# Beyond Offline A/B Testing: Context-Aware Agent Simulation for Recommender System Evaluation

**Nicolas Bougie[1], Gian Marconi Marconi[1], Xiaotong Ye[1], Narimasa Watanabe[1]**
{nicolas.bougie,gianmaria.marconi,tony.yip,narimasa.watanabe}@woven.toyota

[1]Woven by Toyota

## Abstract

Recommender systems are central to online services, enabling users to navigate through massive amounts of content across various domains. However, their evaluation remains challenging due to the disconnect between offline metrics and online performance. The emergence of Large Language Model-powered agents offers a promising solution, yet existing studies model users in isolation, neglecting the contextual factors such as time, location, and needs, which fundamentally shape human decision-making. We introduce ContextSim, an LLM agent framework that simulates believable user proxies by anchoring interactions in daily life activities. A life simulation module generates scenarios specifying when, where, and why users engage with recommendations. To align preferences with genuine humans, we model agents' internal thoughts and enforce consistency at both action and trajectory levels. Experiments across domains demonstrate closer alignment with human behavior than prior work. We further validate our approach through offline A/B testing correlation and show that RS parameters optimized using ContextSim yield improved real-world engagement.

## 1 Introduction

In the era of information explosion, recommender systems (RS) have become an indispensable component of digital platforms, providing personalized recommendations that shape user satisfaction across applications from e-commerce to social media (Li et al., 2024). Nevertheless, evaluating offline their true effectiveness remains an open challenge (Yoon et al., 2024). Traditional evaluation relies on proxy metrics such as `hit rate` or `Recall@N`, but these often fail to predict how users will behave once a system is deployed (Zhang et al., 2019). The root cause lies in a fundamental disconnect: offline metrics capture static preference patterns, whereas real users make decisions dynamically, influenced by context, internal state, and circumstances (Jannach and Jugovac, 2019). Online A/B testing addresses this gap but introduces its own drawbacks, including high costs, privacy issues, and ethical concerns around exposing users to potentially suboptimal experiences.

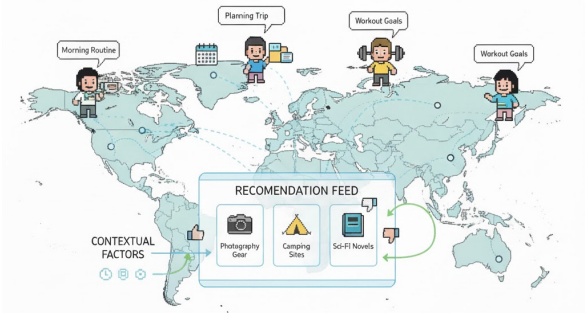


Figure 1: The ContextSim framework for evaluating recommender systems.

Breakthroughs in Large Language Models (LLMs) have shown promise in human behavior modeling by enabling the creation of autonomous agents. In the realm of recommendation systems, RecMind (Wang et al., 2024) explores the concept of autonomous recommender agents equipped with self-inspiring planning and external tool utilization. Recently, InteRecAgent (Huang et al., 2025) has extended this idea by proposing memory components, interactive task planning, and reflection. RecAgent (Wang et al., 2025a) has attempted to introduce more diverse user behaviors, taking into account external social relationships. Another work, SimUSER (Bougie and Watanabe, 2025b), investigates aligning synthetic agents with their human counterparts through path-driven retrieval and image-grounded reasoning. However, a common drawback of prior studies is their *insulated nature* as they primarily rely on past interactions to make decisions, neglecting environmental factors (Adomavicius and Tuzhilin, 2010). For example, a user



*EMNLP 2026 Industry Track*, pages 5–24
October 24-29, 2026 ©2026 Association for Computational Linguistics

shopping in preparation for an upcoming trip may prioritize specific items such as travel accessories, while the same user browsing after a recent move may instead explore furniture or home organization products. Current approaches overlook these contextual factors, leading to synthetic users that behave with unrealistic uniformity. Besides, existing work relies heavily on few-shot exemplars to align agents with historical preferences (Li et al., 2026). In contrast, we argue that explicitly modeling internal thoughts is essential for generalizing preferences to unfamiliar items and maintaining coherent behavior across time and varying contexts.

