillation by repeatedly querying a teacher model under identical discretized context keys $s$ (Section 4) and estimating empirical action distributions for each decision query type. Prompt templates used for distillation are listed in Appendix A.4.

**Sampling Hyperparameters** In our reference instantiation, we use the following configuration:

- **Repetitions per context key** ($K$): 10 samples per unique $(age\_bin, occupation\_bin, day\_type)$ tuple
- **Teacher model**: Gemma 3 27B [36] served locally via Ollama
- **Temperature**: 0.7 for score generation (enabling diverse but coherent responses)
- **Sampling**: No adaptive sampling; uniform $K$ across all context keys

**Hardware** Distillation was performed on a workstation equipped with an RTX 4090 GPU (24GB VRAM), 96GB RAM, and an Intel Core i5-14600K CPU. The teacher model was queried through AgentScope [10].

**Unseen Key Handling** At simulation time, if a context key $s$ was not encountered during distillation (due to rare demographic combinations), we apply a fallback strategy:

1. **Coarse-bin fallback**: Map the unseen key to a coarser bin (e.g., specific occupation → `non_employed`)
2. **Default distribution**: If no matching compiled distribution exists, use a uniform distribution over the candidate action set

In practice, our discretization yields $3 \times 9 \times 2 = 54$ unique context keys for activity preference queries, which are enumerated during offline compilation.