E^{2}SIM: LONG-HORIZON SIMULATION OF EVERYDAY HUMAN BEHAVIOR WITH EVOLVING STATES

Anonymous authors
Paper under double-blind review

ABSTRACT

AI agents are increasingly expected to support users over long periods, across projects, work, and daily life. This requires modeling not only who users are, but also how their goals, interests, activities, and interactions evolve over time. A central challenge in longitudinal user simulation is maintaining consistency between these evolving states and the consequences of actions in a shared environment. We introduce E^{2}Sim, a longitudinal user simulation framework that jointly models evolving actor states and a dynamic shared environment. E^{2}Sim connects actor-state updates with shared-world changes through a feedback loop linking immediate actions, delayed outcomes, and actor-specific observations across work and life. This allows the consequences of earlier actions to shape later behavior across work and life, while preserving differences in what each actor knows. We further propose a suite of longitudinal trajectory diagnostics tailored to longitudinal simulation, measuring dynamic consistency, coherent event progression and collaboration, and non-degenerate trajectory development rather than static consistency alone. In one 60-day simulation, E^{2}Sim produced more extracted event threads judged meaningful by our evaluator and fewer near-duplicate actions than four direct-generation baselines. Together, the framework and evaluation methods provide a basis for studying longitudinal user simulation, with potential applications in training and evaluating AI agents that assist users over extended periods.

1 INTRODUCTION

Large language model (LLM) agents are being developed as assistants for everyday work and life tasks (Yuan et al., 2024; Wu et al., 2025). A personal assistant that supports the same person over months must reason about multiple projects, stable habits, and evolving goals and interests. Events in one part of a person's life can shape what they do or ask in another, sometimes days or weeks later. An isolated interaction may capture the resulting request while omitting the earlier events that shaped it, making these cross-context effects difficult to observe.

A useful way to study these cross-context effects is to simulate an actor's actions together with the evolving situation in which those actions occur. Task-level agent benchmarks evaluate the completion of specified objectives (Chen et al., 2026b), while personalization benchmarks use extended interaction histories to evaluate responses to a user's changing profile (Jiang et al., 2025). These settings do not directly trace how the consequences of everyday actions change an actor's goals and knowledge, and how those changes shape later actions across work and life. Crucially, accumulated history should influence not only *which* action an actor takes next, but also *how* familiar activities are carried out. A PhD student, for example, may spend years reading papers, designing experiments, and writing, even as the papers they attend to, the experiments they design, the details of their execution, and the outcomes they expect change with accumulated knowledge, goals, and interests. Longitudinal simulation should therefore capture development within recurring activities, not merely variation in the sequence of action labels.

This introduces a challenge that is easy to miss when evaluating isolated steps. A trajectory can remain locally plausible while gradually becoming repetitive, returning to the same routines without meaningful progression, or ceasing to generate consequential new developments. At the same time, simply introducing novelty is not sufficient: new actions must remain consistent with long-term goals, prior outcomes, and what each actor has actually observed. In a multi-actor simulation, an

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

outcome should update an actor's state only when it becomes visible to that actor. For example, an inconclusive experiment may change a researcher's next plan, while a collaborator can respond to the result only after learning about it. Long-horizon simulation therefore requires both *continuity*: later behavior should reflect accumulated history, and *development*: the trajectory should continue to produce grounded, consequential change rather than collapse into repetition.

We study the generation of long-horizon everyday action trajectories from an initial scenario that specifies actors, their relationships, and their shared world. The aim is to follow the same actors across work and life as their goals, plans, and knowledge change, while their stable characteristics remain consistent. A trajectory should show what each actor does, what follows from those actions, and which consequences become visible to others, so that later behavior reflects not only the actor's current situation but the accumulated history that produced it. This requires coupling actor-state evolution with changes in the world: actions should leave persistent consequences, those consequences may unfold immediately or later, and different actors may observe different parts of them.

We introduce E^{2}Sim (*Everyday Simulation with Evolving States*), a framework that centers long-horizon simulation on actor states with stable attributes and changing goals, plans, and knowledge. At each time slot, E^{2}Sim uses an actor's current state and longer-term goals to propose and select work and life goals, generates candidate action paths with different possible outcomes, and applies the realized outcome to both the actor and the shared world. World changes are represented through facts, communication, and delayed follow-up events, and update other actors only when they become visible to them. This actor–environment feedback loop allows immediate and delayed consequences to carry across time and actors while preserving differences in what each actor knows, supporting trajectories in which recurring activities can evolve while remaining grounded in prior events.

To evaluate such trajectories, we introduce a suite of trajectory-based longitudinal evaluation metrics covering dynamic consistency, event progression, collaboration, and non-degeneracy. In a 60-day comparison with four direct-generation baselines, E^{2}Sim exhibits fewer near-duplicate actions and a higher density of extracted event threads judged meaningful by the evaluator.

2 Related Work

Simulation Environments and Frameworks LLM-based simulation frameworks model agents and their interactions with an environment. Generative Agents combines memory, reflection, and planning to simulate routines and emergent group behavior in a shared sandbox (Park et al., 2023). Concordia uses a Game Master to resolve agents' actions and determine their effects on the environment (Vezhnevets et al., 2023). SOTOPIA studies goal-directed social interactions in open-ended scenarios (Zhou et al., 2024), with SOTOPIA-S4 extending the setting to configurable large-scale and multi-party simulation (Zhou et al., 2025), and LIFELONG-SOTOPIA carrying interaction histories across multiple social episodes (Goel & Zhu, 2025). Other systems scale LLM-based behavioral simulation to recommendation and social-network environments (Wang et al., 2025) or large synthetic societies (Piao et al., 2026). Together, these works establish mechanisms for persistent memory, interaction, and environment-mediated simulation.

Our focus. Prior simulation frameworks provide environments in which agents interact. E^{2}Sim makes the *feedback loop between evolving actor state and shared-world state* the primary object of long-horizon simulation.

Human Behavior Simulation and Evaluation Another line of work evaluates human behavior simulation against reference responses and actions. A study of self-report-grounded agents tests whether agents built from interviews and surveys reproduce the same individuals' responses on surveys and economic games (Park et al., 2026). SimBench evaluates population-level fidelity by comparing model and human response distributions across diverse behavioral tasks (Hu et al., 2026). BehaviorChain tests whether models can infer successive behaviors from persona information as a scenario unfolds (Li et al., 2025). OmniBehavior evaluates predictions on long-horizon, cross-scenario behavior traces from real platform logs (Chen et al., 2026a). More recent work develops behavior-specialized foundation models (Zhou et al., 2026) and evaluates long-horizon activity simulation against continuous real-world activity traces (Cheng et al., 2026). These works primarily measure fidelity against observed human responses or future behaviors.

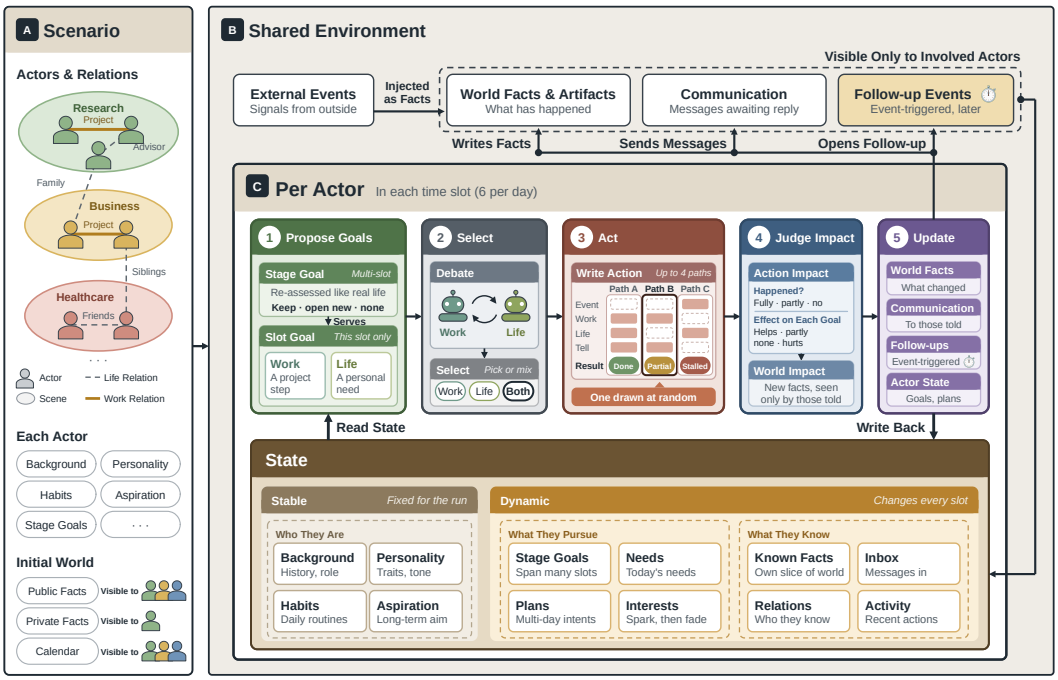


Figure 1: Overview of E^{2}Sim. A scenario (A) initializes actor states and the shared environment (B). At each time slot, actors generate actions from their current states and observations, and the outcomes update both for the next slot (C).

Our focus. Prior behavior-simulation work primarily asks whether generated behavior matches an observed reference. E^{2}Sim asks whether a *generated trajectory remains causally coherent and continues to develop* when no reference future is available.

Dynamic User Modeling and Longitudinal Life Simulation Related work on longitudinal user modeling studies how representations of an individual change with accumulating experience. PERSONAMEM evaluates whether assistants can infer and track evolving user profiles across repeated interactions (Jiang et al., 2025), while DEEPER continually refines user personas from streaming behavioral observations for future behavior prediction (Chen et al., 2025). Recent work on dynamic persona coherence similarly separates stable identity from history-dependent adaptive state in sustained role-playing interactions (QI et al., 2026), and VitaBench 2.0 evaluates agents that must extract, update, and apply evolving user preferences across temporally ordered tasks (Chen et al., 2026c). Most closely related to our setting, LifeSim generates long-horizon life trajectories using a BDI-based cognitive model, physical context, and an event engine, and uses the resulting trajectories to evaluate personalized assistants (Duan et al., 2026).