In light of this, we introduce ContextSim, a framework that simulates contextualized user behavior and produces faithful interaction trajectories through explicit thought synthesis. Rather than modeling isolated browsing sessions, ContextSim generates daily schedules for each agent, specifying activities, locations, goals, and budgets that shape when, where, and why agents engage with the recommender system. To align users with historical preferences, agents engage in explicit thought modeling: they articulate why a chosen action aligns with their persona, preferences, and situational context. Thought synthesis is performed at both the action and trajectory level via two auxiliary tasks, ensuring internal coherence and belief emergence. Agents then interact with the recommender system following the resulting policy conditioned on their persona and memory modules. This produces fine-grained interaction trajectories that enable reliable RS evaluation and metric estimation under realistic, context-aware settings.

## 2 Related Work

Early work on user simulation relied on bandit-based models that learned from binary feedback signals (Christakopoulou et al., 2016). KuaiSim (Zhao et al., 2023) advanced this line of research by introducing a simulation platform that supports richer interaction modes. Subsequent approaches incorporated natural language interfaces, but user responses were still largely constrained to predefined choices, such as binary or multiple-choice feedback (Lei et al., 2020). As a result, these rule-based simulators lack the behavioral diversity and adaptability observed in real users. The advent of large language models has enabled a new generation of generative user simulators with increased flexibility and expressiveness (Zhang et al., 2024c). Such models produce more natural language interactions and reduce reliance on hand-crafted rules, yet challenges remain in calibration, consistency, and behavioral realism. In practice, many LLM-based simulators still depend on scripted interaction flows or weak alignment mechanisms, limiting their ability to exhibit diverse and coherent behaviors over time (Lei et al., 2020; Zhao et al., 2023).

Parallel to this line of work, several studies have explored the use of LLMs as recommender systems themselves, rather than as user simulators (Hou et al., 2024; Li et al., 2023; Kang et al., 2023). These approaches investigate the capacity of LLMs to generate recommendations directly, providing a perspective complementary to ours, which focuses on modeling how users perceive and respond to recommendations (Wang et al., 2025b; Zhang et al., 2024b).

More closely related to our work are LLM-based autonomous user agents for recommendation simulation. RecMind (Wang et al., 2024) pioneers self-inspiring agents but limits interactions to simple rating actions. Yoon et al. (Yoon et al., 2024) conduct a systematic study of LLM effectiveness in conversational recommendation simulation. Agent4Rec (Zhang et al., 2024a) introduces memory mechanisms to maintain interaction history and improve behavioral faithfulness. Recently, RecInter (Jin et al., 2025) seeks to reproduce phenomena such as the Matthew Effect. SimUSER (Bougie and Watanabe, 2025b) further advances this direction by incorporating persona matching via self-consistency scoring, knowledge-graph memory to capture user–item relationships, and visual perception for thumbnail-driven decisions. Similarly, AlignUSER leverages counterfactual reasoning and next-state prediction to make agents aware of the world (Bougie et al., 2026). Despite these advances, existing approaches predominantly rely on weak alignment signals, such as few-shot prompting, and model user behavior largely in isolation from the broader situational context. Meanwhile, the importance of context for recommendation quality has long been established (Adomavicius et al., 2021): temporal dynamics influence preference expression (Koren, 2009), location affects perceived relevance (Liu et al., 2017), and situational factors shape decision criteria (Zheng et al., 2015). However, this rich literature on context-aware recommendation has not yet translated into context-aware user simulation.

## 3 Method

ContextSim presents a framework for evaluating RS through contextually-grounded agent interactions. Our approach comprises three phases: (1) persona initialization from historical data, (2) thought modeling via explicit reasoning, (3) daily life simulation that generates environmental context shaping subsequent interactions with the RS.

**Problem Formulation:** We model user–recommender interaction as a tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, T)$, where $\mathcal{S}$ denotes the state space and $\mathcal{A}$ is the action space. The transition function $T: \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$ governs the environment dynamics. In our study, the state $s$ encodes what is visible to the agent (e.g., a ranked list of items $i \in \mathcal{I}$) and context $c \in \mathcal{C}$ (e.g., temporal, spatial, and situational factors). An action $a_t \in \mathcal{A}$ represents discrete operations such as inspecting an item, clicking, adding to cart, rating, or exiting. Our objective is to simulate realistic trajectories $(s_0, a_0, \ldots, s_T)$ for each user following a policy $\pi_\theta : S \rightarrow A$ parameterized by $\theta$, including predicting the rating $r_{ui}$ for unseen items $i$ given $c$.

### 3.1 Persona Initialization

We represent each agent's profile $p$ through demographic and psychological attributes: **age**, **occupation**, and **personality**. Personality follows the Big Five traits: $\in$ {*Openness*, *Conscientiousness*, *Extraversion*, *Agreeableness*, *Neuroticism*)}, scored on a 1-3 scale. Since real-world datasets rarely include such attributes, we infer them from interaction histories using self-consistent persona selection (Bougie and Watanabe, 2025b). We first prompt the LLM to generate plausible candidate personas, then score via a consistency score. The persona with the highest consistency is assigned to the agent. Beyond these static attributes, $p$ also includes: **habits**, **recent goals**, **preferences**. Habits account for user tendencies in engagement, conformity, and variety (Zhang et al., 2024a). Recent goals are inferred from prior sessions, and preferences are short natural-language descriptions that summarize the user's tastes.

**Memory Module** The episodic memory records the interactions with the RS. This memory is initially populated with the user's viewing, rating history, and preferences. Each time an agent executes a new action or rates an item, the corresponding interaction is added to the episodic memory. We also maintain an emotional memory that records user feelings during system interactions, such as level of fatigue and satisfaction, capturing the psychological feeling stemming from past interactions.

### 3.2 Thought Synthesis

Relying solely on persona-style instructions (e.g., "*act as ...*") often leads LLM agents to overfit persona descriptors rather than understanding their actual interaction patterns. To better align agents with user preferences and enable generalization beyond previously seen items, we argue that explicitly modeling the user's internal thoughts is crucial. To do so, we let agents synthesize their own thoughts, such as *I loved The Matrix and Blade Runner. Interstellar has similar sci-fi themes and Nolan directed it, so I'm clicking to watch*. Given a user's interaction history $H \leftarrow \{(s_t, a_t, c, p, H_a), ...\}$, the agent compares the candidate actions and infers why the human choice aligns with $c$, $p$, and past actions $H_a$. Concretely, we introduce two reasoning tasks at the item and trajectory levels.

**Item Disentanglement.** Given an item $i$ and the user's historical action $a_t$ (e.g., purchase, click, rating), the agent infers why the action reflects its preferences. The agent is instructed to ground the explanation in persona $p$, interaction history $H$, and item features.

**Trajectory Alignment:** In this task, the agent is given a state $s_t$, the historical action $a_t$, alternative navigation actions $A_t$ (e.g., next page, exit), history $H$, and is prompted to generate a brief rationale that explains: (i) why the historical action is preferred over other actions, and (ii) highlights why the action aligns with $p$ and $H$. The prompt explicitly requires the thought to reference observable aspects rather than generic justifications.

The resulting thoughts $c_{ID}$ and $c_{TA}$ are collected into datasets $\mathcal{D}_{ID}$ and $\mathcal{D}_{TA}$, respectively. During data construction, the observed human action $a_t$ is provided to the teacher model to synthesize the corresponding rationale. During policy training, however, $a_t$ is removed from the model input. Given the state, persona, context, interaction history, and available actions, the policy is trained to generate the observed action followed by its rationale. We train $\pi_\theta$ via SFT with a joint objective over reasoning generation:

$$\mathcal{L}_{\text{ST}} = -\sum_{(s_t, a_t, c, p, H_a, c_{ID}) \in \mathcal{D}_{\text{ID}}} \log p_\theta(a_t, c_{ID} \mid s_t, c, p, H_a) - \sum_{(s_t, a_t, A_t, c, p, H_a, c_{TA}) \in \mathcal{D}_{\text{TA}}} \log p_\theta(a_t, c_{TA} \mid s_t, A_t, c, p, H_a). \quad (1)$$

This objective teaches the policy not only which action was selected, but also why it is preferable given the user's persona, history, and context.

### 3.3 Life Simulation Module

In the real world, interactions do not occur in isolation but emerge from the activities and constraints of everyday life. Thus, we utilize a life simulation module (Bougie and Watanabe, 2025a) to produce realistic contexts. For each agent with persona $p$, we generate a daily schedule $\mathcal{S} = \{(t_1, a_1, l_1), \ldots, (t_n, a_n, l_n)\}$ where each tuple represents a time slot $t$, activity $a$, and location $l$. The schedule is conditioned on persona attributes, day type, past activities, and external factors (weather conditions, local events, season). From these schedules, we identify when recommendation interactions would naturally occur and construct a corresponding contextual scenario $c$ comprising:

- **Temporal context** $c_t$: time of day, day of week,
- **Spatial context** $c_l$: current location,
- **Situational context** $c_s$: latest activity, mood, need level (e.g., hunger), and energy level,
- **Goal context** $c_g$: purpose of seeking recommendations,
- **Constraint context** $c_b$: budget, time availability,

The full context vector is: $c = (c_t, c_l, c_s, c_g, c_b)$.