Our focus. Prior longitudinal user models primarily evolve the individual. E^{2}Sim *jointly evolves the actor and the world*, including shared artifacts, actor-specific observations, and delayed consequences across multiple actors.

See Appendix D for a structured comparison with representative prior work.

3 Simulation Framework

Existing agent simulation frameworks give agents persistent memory and let them interact in shared environments (Park et al., 2023; Vezhnevets et al., 2023). For long-horizon simulation, however, we believe actor states should be modeled in greater detail and kept consistent with the environment. In particular, this motivates us to distinguish two sources of influence on actor states: actors' own actions and influences from the shared world.

Figure 1 summarizes E^{2}Sim: a scenario (A), a shared environment (B), and a per-actor loop (C). The scenario initializes actor states and the world. The loop generates actions whose judged outcomes update both, with each actor observing only visible changes.

3.1 Evolving Actor State

In long-horizon simulation, an actor's behavior reflects both enduring characteristics and a changing situation. We therefore represent the state of actor i at time t as S_i^t, with stable and dynamic components (Figure 2). Figure 1A shows how the scenario initializes both components: actor profiles supply stable attributes, while initial *Stage Goals* and actor-visible world facts seed parts of the dynamic state. The dynamic state then evolves across time slots.

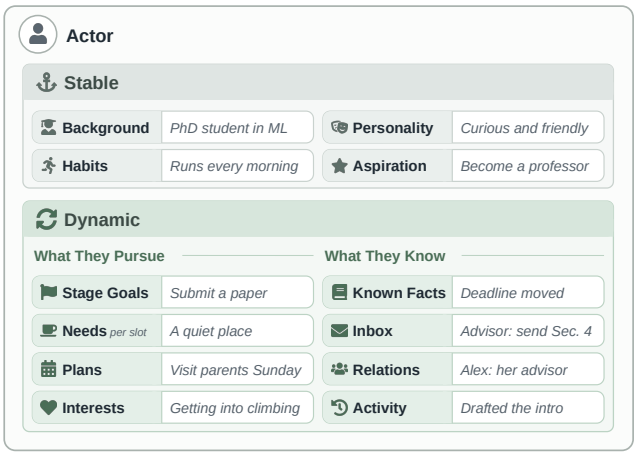


Figure 2: State categories and examples in E^{2}Sim.

The stable state captures attributes that distinguish actors and remain fixed within a simulation run. *Background* describes an actor's history and role. *Personality* describes traits that affect how the actor responds to different situations. *Habits* provide recurring routines, and *Aspiration* gives the actor a long-term direction. For example, a PhD student who prefers to work alone, runs each morning, and aspires to become a professor may keep the morning run and then revise a paper draft independently after receiving advisor comments. An actor with the same background, habits, and aspiration but a personality that favors discussion may first meet the advisor to clarify the comments before revising.

The dynamic state captures the actor's changing objectives and its experience of events and interactions during the simulation. *Stage Goals* specify outcomes the actor works toward over multiple time slots, such as submitting a paper. Completion requires evidence for each specified condition from its designated source, and a newly opened Stage Goal cannot close in the same turn. *Plans* record particular activities it intends to carry out, such as revising a section tomorrow. *Interests* record topics or activities that attract the actor's attention and may later fade. *Needs* represent immediate work or life concerns generated anew for each slot. The state also contains *Known Facts* the actor has learned about the environment, messages received in the *Inbox*, *Relations* with other actors, and *Activity* recording its recent actions. Stage Goals, plans, and interests are reconstructed from append-only event streams.

At the end of each time slot, we update the dynamic state from two sources: the outcome of the actor's own action and changes in the shared world that the actor observes. Its action may advance a *Stage Goal*, change a *Plan*, or enter its *Activity* record. Observed changes in the world may add *Known Facts* or messages to the *Inbox*, and may also lead the actor to revise its goals or plans.

3.2 Actor and Environment Dynamics

Having defined the actor state, we use it as the link between actors and the shared environment. The environment represents the actual outcomes of actions and other changes in the world, while each actor's state retains its characteristics, goals, and knowledge. A change in the world does not mean that every actor observes it or responds in the same way. An actor uses its state and current observation to propose an action, and the simulator then judges whether it occurs and what effects it has. When an outcome or another world change becomes visible to an actor, that observation updates its dynamic state. The updated state then informs later actions, allowing earlier events to continue shaping behavior.

3.2.1 ACTIONS CHANGE THE ENVIRONMENT

Each time slot begins by proposing and selecting goals based on the actor's current state (Steps 1–2 in Figure 1). We derive slot goals that specify reasonable tasks for the current slot because, in preliminary simulations, actors acting directly on multi-day *Stage Goals* compressed months-long research projects into days. Work and life proposers use the actor's state, including its personality, to generate goals for their respective contexts. They then exchange their independently generated proposals and rationales before a selector chooses work, life, or both, allowing the choice to weigh urgent work against the need for rest.

Given the selected slot goal, the actor generates several candidate action paths with different target outcomes and samples one uniformly (Step 3 in Figure 1). We use random selection because the actor favored positive outcomes even when prompted to consider setbacks, and a separate model used to choose among paths showed the same preference in preliminary runs. The selected path then passes to a judge (Step 4 in Figure 1), which first checks whether the action happened fully, partly, or not at all. The judge next assesses its effect on the actor's *Stage Goals* and uses both assessments to determine what changes in the shared world. Only materialized changes update world facts, while anticipated outcomes remain separate until observed.

The simulator then applies the judged outcome (Figure 1, Step 5). Depending on what occurred, the update can revise world facts and affected actors' dynamic states, send messages to relevant actors, and schedule follow-up events for consequences that arise later. For example, an agreement today to travel next week can update the actors' plans now and schedule a follow-up event for the trip.

3.2.2 ENVIRONMENT CHANGES SHAPE ACTIONS

The environment influences later actions through each actor's dynamic state. We distinguish two sources of environmental change: the effects of earlier actor actions and external events introduced independently of those actions. In both cases, actors update their states from the changes they observe, and the updated states guide their next actions.

Within the action loop, a judged action outcome may update world facts, send communication, or schedule follow-up events (Step 5 in Figure 1). World facts record changes to shared objects and conditions, communication conveys information to the actors involved, and follow-up events make delayed consequences available later. As these effects become visible to an actor, they update the relevant parts of its dynamic state, such as known facts, goals, or plans. The actor's next action is then based on this updated state.

Other changes in everyday life are not caused by the actors' previous actions and may not arise from this loop alone. We therefore introduce an external event component that injects random events into the shared world and broadcasts them to the actors. Each actor assesses an event in light of its own state, which may change what it knows or plans and may also create a new *Interest*. For example, after a concert is announced, an actor with a background in music and art may add the concert to its *Interests*, seek more information through later life actions, and eventually decide to attend.

3.3 MULTI-ACTOR INTERACTION

The preceding sections describe how an actor's actions change the environment and, through observation, its own state. Everyday activities also involve other actors who may affect the same project or situation without having the same information. E^2Sim therefore maintains one shared environment but a separate state and observation for each actor. An actor's action can affect another actor when its outcome becomes visible to that actor and updates its dynamic state.

Shared projects and artifacts allow the results of different actors' actions to accumulate in the environment. For example, when one actor revises a paper, the revision remains in the shared document, allowing a collaborator to adjust its plan once it observes the change.

Messages pass through the environment and enter each intended recipient's *Inbox* when delivered. Collaboration or conflict may also change an actor's *Relations*, which record its own view of other actors. Participants can therefore differ in what they know and how they view one another.

When several actors act in the same time slot, each generates its action from a frozen start-of-slot environment snapshot and its own state. The simulator then resolves their proposed actions and makes the resulting changes available after the slot. This prevents an actor from responding to another actor's uncommitted action simply because that action was generated earlier.

3.4 Simulation Loop

E^2Sim initializes actor states and a shared environment from the scenario, then carries both across time slots. Transition records specify the source, activation time, effect, and visibility of changes from actions or external events.

Each slot begins with due follow-up events and new external events. Actors observe available changes and update their dynamic states before acting. The simulator then freezes the environment, so each actor generates behavior from the same world state but its own state and observation. Each actor proposes work and life slot goals, selects what to pursue, and generates candidate action paths. Goal proposers receive stable actor and scenario context, a chronological event ledger, and the current-slot delta through task-specific views. The simulator samples one path and judges its outcome (Steps 1–4 in Figure 1).

The simulator resolves judged outcomes centrally, updating world facts and actor states, delivering messages, and scheduling follow-up events (Step 5 in Figure 1). If multiple actors mark the same shared *Stage Goal* complete within a slot, it arbitrates these completion commits before merging updates. Goal openings and progress updates are then merged before removals, goal events are rebased, and project views recomputed. Actors observe the resulting changes according to their visibility in the next slot.

4 Longitudinal Evaluation

To assess whether trajectories remain coherent and continue to develop, we introduce longitudinal metrics for dynamic consistency, event progression, collaboration, and non-degeneracy. A model-based evaluator scores eligible actor transitions, event threads, and collaboration episodes. Eligibility and coverage appear in Appendix Table 2. These scores characterize coherence within the simulated world, not fidelity to real individuals or effectiveness in downstream assistant tasks.

Individual Dynamic Consistency (IDC). IDC asks whether changes in communication, behavior, goals, and interests follow from an actor's prior experience. It also identifies unsupported changes, missed adaptations, and spillover to unrelated facets. IDC Macro averages alignment over eligible transitions within actors and then across actors on a 0–100 scale. Higher values indicate stronger support from history. Lower Missed Adaptation Rate and Spillover Rate indicate fewer corresponding errors.