### 3.4 Context-Aware Interactions

We now consider two ways of interacting with the RS: CONTEXTSIM(SIM), CONTEXTSIM(SUM). In the former, at each step, we run the life simulation module, simulate the context, and if the agent decides to seek recommendations, it engages with the RS. Thus, each interaction is conditioned on dynamic factors such as the current time, location, recent activity, mood, goal, and constraints. In the latter, we run the life simulation module for 30 days, then summarize contextual factors and append them to the agent's persona. Consequently, CONTEXTSIM(SUM) acts as an enriched static user profile, whereas CONTEXTSIM(SIM) provides interaction-specific dynamic context.

When interacting with the RS, the agent conditions its decisions on the current context $c$. It first senses the page and evaluates each item by estimating its alignment with the persona, context, and retrieved evidence from the episodic memory, producing a shortlist of `[WATCH]` or `[SKIP]` intentions. Following this, the agent infers its fatigue, curiosity, and boredom, before action selection. When selecting an action, the agent generates an internal thought that weighs the available actions (e.g., scroll, click, add to cart, rate, exit) against its goals and constraints. Each reasoning step is accompanied by a *thought* to maintain behavioral consistency. This loop is repeated until it decides to stop interacting with the RS for the current session or selects a final action (e.g., `[PURCHASE CART]`, `[EXIT]`). After acting, the agent performs self-reflection, updating the memory module with concise reflections explaining its decisions and evolving tastes.

## 4 Experiments

**Settings.** We evaluate CONTEXTSIM with Qwen3-8B (Yang et al., 2025) as backbone. Unless otherwise specified, we report results obtained with the summary-based variant, CONTEXTSIM(SUM). All experiments are conducted with 1,000 agents.

**Datasets.** Evaluation spans four domains: MovieLens-1M (movies), AmazonBook (books), and Steam (video games), and OPeRA (Wang et al., 2025c) (EC site). We follow standard preprocessing and use time-based 80/10/10 splits for train/-validation/test. For temporal analysis, we leverage MovieLens and OPeRA timestamps to establish ground-truth interaction patterns. Besides, for trajectory-level evaluation, we rely on the OPeRA dataset (Wang et al., 2025c).

**Baselines.** We compare our method against RecAgent, Agent4Rec, and SimUSER as LLM-powered agent baselines. When possible, we also include RecMind results. All baselines are implemented using GPT-4o-mini, as reported in their original experimental settings. We additionally rerun all methods with Qwen3-8B to control for backbone choice (Appendix F.17).

### 4.1 Preference Alignment

For synthetic users to provide useful feedback to a recommender system, they should preserve the preference patterns of the real users they represent. We therefore prompt the agents to classify items based on whether their human counterparts have interacted with them or not. For each of 1,000 agents, we sample 20 candidate items and vary the ratio of interacted to non-interacted items as 1:$m$, where $m \in \{1, 3, 9\}$ and non-interacted items are labeled as $y_{ui} = 0$. This yields a binary classifica-