Emergent Event Evaluation (EEE). With no predefined event list, we extract action threads around a goal, problem, or shared activity. EEE asks whether each thread has a grounded start (Event Initiation Plausibility, EIP), progresses consistently with earlier outcomes and elapsed time (Event Progression Coherence, EPC), and has a status supported by its history (Event Status Appropriateness, ESA). A thread may properly remain open or waiting. All three scores use a 0–100 scale over eligible threads, with higher values indicating stronger support.

Event Collaboration Dynamics (ECD). A shared event may progress coherently overall while a participant forgets a commitment or acts before a dependency is met. ECD evaluates each decision against the participant's role, commitments, event state, and other actors' steps. Acting and waiting may both be appropriate. Episode Collaboration Plausibility (ECP) summarizes coordination on a 0–100 scale, with higher scores indicating better coordination. Context-Sensitive Response Calibration (CSRC) checks whether responses vary appropriately across situations. ECD measures coordination within shared events, not general relationship quality.

Non-degeneracy Diagnostics (NDD). High scores on eligible transitions can coexist with inactive or repetitive trajectories. NDD counts meaningful event threads per 1,000 actor-slots (one actor in

Table 1: Longitudinal results from one 60-day run per method. Event quality scores cover eligible extracted threads or episodes. Bold marks the best value per metric.

Group	Metric	Scale	E^{2}Sim	Direct	Guided	Per-Actor	Summary
IDC	IDC Macro ↑	0–100	52.06	**56.51**	52.87	54.24	47.75
	Missed Adaptation Rate ↓	%	**27.4**	33.4	35.6	29.1	33.9
	Spillover Rate ↓	%	54.0	47.5	48.9	**47.2**	56.2
ECD	Collaboration Plausibility ↑	0–100	93.25	**95.92**	93.00	95.24	92.76
EEE	Event Initiation Plausibility ↑	0–100	**87.74**	77.34	83.06	85.94	85.34
	Event Progression Coherence ↑	0–100	88.86	71.33	84.58	**90.10**	84.57
	Event Status Appropriateness ↑	0–100	**94.85**	82.14	92.19	93.06	94.23
NDD	Near-Duplicate Actions ↓	%	**22.3**	73.5	62.8	60.5	71.0
	Meaningful Event Threads ↑	per 1k actor-slots	**21.4**	3.9	10.3	8.3	6.4

one time slot) and measures near-duplicate actions. It also checks whether new threads are grounded in context, events have later consequences, and decisions use relevant history. Lower duplication is preferable, whereas event density has no universal target and must be read with the other diagnostics.

Evaluation reliability. A post-hoc judge may reward an observed response based on a plausible explanation constructed afterward. We require trajectory evidence and test the judge on injected errors, including forgotten commitments and violated dependencies. Perturbation Win Rate (PWR) is the share of original–corrupted episode pairs for which the judge scores the original higher. Higher PWR indicates better detection of these errors, not agreement with human ratings. Appendix A details the rubrics, extraction procedure, and aggregation rules.

5 Experiment

We generate 60-day action trajectories for ten actors across four connected settings, with six time slots per day. Each trajectory contains 3,600 actor-slot action records. We run E^{2}Sim and each of four baselines once from the same scenario file. All five generation runs use `gpt-5.6-luna`. Appendix C details the actor roster, initial goals, and world configuration.

We compare E^{2}Sim with four direct-generation baselines. **Direct** prompts the model once per day with the full scenario and preceding trajectory to generate actions for all actors. **Guided** retains this setup but adds instructions targeting errors observed in Direct, testing how far prompting alone can address them. **Per-Actor** generates each actor separately using its own profile, visible initial facts, own history, and other actors' entries that mention it. **Summary** retains joint daily generation but replaces the full history with a summary maintained by the model and the most recent three days, examining the effect of a more compact history. We evaluate the five exported trajectories using the longitudinal measures in Section 4.

We use `gpt-5.6-luna` with medium reasoning effort as the evaluation judge. Because event quality scores require eligible extracted threads or episodes, we report their coverage in Appendix A.3. The extraction and scoring procedures are detailed in Appendix A.

5.1 Baseline Comparison

Table 1 compares the complete 60-day trajectories. E^{2}Sim has a higher density of extracted threads judged meaningful by our evaluator than any baseline (21.4 versus 3.9–10.3 per 1,000 actor-slots) and a lower near-duplicate rate (22.3% versus 60.5–73.5%). It also receives the highest event initiation and status scores, while Per-Actor scores slightly higher on event progression. The consistency results are mixed: Direct has higher IDC and collaboration plausibility, and three baselines have lower spillover rates. Thus, the clearest observed differences are in the density of extracted threads judged meaningful and in repetition, rather than a uniform improvement across all measures.

The event-based scores require additional care because the thread extractor links 1,276 of E^{2}Sim's 3,600 actions, compared with 1,906–2,298 actions for the baselines. The event quality scores therefore describe different eligible subsets (Appendix A.3), while meaningful-thread density also de-

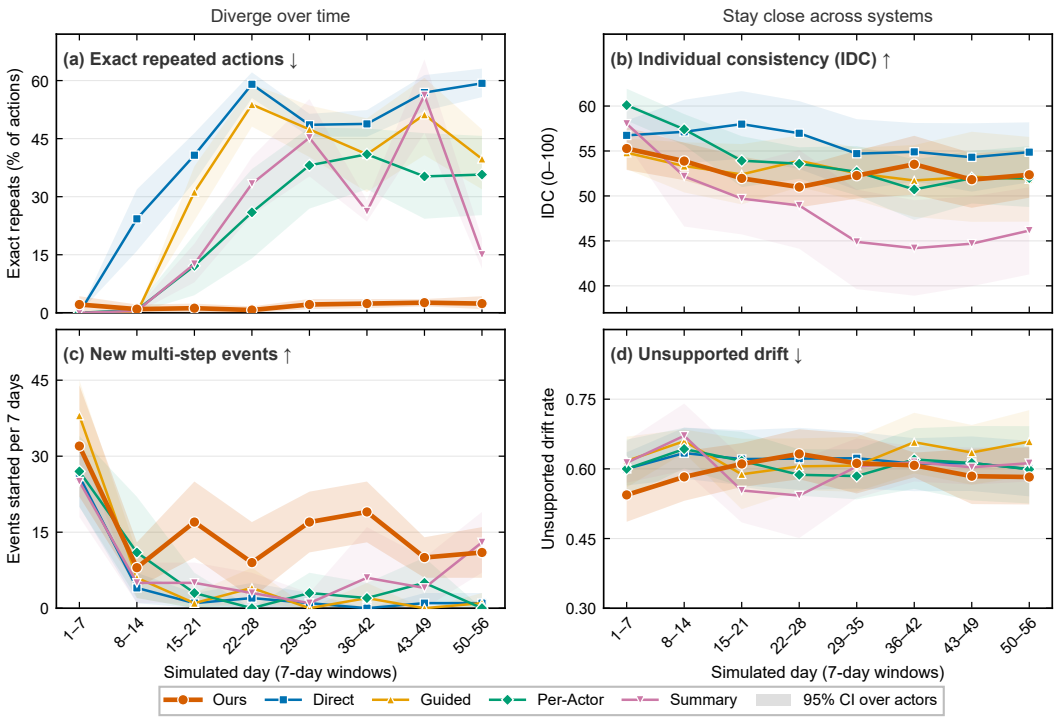


Figure 3: Trajectory measures across days 1–56. Shading shows 95% bootstrap intervals over actors. Exact repeats in (a) differ from judged near-duplicates in Table 1.

pends on how the extractor groups actions. Each method was run once, so the comparison describes the observed trajectories without estimating variation across independent runs.

5.2 Long-Horizon Trajectory Dynamics

The aggregate comparison hides how repetition and event activity change over time. We divide days 1–56 of each run into eight nonoverlapping seven-day windows and track exact action repetition, new multi-step events, individual dynamic consistency (IDC), and unsupported drift (Figure 3). Intervals resample the ten actors and do not capture variation across independent simulation runs.

Exact repetition separates the systems over time (Figure 3a): it stays around 1–3% for E^{2}Sim but rises to roughly 35–60% for the four baselines in later windows. The estimated increase is 0.18 percentage points per window for E^{2}Sim, versus 5.19–7.05 for the baselines. Thus, the gap grows over time rather than appearing only in the aggregate.

The event-start series suggests a related pattern (Figure 3c). All systems begin with many events tied to initial goals, but E^{2}Sim continues to start multi-step events in later windows while most baselines start few. IDC and unsupported drift remain relatively close (Figure 3b,d), with Direct maintaining higher IDC than E^{2}Sim. The clearest longitudinal difference is sustained activity with less exact repetition, not a uniform gain in state-transition scores.

5.3 Component Analysis

From the day-60 checkpoint, we run one 14-day continuation for the full system and each ablation, removing the random draw among candidate paths, external events, or new *Stage Goal* creation. The branches use the same model and otherwise identical code, with switches verified in invocation logs. Figure 4 shows trajectories and matched actor-time examples. One continuation per condition cannot estimate run-to-run variation.

Without the outcome draw, the share of actions not realized as stated falls from 49% to 19%, while exact repetition rises from 2.0% to 8.0% (Figure 4a). In matched examples, all three intended

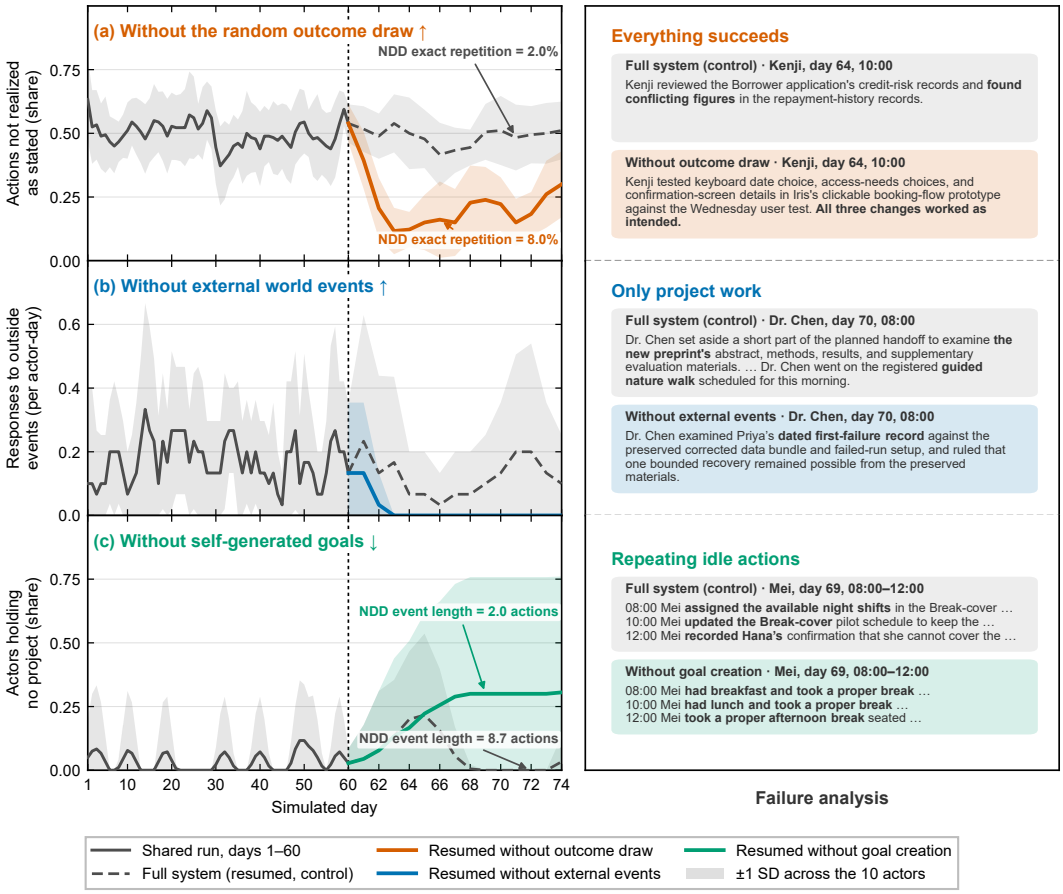


Figure 4: Ablations over 14 days from the same day-60 state. Shading shows one standard deviation across ten actors. Examples show the full system and each ablation.

changes succeed without the draw, whereas the control encounters conflicting records. This is consistent with the favorable-outcome tendency observed in preliminary runs.

Without external events, recorded responses to them fall to zero by construction (Figure 4b). The control responds to a new preprint and a planned walk, while the matched ablation continues project work. This changes action content, although our evaluation suite has no direct metric for that effect.

Without new *Stage Goals*, 30% of actor-slots lack an active project on days 68–74, versus 1% in the control (Figure 4c). The lag suggests existing goals briefly sustain work. Average new-thread length falls from 8.7 to 2.0 actions, and Mei shifts from break-cover coordination to repeated breaks. IDC rises from 53.6 to 58.5, showing that local consistency does not ensure sustained activity.

Two independent LLM readers per ablation compared the 14-day traces under randomized A/B labels, and all six identified the ablated trace. Because they knew the removed component, this check measures distinguishability rather than human-rated quality.

6 Conclusion

We introduced E^{2}Sim, which preserves stable actor characteristics while updating goals, plans, and knowledge as actions change a shared world and their outcomes become visible. We also proposed longitudinal measures of state consistency, event development, collaboration, and repetition. In one 60-day run, E^{2}Sim produced a higher density of extracted threads judged meaningful by our evaluator and fewer near-duplicate actions than four direct-generation baselines. Component analyses show how outcome sampling, external events, and new goals affect the continuation of activity.

AI USE STATEMENT

In this work, we used generative AI tools as core components of the proposed simulation framework, the baseline systems, and the model-based evaluation pipeline, as described in the main text and appendix. We also used generative AI tools for polishing the writing of the paper. We have reviewed all AI-assisted work. AI-assisted code and generated outputs were manually inspected and validated where applicable. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work studies simulated longitudinal human behavior and does not involve human subjects or private personal data. The generated trajectories are synthetic and are intended for research on long-horizon user simulation and evaluation.

Because such simulations may appear behaviorally plausible, they should not be interpreted as faithful representations of real individuals or used to make high-stakes decisions about people. In particular, our evaluation measures coherence within the simulated world rather than fidelity to real human behavior. Future applications of longitudinal user simulation should therefore consider risks related to overgeneralization, stereotyping, privacy, and inappropriate reliance on simulated behavior in sensitive settings.

REPRODUCIBILITY STATEMENT

We provide the information needed to reproduce the simulation and evaluation protocol across the main text and appendices. Section 3 specifies E^2Sim's actor-state representation, actor–environment transition mechanism, multi-actor visibility, and simulation loop, while Section 4 defines the longitudinal evaluation metrics and their interpretation. Section 5 reports the experimental setup, including the 60-day, ten-actor scenario, baseline definitions, generation model, and evaluation model. Appendix A documents the evaluation pipeline, formal metric definitions, aggregation procedures, eligibility criteria, and abstention rules. Appendix B.2 describes the adapted SOTOPIA evaluator. Appendix C specifies the actor roster, initial goals, relationships, and world configuration used across conditions.

We additionally report metric coverage and denominators for model-based judgments, and explicitly note that each main comparison condition is represented by a single trajectory, so the reported results should not be interpreted as estimates of run-to-run variability.

REFERENCES

Aili Chen, Chengyu Du, Jiangjie Chen, Jinghan Xu, Yikai Zhang, Siyu Yuan, Zulong Chen, Liangyue Li, and Yanghua Xiao. DEEPER insight into your user: Directed persona refinement for dynamic persona modeling. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 24157–24180, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.1177. URL https://aclanthology.org/2025.acl-long.1177/.

Jiawei Chen, Ruoxi Xu, Boxi Cao, Ruotong Pan, Yunfei Zhang, Yifei Hu, Yong Du, Tingting Gao, Yaojie Lu, Yingfei Sun, Xianpei Han, Le Sun, Xiangyu Wu, and Hongyu Lin. Towards real-world human behavior simulation: Benchmarking large language models on long-horizon, cross-scenario, heterogeneous behavior traces, 2026a. URL https://arxiv.org/abs/2604.08362.

Kaiyuan Chen, Qimin Wu, Taiyu Hou, Tianhao Tang, Xueyu Hu, Yuchen Hou, Bikun Li, Chengming Qian, Guoyin Wang, Haolin Chen, Haotong Tian, Haoye Zhang, Haoyu Bian, Hongbing Pan, Hongkang Zhang, Hongyi Zhou, Jiaqi Cai, Jiewu Rao, Jiyuan Ren, Keduan Huang, Lucia Zhu Huang, Mingyu Yuan, Naixu Guo, Qicheng Tang, Qinyan Zhang, Shuai Chen, Siheng Chen, Ting Ting Li, Xiaoxing Guo, Yaocheng Zuo, Yaoqi Guo, Yinan Wang, Yinzhou Yu, Yize Wang, Yuan Jiang, Yuan Tian, Yuanshuo Zhang, Yuxuan Liu, Yvette Yan Zeng, Zenyu Shan, Zihan Yin,

Xiaobo Hu, Yang Liu, Yixin Ren, and Yuan Gong. Agentif-oneday: A task-level instruction-following benchmark for general ai agents in daily scenarios, 2026b. URL https://arxiv.org/abs/2601.20613.

Yuxin Chen, Yi Zhang, Zhengzhou Cai, Yaorui Shi, Zhiyuan Yao, Chenhang Cui, Jingnan Zheng, Yaqi Huo, Xi Su, Qi Gu, et al. Vitabench 2.0: Evaluating personalized and proactive agents in long-term user interactions. *arXiv preprint arXiv:2605.27141*, 2026c.

Yi Fei Cheng, Fan Yang, Iremsu Bas, Koichiro Niinuma, Narishige Abe, and David Lindlbauer. Measuring the behavioral fidelity of long-horizon human activity simulations, 2026. URL https://arxiv.org/abs/2609.01257.

Feiyu Duan, Xuanjing Huang, and Zhongyu Wei. Lifesim: Long-horizon user life simulator for personalized assistant evaluation, 2026. URL https://arxiv.org/abs/2603.12152.

Hitesh Goel and Hao Zhu. Lifelong sotopia: Evaluating social intelligence of language agents over lifelong social interactions. *arXiv preprint arXiv:2506.12666*, 2025.

Tiancheng Hu, Joachim Baumann, Lorenzo Lupo, Nigel Collier, Dirk Hovy, and Paul Röttger. Simbench: Benchmarking the ability of large language models to simulate human behaviors, 2026. URL https://arxiv.org/abs/2510.17516.

Bowen Jiang, Zhuoqun Hao, Young-Min Cho, Bryan Li, Yuan Yuan, Sihao Chen, Lyle Ungar, Camillo J. Taylor, and Dan Roth. Know me, respond to me: Benchmarking llms for dynamic user profiling and personalized responses at scale, 2025. URL https://arxiv.org/abs/2504.14225.

Rui Li, Heming Xia, Xinfeng Yuan, Qingxiu Dong, Lei Sha, Wenjie Li, and Zhifang Sui. How far are LLMs from being our digital twins? a benchmark for persona-based behavior chain simulation. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 15738–15763, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.813. URL https://aclanthology.org/2025.findings-acl.813/.

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*, pp. 1–22, 2023.

Joon Sung Park, Carolyn Q. Zou, Jonne Kamphorst, Niles Egan, Aaron Shaw, Benjamin Mako Hill, Carrie Cai, Meredith Ringel Morris, Percy Liang, Robb Willer, and Michael S. Bernstein. Llm agents grounded in self-reports enable general-purpose simulation of individuals, 2026. URL https://arxiv.org/abs/2411.10109.