| Method(1:m) | MovieLens | | | | AmazonBook | | | | Steam | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Accuracy | Precision | Recall | F1 Score | Accuracy | Precision | Recall | F1 Score | Accuracy | Precision | Recall | F1 Score |
| RecAgent (1:1) | 0.5807 | 0.6391 | 0.6035 | 0.6205 | 0.6035 | 0.6539 | 0.6636 | 0.6587 | 0.6267 | 0.6514 | 0.6490 | 0.6499 |
| RecAgent (1:3) | 0.5077 | 0.7396 | 0.3987 | 0.5181 | 0.6144 | 0.6676 | 0.4001 | 0.5003 | 0.5873 | 0.6674 | 0.3488 | 0.4576 |
| RecAgent (1:9) | 0.4800 | 0.7491 | 0.2168 | 0.3362 | 0.6222 | 0.6641 | 0.1652 | 0.2647 | 0.5995 | 0.6732 | 0.1744 | 0.2772 |
| Agent4Rec (1:1) | 0.6912 | 0.7460 | 0.6914 | 0.6982 | 0.7190 | 0.7276 | 0.7335 | 0.7002 | 0.6892 | 0.7059 | 0.7031 | 0.6786 |
| Agent4Rec (1:3) | 0.6675 | 0.7623 | 0.4210 | 0.5433 | 0.6707 | 0.6909 | 0.4423 | 0.5098 | 0.6505 | 0.7381 | 0.4446 | 0.5194 |
| Agent4Rec (1:9) | 0.6175 | 0.7753 | 0.2139 | 0.3232 | 0.6617 | 0.6939 | 0.2369 | 0.3183 | 0.6021 | 0.7213 | 0.1901 | 0.2822 |
| SimUSER (1:1) | 0.7912 | 0.7976 | 0.7576 | 0.7771 | 0.8221 | 0.7969 | 0.7841 | 0.7904 | 0.7905 | 0.8033 | 0.7848 | 0.7939 |
| SimUSER (1:3) | 0.7737 | 0.8173 | 0.5223 | 0.6373 | 0.6629 | 0.7547 | 0.5657 | 0.6467 | 0.7425 | 0.8048 | 0.5376 | 0.6446 |
| SimUSER (1:9) | 0.6791 | 0.8382 | 0.3534 | 0.4972 | 0.6497 | 0.7588 | 0.3229 | 0.4530 | 0.7119 | 0.7823 | 0.2675 | 0.3987 |
| ContextSim (1:1) | 0.824 | 0.831 | 0.789 | 0.809 | 0.851 | 0.823 | 0.812 | 0.817 | 0.819 | 0.828 | 0.801 | 0.814 |
| ContextSim (1:3) | 0.798 | 0.841 | 0.551 | 0.666 | 0.689 | 0.779 | 0.592 | 0.673 | 0.768 | 0.829 | 0.561 | 0.669 |
| ContextSim (1:9) | 0.712 | 0.861 | 0.381 | 0.528 | 0.678 | 0.784 | 0.351 | 0.485 | 0.739 | 0.809 | 0.294 | 0.431 |

Table 1: User preference alignment across MovieLens, AmazonBook, and Steam datasets.

| **Methods** | **MovieLens** | | **AmazonBook** | | **Steam** | |
|---|---|---|---|---|---|---|
| | **RMSE** | **MAE** | **RMSE** | **MAE** | **RMSE** | **MAE** |
| MF | 1.214 | 0.997 | 1.293 | 0.988 | 1.315 | 1.007 |
| AFM | 1.176 | 0.872 | 1.301 | 1.102 | 1.276 | 0.972 |
| RecAgent | 1.102 | 0.763 | 1.259 | 1.119 | 1.077 | 0.960 |
| Agent4Rec | 0.761 | 0.714 | 0.879 | 0.671 | 0.758 | 0.688 |
| SimUSER | 0.502 | 0.446 | 0.568 | 0.421 | 0.587 | 0.532 |
| ContextSim | **0.451** | **0.392** | **0.511** | **0.369** | **0.528** | **0.471** |
| ContextSim(fs) | 0.488 | 0.421 | 0.542 | 0.408 | 0.561 | 0.503 |

Table 2: Rating prediction performance. **Bold**: best results; underlined: second-best. ContextSim's improvements are statistically significant ($p < 0.05$).

| **Method** | **Consistency Rate** | **Contradiction Rate** |
|---|---|---|
| RecAgent | 17.3% | 38.2% |
| Agent4Rec | 21.8% | 34.6% |
| SimUSER | 29.2% | 29.8% |
| ContextSim | **84.1%** | **5.3%** |

Table 3: Persona-action consistency rates evaluated by GPT-4o. Improvements are statistically significant ($p < 0.05$).

tion task over user–item pairs. Table 1 highlights that ContextSim agents accurately identified items aligned with their tastes, significantly outperforming baselines across all distractor levels (paired t-tests, 95% confidence, $p < 0.001$). The improvement stems from modeling thoughts, as agents can understand what features or attributes (e.g., genre, brand, price range) drive their preferences, enhancing alignment.

### 4.2 Rating Items

A fundamental task when interacting with a RS is rating items. We compare several LLM-based baselines, along with traditional recommendation baselines: MF (Koren et al., 2009) and AFM (Xiao et al., 2017). We also compare against a GPT-4o-mini variant, denoted CONTEXTSIM(FS), which replaces thought-synthesis pretraining with thoughts as few-shot exemplars. Across all tasks (Table 2), ContextSim outperforms other LLM-powered agents. Notably, the thought-synthesis–trained model, CONTEXTSIM, surpasses the few-shot variant CONTEXTSIM(FS), even when the latter relies on a substantially larger LLM. This gap highlights an inherent limitation of few-shot prompting: without explicit training on structured reasoning, the model fails to internalize how preferences, context, and history jointly shape user ratings, resulting in shallow and often brittle decision behavior.