Jinghua Piao, Yuwei Yan, Jun Zhang, Nian Li, Junbo Yan, Xiaochong Lan, Zhihong Lu, Zhiheng Zheng, Jing Yi Wang, Di Zhou, Chen Gao, Fengli Xu, Fang Zhang, Ke Rong, Jun Su, and Yong Li. Agentsociety: Large-scale simulation of llm-driven generative agents advances understanding of human behaviors and society, 2026. URL https://arxiv.org/abs/2502.08691.

Yirui QI, Xiaoming Zhang, Ruilin Zeng, Mengyao Liu, Ziyi Zhou, Dezhuang Miao, Bingyu Yan, and Zhenyu Guan. Beyond static persona consistency: Dynamic persona coherence in LLM role-playing. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 28942–28956, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.1336. URL https://aclanthology.org/2026.acl-long.1336/.

Alexander Sasha Vezhnevets, John P. Agapiou, Avia Aharon, Ron Ziv, Jayd Matyas, Edgar A. Duéñez-Guzmán, William A. Cunningham, Simon Osindero, Danny Karmon, and Joel Z. Leibo. Generative agent-based modeling with actions grounded in physical, social, or digital space using concordia, 2023. URL https://arxiv.org/abs/2312.03664.

Lei Wang, Jingsen Zhang, Hao Yang, Zhi-Yuan Chen, Jiakai Tang, Zeyu Zhang, Xu Chen, Yankai Lin, Hao Sun, Ruihua Song, et al. User behavior simulation with large language model-based agents. *ACM Transactions on Information Systems*, 43(2):1–37, 2025.

Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. Longmemeval: Benchmarking chat assistants on long-term interactive memory, 2025. URL https://arxiv.org/abs/2410.10813.

Ruifeng Yuan, Shichao Sun, Yongqi Li, Zili Wang, Ziqiang Cao, and Wenjie Li. Personalized large language model assistant with evolving conditional memory, 2024. URL https://arxiv.org/abs/2312.17257.

Xuhui Zhou, Hao Zhu, Leena Mathur, Ruohong Zhang, Haofei Yu, Zhengyang Qi, Louis-Philippe Morency, Yonatan Bisk, Daniel Fried, Graham Neubig, and Maarten Sap. Sotopia: Interactive evaluation for social intelligence in language agents, 2024. URL https://arxiv.org/abs/2310.11667.

Xuhui Zhou, Zhe Su, Sophie Feng, Jiaxu Zhou, Jen-tse Huang, Hsien-Te Kao, Spencer Lynch, Svitlana Volkova, Tongshuang Wu, Anita Woolley, Hao Zhu, and Maarten Sap. SOTOPIA-s4: a user-friendly system for flexible, customizable, and large-scale social simulation. In Nouha Dziri, Sean (Xiang) Ren, and Shizhe Diao (eds.), *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (System Demonstrations)*, pp. 350–360, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-191-9. doi: 10.18653/v1/2025.naacl-demo.30. URL https://aclanthology.org/2025.naacl-demo.30/.

Xuhui Zhou, Weiwei Sun, Weihua Du, Jiarui Liu, Haojia Sun, Qianou Ma, Tongshuang Wu, Yiming Yang, and Maarten Sap. Odyssim: Building foundation models for human behavior simulation. *arXiv preprint arXiv:2606.14199*, 2026.

A Longitudinal Evaluation: Implementation Details

This appendix documents the reference implementation of the evaluation framework in Section 4.

A.1 Pipeline overview

The pipeline operates on a parsed simulation trajectory (per-actor, per-time-slot records) and proceeds in four stages.

Event-thread extraction. Atomic events are extracted from the raw trajectory, then linked into candidate threads using shared actors, shared objects or goals, explicit references, causal compatibility, and temporal plausibility. Linking allows an event to be assigned to a `NEW_THREAD` or left as `UNCERTAIN` rather than forcing every event into an existing thread. Each resulting thread is captured as an event card recording its actors, type, initiation time, core object, state sequence, current status, open dependencies, and supporting evidence. EEE and ECD are both computed from these cards.

Judge interface. Scoring is abstracted behind a common `Judge` interface, and every run reported in this paper uses `LLMJudge` rather than the deterministic `HeuristicJudge` kept for offline unit tests. This means every *semantic label* – Pass A/B transition labels, ECD episode dimensions, EEE dimensions, CSRC dimensions, and the per-item NDD decisions – is produced by the judge model, with response caching, retry handling, and batched/threaded requests for throughput. It does not mean every *number* in the report is itself a model output: taxonomy-mode Transition Alignment Scores are a fixed deterministic lookup applied to the Pass A/B labels (Appendix A.2), and all rates, counts, and descriptive statistics are computed from judged labels in Python rather than requested from the model directly.

Per-block scoring. IDC scoring follows the three-pass protocol of Appendix A.2. ECD scoring computes the five episode dimensions and the Perturbation Win Rate against six corruption operators (unrelated response, dependency violation, commitment amnesia, premature closure, mechanical overreaction, mechanical inertia), each applied at the episode level. EEE scoring computes

Event Initiation Plausibility, Event Progression Coherence, and Event Status Appropriateness per Event Card. A dedicated LLM diagnostics stage judges the Context-Sensitive Response Calibration dimensions and the per-item decisions underlying all five NDD diagnostics (Appendix A.2) using earlier judge outputs and retrieved downstream evidence. Deterministic lexical versions of both exist in the code as legacy scaffolding for the offline `HeuristicJudge` path, but every reported CSRC and NDD number aggregates the LLM-judged version.

Abstention and reporting. Per-item judgments the pipeline cannot support with sufficient evidence (e.g. INSUFFICIENT_EVIDENCE Pass A/B labels, undecidable NDD windows) are quarantined rather than merged or imputed. Available eligibility denominators are reported in Table 2. Unavailable counts are marked NR. Scores remain conditional on eligibility, and differing abstention patterns can affect comparisons. One exception: IDC Macro and the three pooled IDC rates – Unsupported Drift Rate, Missed Adaptation Rate, and Spillover Rate – return 0.0 rather than N/A by convention when the relevant denominator is empty. The denominator counts eligible actor–facet transitions for IDC Macro, eligible STAY cases for Unsupported Drift Rate, eligible SHOULD_CHANGE cases for Missed Adaptation Rate, and triggered transitions for Spillover Rate. The denominator is reported alongside each value, so a 0.0 next to a denominator of zero should be read as "not evaluated" rather than "no failures observed" (for the rates) or "perfectly consistent" (for IDC Macro). A final aggregation step assembles all blocks into a single report, which we render as the tables in this appendix.

A.2 Formal metric definitions

Individual Dynamic Consistency (IDC). An actor's situation can change without every aspect of its behavior changing with it. IDC assesses whether transitions in communication style, behavioral patterns, goals and commitments, and interests and preferences are supported by intervening events. It also checks whether changes spread to unrelated facets or whether the actor fails to adapt when the accumulated context provides strong reason to do so. Each eligible actor–facet transition receives a Transition Alignment Score (TAS) from 0 to 4. We report the average across actors as

$$\text{IDC Macro} = \frac{100}{|\mathcal{A}|} \sum_{i \in \mathcal{A}} \frac{1}{|\mathcal{U}_i|} \sum_{u \in \mathcal{U}_i} \frac{s_u}{4}, \tag{1}$$

where i denotes an actor, $\mathcal{A}$ is the set of actors with at least one eligible transition, $\mathcal{U}_i$ is the set of eligible actor–facet transitions for actor i, u denotes one such transition, and $s_u \in \{0, 1, 2, 3, 4\}$ is its alignment score. The vertical bars give the number of elements in each set, and the factors $1/4$ and 100 convert the score to a 0–100 scale. A higher IDC Macro means that an actor's observed stability and changes are more consistently supported by its history.

To obtain s_u, an expected-update pass sees the pre-event state and an approximate trigger clause extracted from the current action, and labels the facet STAY, MAY_CHANGE, SHOULD_CHANGE, or INSUFFICIENT_EVIDENCE. A separate observed-update pass sees the pre/post trajectory and labels the observed change NONE, MINOR, MATERIAL, or INSUFFICIENT_EVIDENCE, with supporting evidence. Both passes query the judge. A deterministic lookup converts the two labels into TAS for STAY and SHOULD_CHANGE, and for MAY_CHANGE when the taxonomy mode is used. The alternative mode queries the judge for the plausibility of a realized MAY_CHANGE transition. The trigger clause is split from action text using predefined connective words. If no split is found, the full action is used for both trigger and response, so the expected-update pass is not guaranteed to exclude response information.

We also report Unsupported Drift Rate (UDR) for material changes without sufficient support, Missed Adaptation Rate (MAR) for failures to update when the context strongly supports adaptation, and Spillover Rate for unjustified changes in unrelated facets:

$$\text{UDR} = \frac{\#\text{unsupported material changes}}{\#\text{eligible STAY cases}},$$

$$\text{MAR} = \frac{\#\{\text{SHOULD_CHANGE cases with observed change NONE or MINOR}\}}{\#\text{eligible SHOULD_CHANGE cases}}.$$

Here $\#$ denotes a count. Spillover Rate divides the number of triggered transitions with an unjustified change in an unrelated facet by the number of triggered transitions eligible for that judgment.

Lower values are better for all three rates. UDR, MAR, and Spillover Rate pool eligible cases across actors, whereas IDC Macro first averages within actors. MAR measures the magnitude of an observed update, not whether a material update moved in the correct direction. The taxonomy TAS for SHOULD_CHANGE has the same limitation.

Because many triggers admit multiple reasonable continuations, the boundary between MAY_CHANGE and SHOULD_CHANGE can be unreliable. Once per evaluation, the pipeline runs the expected-update pass on a sampled batch of 24 transitions by default and measures the share of decidable facet labels that are MAY_CHANGE. If this share is at least 50%, MAY_CHANGE facets use the realized-transition-plausibility judge rather than the fixed taxonomy lookup for the rest of the run. Otherwise, the taxonomy lookup is retained. The STAY and SHOULD_CHANGE labels always use the deterministic lookup. The report records the mode selected for each run.

Emergent Event Evaluation (EEE). The simulator does not follow a fixed list of events to complete, so we identify event threads from actions concerning a continuing goal, problem, or shared activity. EEE evaluates three questions about each thread. Event Initiation Plausibility (EIP) asks whether its beginning is grounded in the actor's state or the world. Event Progression Coherence (EPC) asks whether later developments respect prior outcomes, dependencies, and elapsed time. Event Status Appropriateness (ESA) asks whether its current status, including remaining open or waiting, follows from its history. This last measure avoids treating rapid completion as inherently better than a justified ongoing process. Each measure averages its applicable rubric scores and rescales them to 0–100:

$$Q_m(e) = \frac{100}{4|D_m(e)|} \sum_{d \in D_m(e)} s_{e,d}, \qquad m \in \{\mathrm{EIP}, \mathrm{EPC}, \mathrm{ESA}\}, \tag{2}$$

where e denotes an event thread, m identifies one of the three EEE measures, and $Q_m(e)$ is that measure's score for thread e. The set $D_m(e)$ contains the applicable rubric criteria, d denotes one criterion, and $s_{e,d} \in \{0, 1, 2, 3, 4\}$ is its score. The vertical bars give the number of applicable criteria. Higher EIP, EPC, and ESA scores indicate stronger support for the event's initiation, progression, and current status, respectively. EIP assesses contextual grounding, actor compatibility, consistency with established history, and potential future consequences. EPC assesses causal coherence, state continuity, timing, and dependency handling. ESA assesses whether the status and, for terminated events, the resolution fit the thread's history. We average eligible thread scores for each measure.

Event Collaboration Dynamics (ECD). An event thread can progress coherently overall while one participant forgets a commitment or acts before another has completed a required step. ECD therefore evaluates decision episodes within multi-actor threads. For each episode, it considers the current event state, the focal actor's role and commitments, the relevance of its response to the trigger, dependencies among participants, and whether the response remains consistent with subsequent steps. Acting and waiting can both be appropriate when supported by the actor's information and the event history. We summarize these five dimensions with Episode Collaboration Plausibility (ECP):

$$\mathrm{ECP}(c) = \frac{100}{4|K_c|} \sum_{k \in K_c} r_{c,k}, \tag{3}$$

where c denotes a collaboration episode, K_c is the set of dimensions with sufficient evidence, k denotes one such dimension, and $r_{c,k} \in \{0, 1, 2, 3, 4\}$ is its score. The vertical bars give the number of scored dimensions. A higher ECP score means the participant's response better respects the shared event state, commitments, dependencies, and subsequent trajectory. We average episode scores within each event thread before averaging across threads, so a long event does not dominate the result. Context-Sensitive Response Calibration (CSRC) additionally checks whether responses differ appropriately across episodes with different triggers, accumulated evidence, commitments, dependencies, and actor histories. A higher CSRC score indicates more appropriate context sensitivity. ECD measures coordination within shared events, not general interpersonal relationship quality.

Non-degeneracy Diagnostics (NDD). Strong scores on eligible transitions and events do not ensure that a full trajectory develops meaningfully. A simulator might avoid inconsistencies by producing almost no consequential activity, repeat similar interactions without lasting effects, or con-

tinually introduce weakly grounded events. NDD therefore reports five complementary diagnostics. Meaningful Event-Thread Rate counts judge-confirmed meaningful threads per $1{,}000$ observed actor-slots, where one actor-slot is one actor in one simulation time slot. A low rate can indicate stagnation, but a high rate alone is not evidence of quality. Supported Novelty Ratio is the share of newly initiated threads whose grounding and actor-compatibility scores are both at least 3 out of 4. Repetition/Near-Duplicate Rate is the share of decision windows that repeat an earlier pattern without a new consequence. Consequence Rate is the share of meaningful events with a downstream effect found in later observations. History-Dependence Rate is the share of decisions with relevant prior history that use it appropriately. We compute the five diagnostics as follows, using MTR for Meaningful Event-Thread Rate, SNR for Supported Novelty Ratio, RNR for Repetition/Near-Duplicate Rate, CR for Consequence Rate, and HDR for History-Dependence Rate:

$$\begin{aligned}
\mathrm{MTR} &= 1000\,\frac{\#\text{meaningful threads}}{\#\text{observed actor-slots}},\\
\mathrm{SNR} &= 100\,\frac{\#\text{supported new threads}}{\#\text{eligible new threads}},\\
\mathrm{RNR} &= 100\,\frac{\#\text{repeated windows}}{\#\text{decidable windows}},\\
\mathrm{CR} &= 100\,\frac{\#\text{threads with downstream effects}}{\#\text{consequence-eligible threads}},\\
\mathrm{HDR} &= 100\,\frac{\#\text{decisions appropriately using history}}{\#\text{history-eligible decisions}},
\end{aligned} \tag{4}$$

where $\#$ denotes a count. An eligible new thread is judged to have begun within the observed window and has both grounding and actor-compatibility scores. It is supported when both scores are at least 3 out of 4. The repetition denominator contains windows for which repetition can be judged. Consequence-eligible threads are meaningful threads for which a later effect can be assessed. History-eligible decisions have relevant prior history and a decidable judgment of whether the decision uses it appropriately. MTR is a density per $1{,}000$ actor-slots, while the other four measures are percentages. Higher SNR, CR, and HDR and lower RNR are preferable. MTR has no universal target. We interpret the measures together and do not assume a universal target frequency for events or state changes. Undecidable cases are excluded from the relevant denominator, and we report eligibility alongside each rate in Appendix A.3.

Evaluation reliability. Post-hoc judgments risk accepting an observed response merely because a plausible explanation can be constructed for it. We require the judge to ground its scores in trajectory evidence and compare original collaboration episodes with controlled corruptions, including forgotten commitments and violated dependencies. Perturbation Win Rate (PWR) records how often the judge scores the original episode higher. Following the implemented aggregation, we average these comparisons within threads, then across threads for each corruption type, and finally across corruption types:

$$\begin{aligned}
\mathrm{PWR} = \frac{1}{|\mathcal{C}|}\sum_{c\in\mathcal{C}}\frac{1}{|\mathcal{T}_c|}\sum_{t\in\mathcal{T}_c}\\
\times\frac{1}{|\mathcal{P}_{c,t}|}\sum_{j\in\mathcal{P}_{c,t}}\mathbf{1}[q(x_j) > q(\tilde{x}_j)]\,,
\end{aligned} \tag{5}$$

where $\mathcal{C}$ is the set of eligible corruption types, $\mathcal{T}_c$ is the set of eligible event threads for type c, and $\mathcal{P}_{c,t}$ is the set of eligible original–corrupted episode pairs in thread t for type c. The index j identifies a pair, x_j is its original episode, $\tilde{x}_j$ is its corrupted version, and q is the judge's score on dimensions available for both. The indicator $\mathbf{1}[\cdot]$ is 1 when the original scores higher and 0 otherwise, so ties are not wins. PWR ranges from 0 to 1. A higher value indicates that the judge better distinguishes coherent episodes from controlled errors. It checks the evaluator, not the quality of a generated trajectory.

Table 2: Coverage and available eligibility denominators for the five longitudinal evaluations. NR means not reported in the available Markdown export. It does not mean zero or ineligible.

Evaluation unit or denominator	E^2Sim	Direct	Guided	Per-Actor	Summary
Observed actor-slots	3600	3600	3600	3600	3600
Accepted threads	84	25	45	36	28
IDC eligible STAY cases	5696	4223	NR	NR	7185
IDC eligible SHOULD_CHANGE cases	3373	5467	NR	NR	3249
IDC triggered transitions	1909	2940	NR	NR	1922
ECD scorable threads	26	5	13	14	9
ECD decision episodes	1550	339	871	1898	1678
CSRC eligible actors	10	10	10	10	10
EIP eligible cards	79	24	45	36	28
EPC eligible cards	84	25	45	36	27
ESA eligible cards	51	7	16	18	13
Meaningfulness-decidable threads	77	14	37	31	25
Confirmed meaningful threads	77	14	37	30	23
Novelty-eligible threads	50	4	13	14	10
Consequence-eligible threads	68	14	37	30	23
Repetition-decidable windows	2659	3127	3423	3148	3041
History-use-decidable decisions	2621	3105	3400	3133	2995

A.3 AGGREGATION AND COVERAGE ACROSS CONDITIONS

Each condition in Table 1 contributes one trajectory. No metric is averaged across independent simulation runs. The within-trajectory aggregation procedures are:

- **IDC Macro:** average eligible transition–facet TAS values within each actor, then average equally across eligible actors and scale to 0–100. Unsupported Drift Rate, Missed Adaptation Rate, and Spillover Rate instead pool their respective eligible cases across actors.
- **Episode Collaboration Plausibility:** average applicable dimensions within each decision episode, average episode scores within each thread, then average equally across scorable threads and scale to 0–100.
- **Context-Sensitive Response Calibration:** average applicable calibration dimensions within each actor, then average equally across eligible actors and scale to 0–100.
- **Perturbation Win Rate:** compare original and corrupted episodes using dimensions scored in both versions. Average binary wins over eligible episodes within each thread, then equally across eligible threads for each corruption type, and finally equally across applicable corruption types. Ties are not wins.
- **EEE:** for each of EIP, EPC, and ESA, average applicable dimensions within each Event Card, then average equally across cards eligible for that metric and scale to 0–100.
- **NDD:** Meaningful Event Threads divides the number of confirmed meaningful threads by observed actor-slots and multiplies by 1,000. Supported Novelty and Consequence pool eligible threads. Near-Duplicate Actions pools decidable decision windows. History-Dependence pools decisions with relevant history and a decidable history-use judgment. These four proportions are expressed on 0–100 scales.