### 4.3 Thought Consistency

We next assess whether explicit thought modeling helps agents maintain persona-consistent behavior. We use the OPeRA dataset (Wang et al., 2025c), which provides shopping sessions annotated with user personas, step-level rationales, and actions. For each rationale, the agent receives the same persona, the observation, and the interaction history, and is prompted to generate both an internal rationale and a next action. Following prior work, GPT-4o acts as an automatic judge and labels each (rationale, action) pair as *coherent*, *partially coherent*, or *contradictory* with respect to the persona and context. We report the proportion of coherent steps as the consistency rate and the proportion of contradictory steps as the contradiction rate. As shown in Table 3, ContextSim demonstrates a substantially higher consistency rate than all baselines, while reducing contradiction.

### 4.4 Believability of Synthetic Users

Following prior work on LLM-based evaluators (Chiang and Lee, 2023; Bougie and Watanabe, 2025a), which has shown that LLM judges

| | MovieLens | AmazonBook | Steam | OPeRA |
|---|---|---|---|---|
| RecAgent | 3.01 ± 0.14 | 3.14 ± 0.13 | 2.96 ± 0.17 | 3.05 ± 0.15 |
| Agent4Rec | 3.04 ± 0.12 | 3.21 ± 0.14 | 3.09 ± 0.16 | 3.15 ± 0.17 |
| SimUSER | 4.41±0.16 | 3.99±0.18 | 4.02±0.23 | 4.05±0.20 |
| ContextSim(sim) | 4.58 ± 0.13* | **4.32 ± 0.17*** | 4.26 ± 0.21* | 4.21±0.20* |
| ContextSim(sum) | **4.60 ± 0.14*** | 4.28 ± 0.16* | **4.31 ± 0.19*** | **4.21±0.18*** |

Table 4: Human-likeness score evaluated by GPT-4o across recommendation domains. *Significant improvements over best baseline ($p < 0.05$).

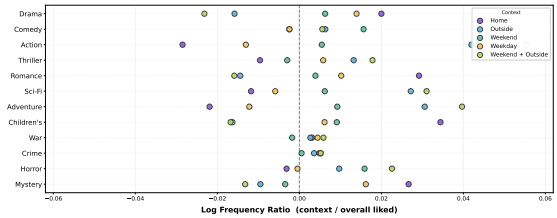


Figure 2: Context effects on agent likes in simulation. Each point shows the log frequency ratio of a genre within a context relative to the overall liked distribution across all contexts.

can approximate human preferences, we use GPT-4o to assess whether agent interactions appear human or AI-generated using a 5-point Likert scale, with higher scores indicating stronger resemblance to human-like responses. As reported in Table 4, ContextSim achieves the highest scores across all four domains, with statistically significant improvements over SimUSER in every dataset ($p < 0.05$). We noticed that the life-context grounding contributes to the faithfulness of our method, especially in the generated rationales. In addition, explicit thought synthesis reduces overly generic or uniformly positive reactions that exhibit prior work.

### 4.5 Context Effects on Preferences

A core motivation of our framework is that user decisions are shaped by *context* (e.g., being at home vs outside, weekday vs weekend). To test whether our agents respond to context in a behaviorally meaningful way, we analyze how the distribution of *liked* items changes across contexts produced by our life simulation module. We run the simulator on MovieLens items and log each interaction together with the agent's context label (location: home/outside; day type: weekday/weekend) assigned by our daily-life module. We define an interaction as positive ("liked") when the agent issues a positive feedback event (e.g., rating ≥ 4). For each context $c$ and genre $g$, we report the log frequency ratio: $\log \frac{p_{\text{sim}}(g|c)}{p_{\text{sim}}(g|\text{all})}$, where $p_{\text{sim}}(g \mid c)$ is the empirical genre frequency among liked interactions