Table 2 reports coverage separately from scores. Counts of actors, threads, episodes, and decision windows are within-trajectory evaluation units, not independent simulation replicates. Eligibility differs across metrics, so high scores should be interpreted alongside their coverage.

The ECD episode counts are not universal denominators for Perturbation Win Rate: individual comparisons require jointly scored dimensions, and mechanical-inertia corruptions additionally require prior context. Perturbation results therefore follow the hierarchical aggregation described above.

A.4 Detailed aggregate report for Direct

Table 3 shows the detailed aggregate output of the pipeline underlying the **Direct** column of Table 1 (evaluation protocol v5, LLM judge). We include it here to illustrate the reporting format, denominators, and calibration diagnostics the pipeline tracks. The summary values correspond to the Direct column of Table 1, subject to rounding. This is not an independent exploratory configuration.

Table 3: Detailed aggregate evaluation report for the **Direct** baseline (evaluation protocol v5, LLM judge). Thread extraction: `semantic-subevents-v3` (LLM judge), 25 accepted threads (largest: 239 actions). Uncertainty policy: `quarantine-v3-abstention-validation`. Among event actions, 2,298 were accepted and 2,408 were excluded as uncertain (excluded from ECD/EEE). Accepted subevents: 5489/13621. Source-level accepted/routine/uncertain counts may overlap.

Block	Metric	Value	Range
IDC	IDC Macro	56.51	0–100
IDC	Unsupported Drift Rate	0.615	0–1
IDC	Missed Adaptation Rate	0.334	0–1
IDC	Spillover Rate	0.475	0–1
IDC	Pass-C mode	taxonomy	–
ECD	Collaboration Plausibility	95.92	0–100
ECD	D1 / D2 / D3 / D4 / D5	4.00 / 3.98 / 3.99 / 3.62 / 3.60	0–4 each
ECD	Context-Sensitive Resp. Calibration	84.0	0–100
ECD	Perturbation Win Rate	0.896	0–1
EEE	Event Initiation Plausibility	77.34	0–100
EEE	Event Progression Coherence	71.33	0–100
EEE	Event Status Appropriateness	82.14	0–100
NDD	Meaningful Event Threads	3.889	per 1,000 actor-slots
NDD	Supported Novelty Ratio	100.0	0–100
NDD	Near-Duplicate Actions	73.52	%
NDD	Consequence Rate	100.0	0–100
NDD	History-Dependence Rate	78.68	0–100
Calib.	Human agreement	N/A	–
Calib.	Taxonomy vs. plausibility closeness	77.5	%

For reference, this Direct run had 5 collaboration-eligible threads out of 25 (individual, single-actor event coherence on the remaining 20 threads was 82.5/100 under a separate rubric). It had 339 ECD decision episodes over those 5 threads (dimension coverage 339/339/339/339/337 for D1–D5). The EEE card counts were 24, 25, and 7 for EIP, EPC, and ESA, respectively. CSRC covered 10/10 eligible actors. NDD denominators were 3600 observed actor-slots, 14 meaningfulness-decidable threads (14 meaningful), 4 novelty-eligible threads (11 observed new threads), 14 consequence-eligible threads, and 3105/3127 decisions decidable for history-use and repetition respectively (56 history-citation and 56 history-row abstentions, 1089 downstream-retrieval batches with 8 abstentions, and 11 event-diagnostic abstentions). A purely descriptive statistic, exact normalized action repetition, was separately computed at 65.4% and is reported for context rather than as a scored metric. All LLM-judged scores in this table remain model-based semantic audits rather than human validation.

B Additional SOTOPIA Analyses

We conduct two analyses involving SOTOPIA (Zhou et al., 2024). The first scores SOTOPIA episodes with our longitudinal metrics. The second scores our 60-day trajectories with an adapted version of SOTOPIA's native evaluator. Because the data and scoring procedures differ, the resulting scores are not directly comparable.

Table 4: Longitudinal metrics applied to a stratified sample of 160 SOTOPIA episodes (10 per ordered acting-model pair, 16 pairs), of which 159 completed the pipeline (Table 5 reports per-pair completion, with one pair completing 9/10). Higher is better except where marked ↓. Statistics are equal-episode-weighted means and sample SDs over eligible nonmissing scores from the 159 completed episodes. Valid n varies across metrics because zero-denominator and insufficient-evidence cases are excluded rather than assigned a value (Appendix A).

Metric	Scale	Valid n	Mean	SD
IDC Macro	0–100	132	58.08	15.06
Unsupported Drift Rate ↓	0–1	130	0.221	0.186
Missed Adaptation Rate ↓	0–1	126	0.778	0.205
Spillover Rate ↓	0–1	126	0.164	0.199
Episode Collab. Plausibility	0–100	87	88.48	7.95
Context-Sensitive Resp. Calib.	0–100	128	69.56	15.02
Perturbation Win Rate	0–1	87	0.950	0.050
Event Initiation Plausibility	0–100	122	68.70	17.46
Event Progression Coherence	0–100	119	84.26	13.03
Event Status Appropriateness	0–100	102	95.42	13.10
Meaningful Thread Rate	per 1,000 actor-slots	110	230.32	237.83
Supported Novelty Ratio	0–100	40	66.67	45.76
Repetition Rate ↓	0–100	138	27.24	26.39
Consequence Rate	0–100	82	100.00	0.00
History-Dependence Rate	0–100	128	69.41	30.10

B.1 Longitudinal Metrics on SOTOPIA Episodes

We draw a fixed-seed (seed $= 42$) stratified random sample of 160 episodes from 7,049 usable SOTOPIA episodes, allocating 10 episodes to each of 16 ordered acting-model pairs. The pipeline completed 159/160 episodes (one failure from a thread-linking error, retained rather than imputed), spanning 76 distinct environments and 1,312 retained actions.

A zero-denominator convention in the underlying pipeline writes 0 for IDC when no actor–facet is eligible, and for the three drift/adaptation/spillover rates when no eligible case exists (Appendix A). Excluding these 27–33 zero-denominator episodes per metric, rather than treating the default as a true zero, raises IDC Macro from 48.21 to 58.08, Unsupported Drift Rate from 0.181 to 0.221, Missed Adaptation Rate from 0.617 to 0.778, and Spillover Rate from 0.130 to 0.164. Table 4 reports only the eligibility-adjusted values.

Table 5 breaks the block-level scores down by ordered actor-model pair. We caution against over-interpreting this breakdown: episode counts per pair are small (typically $n \leq 10$ eligible episodes) and episodes sharing an environment or character configuration may be dependent, so the reported SDs describe between-episode dispersion rather than a basis for significance testing. Original SOTOPIA profiles, goals, and human ratings were not supplied to the metric judge, and no cross-metric overall score is computed given the differing scales and eligibility across metrics.

B.2 Native SOTOPIA Evaluation of Our Trajectories

For this analysis, we adapt SOTOPIA's native `EpisodeLLMEvaluator` (Zhou et al., 2024) (repository commit `a0aaafb440e570e5e61b7c44a44e5e417c545383`).

We treat each complete 60-day trajectory as one multi-actor episode and map its ten actors and narrated actions to the native evaluator's agent and message representations. We retain the seven upstream `SotopiaDimensions` rubric definitions and native generation and parsing pipeline, while extending the response schema to record nullable scores, evidence status, source-action citations, and limitations. Because our trajectories do not include original SOTOPIA character profiles, assigned goals, or confidentiality specifications, Believability is marked as based on partial evidence, while Goal and Secret scores are set to N/A and excluded from aggregation. A compatibility patch provides explicit ten-actor key mapping, current score-object extraction, and failure propaga-

Table 5: Block-level scores by ordered acting-model pair (Actor 1, Actor 2), mean ± sample SD (n), 0–100 scales. Because SOTOPIA interactions are dyadic and asymmetric, pair-level scores reflect the joint episode and should not be attributed to either actor model alone.

Actor 1	Actor 2	Compl.	IDC	ECD	EIP	EPC	ESA
gpt-3.5-turbo	gpt-3.5-turbo	10/10	56.3±15.2(10)	89.2±6.9(9)	79.2±13.0(9)	92.4±7.7(9)	100.0±0.0(6)
gpt-3.5-turbo	gpt-4	10/10	60.1±12.0(10)	91.9±4.3(9)	69.6±16.7(10)	95.0±4.7(10)	94.4±9.1(9)
gpt-3.5-turbo	llama-2-70b-chat	10/10	57.8±11.9(9)	87.9±7.1(9)	70.1±23.2(9)	86.5±9.6(9)	92.9±12.2(7)
gpt-3.5-turbo	mpt-30b-chat	10/10	71.4±7.2(4)	N/A(0)	59.4±18.8(4)	79.2±26.0(3)	96.9±6.3(4)
gpt-4	gpt-3.5-turbo	10/10	53.8±15.1(10)	92.9±5.1(7)	67.4±22.2(8)	89.1±12.5(8)	100.0±0.0(6)
gpt-4	gpt-4	10/10	56.8±8.3(10)	93.5±7.0(9)	74.2±16.5(9)	92.3±6.8(10)	98.4±2.9(8)
gpt-4	llama-2-70b-chat	10/10	56.4±14.2(7)	88.5±7.6(4)	60.9±25.6(8)	73.0±11.9(8)	100.0±0.0(6)
gpt-4	mpt-30b-chat	10/10	62.5±29.3(4)	90.9±4.4(3)	64.6±21.5(6)	89.9±9.1(6)	87.5±30.6(6)
llama-2-70b-chat	gpt-3.5-turbo	10/10	57.2±13.0(10)	86.7±8.2(8)	67.3±16.3(9)	83.8±12.6(9)	95.8±9.3(7)
llama-2-70b-chat	gpt-4	10/10	58.3±11.6(9)	84.9±7.6(8)	71.8±12.0(10)	81.4±7.7(10)	90.0±24.2(10)
llama-2-70b-chat	llama-2-70b-chat	10/10	60.9±10.5(9)	76.7±15.0(5)	66.1±8.9(6)	76.0±16.0(6)	93.8±10.5(6)
llama-2-70b-chat	mpt-30b-chat	10/10	54.5±37.2(5)	90.7±1.6(2)	65.3±16.8(3)	83.3±14.4(3)	91.7±14.4(3)
mpt-30b-chat	gpt-3.5-turbo	10/10	53.8±12.2(10)	87.3±1.0(3)	69.8±19.6(9)	77.7±14.7(8)	100.0±0.0(6)
mpt-30b-chat	gpt-4	10/10	63.1±18.1(10)	91.5±3.4(2)	72.3±20.7(7)	82.1±15.6(6)	92.9±18.9(7)
mpt-30b-chat	llama-2-70b-chat	9/10	54.5±15.3(9)	88.4±4.6(4)	65.4±14.6(8)	83.0±10.7(7)	95.0±11.2(5)
mpt-30b-chat	mpt-30b-chat	10/10	61.5±17.5(6)	84.4±10.7(5)	63.8±11.9(7)	71.6±16.0(7)	100.0±0.0(6)