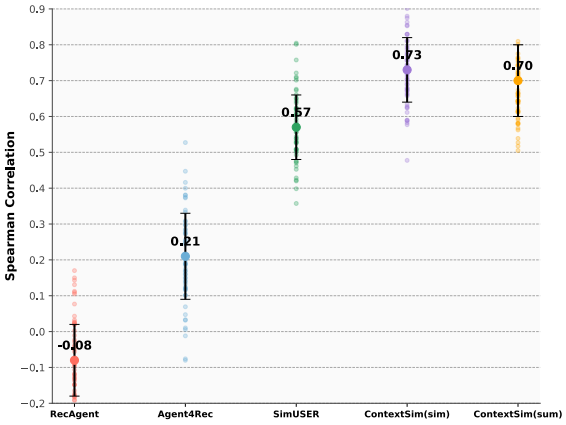


Figure 3: Spearman correlation between estimated and actual engagement metrics. Higher values indicate better alignment with ground-truth metrics.

in context $c$. Figure 2 visualizes these shifts for the most frequent genres. Positive values indicate that a genre is over-represented among liked items in a given context compared to the global average, reflecting a positive contextual bias, while negative values indicate under-representation. For example, *Drama* is more likely to be liked at home, whereas some genres (e.g., *Crime*) show near-zero shifts, suggesting limited sensitivity to contextual factors.

### 4.6 Offline A/B Testing

We further examine whether ContextSim can serve as a reliable proxy for online A/B tests. We use a proprietary dataset of 55 historical A/B experiments on a large-scale food recommendation platform, each involving thousands of user sessions and hundreds of thousands of recommended item impressions. Every test compares multiple recommendation strategies, with the average number of visited pages used as the primary business metric. For each historical test, we instantiate one agent for each user in the corresponding A/B population and initialize the agent only with information available before the test period, such as past interactions and preference history. For each strategy, we run the corresponding simulator and estimate the same engagement metric, then compute the Spearman correlation between simulated and real-world outcomes across the 55 tests. As illustrated in Figure 3, our approach achieves the highest correlation with ground truth, outperforming all other baselines. Statistical tests confirm that the improvements over all baselines are significant ($p < 0.05$).

| Method | $\overline{P}_{\text{view}}$ | $\overline{N}_{\text{like}}$ | $\overline{P}_{\text{like}}$ | $\overline{N}_{\text{exit}}$ | $\overline{S}_{\text{sat}}$ |
|---|---|---|---|---|---|
| Baseline | 0.521 | 5.44 | 0.458 | 3.21 | 3.82 |
| Traditional (nDCG@10) | 0.535 | 5.52 | 0.462 | 3.26 | 3.86 |
| SimUSER-optimized | 0.561 | 5.80 | 0.517 | 3.87 | 4.09 |
| ContextSim-optimized | **0.589** | **6.12** | **0.543** | **4.15** | **4.24** |

Table 5: Performance comparison of parameter selection strategies on various engagement metrics.

### 4.7 Optimizing RS Parameters

We now assess whether choosing RS parameters based on ContextSim(sim)'s evaluations leads to measurable gains in a real deployment. ContextSim is used to rank candidate parameter settings before deployment, and Table 5 reports the metrics observed after deploying the selected settings in the real world. Namely, the table compares the baseline RS, with three selection strategies: traditional offline metrics (nDCG@10), SimUSER, and ContextSim. Results depicted in the table are computed from real logs collected after deploying the *optimized* recommender system. Optimizing for nDCG@10 yields only marginal improvements over the baseline, echoing prior findings that offline accuracy metrics do not reliably translate into business impact (Jannach and Jugovac, 2019). In contrast, parameters selected using ContextSim achieve the best results across all engagement metrics, including higher average viewing ratio ($\overline{P}_{\text{view}}$), more liked items ($\overline{N}_{\text{like}}$), and higher satisfaction scores ($\overline{S}_{\text{sat}}$). Compared to SimUSER-optimized parameters, ContextSim yields a higher viewing ratio and satisfaction, indicating that richer contextual grounding provides more actionable guidance for real-world RS tuning.

### 4.8 Human Pairwise Preference

To evaluate the quality of simulator-generated trajectories, five evaluators were given 200 samples, each with two anonymous trajectories generated by two different methods for the same underlying input. The evaluator was tasked with selecting the preferred trajectory between the two provided (ties allowed). We ranked methods using a win matrix (Figure 4) and Bradley–Terry (BT) model coefficients. The win matrix records matchup outcomes, where element $w_{ij}$ indicates the probability that method $i$ defeats method $j$, with ties counted as half wins. Overall, the resulting rankings are consistent with those obtained from the LLM-based judges (see section 4.4).