Table 6: Adapted native SOTOPIA evaluator scores (actor mean ± sample SD, $n = 10$ actors) on five 60-day, 10-actor, 3,600-action trajectories, each scored in one joint GPT-4.1 evaluator call. Believability is marked with an asterisk because original character profiles were unavailable, so scores reflect partial evidence only. Secret and Goal are N/A throughout because our setting does not define assigned secrets or goal targets. Because each condition has a single joint evaluation, actor-level SD is descriptive (dispersion across actors within one run) and does not indicate run-to-run reliability or statistical significance.

Metric	E^2Sim	Direct	Guided	Per-Actor	Summary
Believability (0–10)	9.00±0.00*	9.00±0.00*	9.00±0.00*	9.00±0.00*	9.00±0.00*
Relationship (−5–5)	0.00±0.00	1.00±0.00	0.00±0.00	1.00±0.00	1.40±0.52
Knowledge (0–10)	**7.50**±0.71	6.60±0.97	6.80±0.63	7.40±0.52	6.80±0.63
Secret (−10–0)	N/A	N/A	N/A	N/A	N/A
Social rules (−10–0)	0.00±0.00	0.00±0.00	0.00±0.00	0.00±0.00	0.00±0.00
Financial/material (−5–5)	N/A	0.00±0.00	N/A	N/A	N/A
Goal (0–10)	N/A	N/A	N/A	N/A	N/A

tion. Consequently, these results use an adapted native evaluator and are not directly comparable to scores from the unmodified SOTOPIA benchmark.

C Scenario Initialization

Figure 5 shows the initial scenario shared by all five 60-day runs. It places ten actors in four settings within one city: a university machine-learning lab (Alex, Dr. Chen, and Priya), a small lending company (Wei, Nadia, and Tomas), a children's hospital ward (Mei and Hana), and a software studio (Kenji and Iris). The actors' backgrounds connect these settings through family, housing, former advising, and work relationships. Mei is Dr. Chen's spouse and Tomas's sister, Hana and Iris share an apartment, and Kenji is Dr. Chen's former doctoral student and Nadia's work contact.

Each actor starts with a role, background, personality, stable profile, long-term life goal, and two or three initial *Stage Goals*. Three shared goal groups initialize joint projects: an evidence-conflict submission for Alex and Dr. Chen, an attribution paper for Dr. Chen and Priya, and a credit-product launch for Wei and Nadia. Three seed goals have target days, including Priya's coursework on day 2 and the two attribution-paper goals on day 4. The initial world contains public facts, actor-specific visible facts, calendar entries, and setting descriptions. Before E^2Sim starts, a model derives a relationship graph from the actors' backgrounds. The baselines use the same initial scenario without this derived graph.

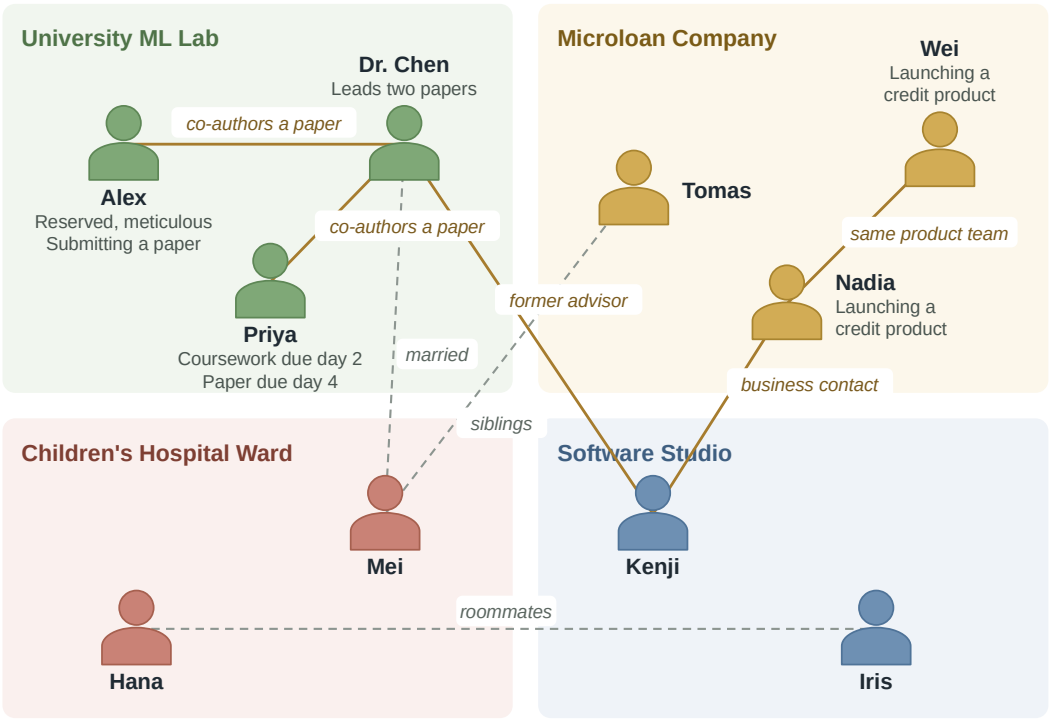


Figure 5: Initial scenario for the 60-day simulations. Ten actors occupy four connected settings, with links showing their project collaborations and other relationships. Selected actor attributes and goals are also shown.

D Positioning Relative to Prior Work

Table 7 provides a structured comparison of E^{2}Sim with representative prior work along the dimensions most relevant to our setting, including long-horizon simulation, evolving actor state, persistent shared environments, multi-actor interaction, actor-specific visibility, and delayed effects.

Table 7: Positioning of E^{2}Sim relative to representative work.

Work	Long horizon	Evolving actor state	Env. grounded	Persistent shared world	Multi-actor	Action–World feedback	Actor-specific visibility	Delayed effects
Generative Agents (Park et al., 2023)	∼	∼	✓	✓	✓	∼	∼	–
Concordia (Vezhnevets et al., 2023)	∼	∼	✓	✓	✓	✓	∼	∼
SOTOPIA (Zhou et al., 2024)	–	–	✓	–	✓	∼	∼	–
LIFELONG-SOTOPIA (Goel & Zhu, 2025)	✓	∼	✓	–	✓	∼	∼	–
AgentSociety (Piao et al., 2026)	✓	∼	✓	✓	✓	✓	∼	–
BehaviorChain (Li et al., 2025)	∼	∼	∼	–	–	–	–	–
OmniBehavior (Chen et al., 2026a)	✓	∼	∼	–	–	–	–	–
PERSONAMEM / DEEPER (Jiang et al., 2025; Chen et al., 2025)	✓	✓	–	–	–	–	–	–
LifeSim (Duan et al., 2026)	✓	✓	✓	–	–	∼	–	∼
E^{2}Sim	✓	✓	✓	✓	✓	✓	✓	✓

✓: explicit mechanism or primary capability, ∼: partial or implicit support, –: not a primary part of the setting.

E Exploratory Question Answering Study

We constructed an 80-question, four-choice benchmark from the 60-day simulated trajectory to explore whether its history supports questions about actors and events. The questions cover 13 categories, with 40 easy questions answerable from a single record and 40 hard questions requiring information across longer periods or multiple actors. We checked that each answer could be recovered from the full trajectory. Figure 6 shows the question distribution.



Figure 6: Question categories and scenario domains for the 80 four-choice questions. The inner ring shows `gpt-5.6-luna` accuracy with compressed memory, averaged over three answers per question.

We tested direct API question answering under two input conditions. With full history, the model receives all records up to the question time when answering each question. With compressed memory, it reads the records sequentially, updates compressed notes as new information arrives, and answers using those notes plus the most recent three days of records. This condition more closely approximates an assistant that learns about a user through ongoing interactions and must compress earlier information for later use. Each question is answered three times. Table 8 reports mean accuracy. With full history, the two stronger models answer at least 90% of hard questions, showing that most answers are recoverable. On hard questions, accuracy falls to 33–35% when the history is processed sequentially and compressed, supporting our concern about long-horizon information use.

Table 8: Accuracy (%) on the exploratory question-answering benchmark, averaged over three answers per question. Each difficulty level contains 40 questions.

	Full history		Compressed memory	
Model	Easy	Hard	Easy	Hard
`gpt-6-luna`	100.0	93.3	65.0	35.0
`gpt-5.6-luna`	94.2	90.0	70.0	33.3
`gpt-5.4-nano`	69.2	43.3	55.0	32.5

This is a preliminary test on questions from one synthetic trajectory. The answers are produced by direct API calls to language models, without an acting agent or an agentic retrieval workflow. The scores therefore measure question-answering accuracy under the specified inputs, not the performance of a deployed assistant agent.