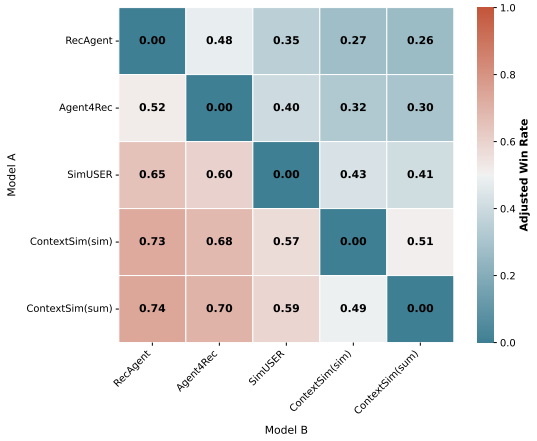


Figure 4: Win matrix heatmap based on pairwise preference evaluation. Each entry $w_{ij}$ denotes the adjusted win probability that method $i$ is preferred over method $j$.

## 5 Conclusion

We propose CONTEXTSIM, a simulation framework that grounds user agents in realistic daily life contexts. Rather than relying on few-shot prompting, we pretrain agents through thought synthesis tasks that teach them to articulate why their decisions align with their persona, history, and situational context. Across multiple datasets and evaluation metrics, CONTEXTSIM produces more believable, context-aware trajectories than existing techniques. Our agents align closely with human preferences, capture temporal interaction patterns overlooked by prior work, and exhibit a strong correlation with real-world A/B test outcomes. Beyond mere correlation, we demonstrate that RS parameters optimized using our approach translate directly into higher real-world KPIs, confirming that user simulation can drive measurable improvements in production systems.

## 6 Ethics Statement

This work proposes an LLM-driven agent framework that simulates user interactions with recommender systems. While synthetic users can reduce reliance on costly or intrusive human experiments, they also raise ethical concerns. Detailed daily life simulations could, if misused, enable fine-grained profiling, behavioral manipulation, or the reinforcement of harmful stereotypes.

We emphasize that ContextSim is intended for *system-level* evaluation and analysis, not for targeting or inferring attributes of specific individuals. All datasets used in our experiments are ei-

ther public benchmarks or proprietary logs that were anonymized and aggregated before analysis, following the data governance policies of our organization. The explicit thought modeling in ContextSim provides a degree of transparency: it makes it easier to inspect and understand long-tail behaviors, rather than hiding them inside a black-box policy.

We view synthetic users as a complement, not a replacement, for real user feedback, especially for populations that may be underrepresented or poorly modeled by LLMs. Practitioners deploying frameworks like ContextSim should ensure that their use complies with applicable privacy regulations, includes regular audits of potential biases, and incorporates human oversight in high-stakes decisions.

## 7 Limitations

Although ContextSim achieves strong performance across the evaluated metrics, several limitations remain. First, reproducibility is constrained by the fact that part of our evaluation relies on one proprietary interaction dataset. Second, as with all LLM-based simulators, ContextSim may inherit cultural, gender, and socioeconomic biases present in foundation models. We also observe occasional hallucinations, particularly when the model generates appraisals for rare or recently added or popular items, which can introduce noise into simulated interactions. Moreover, the fidelity of the simulated behavior ultimately depends on the capabilities and failure modes of the underlying LLMs. Inconsistent reasoning, biased interpretations, or unfounded assumptions made by the base model can propagate through the simulation pipeline. Another limitation is that part of the framework remains prompt-dependent. Although the interaction policy is fine-tuned with decision-rationale supervision, modules such as persona initialization, life simulation, and reflection may remain sensitive to prompt design. We evaluate prompt robustness in Section F.13. Nevertheless, transferring the framework to substantially different domains or interfaces may require revising these prompts and validating their effects. Finally, ContextSim integrates multiple interacting modules (life simulation, thought modeling), making it challenging to attribute improvements to a single component. While our ablation studies attempt to evaluate the contribution of each module, fully isolating their effects remains difficult.

## References


Gediminas Adomavicius, Konstantin Bauman, Alexander Tuzhilin, and Moshe Unger. 2021. Context-aware recommender systems: From foundations to recent developments. In *Recommender systems handbook*, pages 211–250. Springer.

Gediminas Adomavicius and Alexander Tuzhilin. 2010. Context-aware recommender systems. In *Recommender systems handbook*, pages 217–253. Springer.

Nicolas Bougie, Gian Maria Marconi, Tony Yip, and Narimasa Watanabe. 2026. Alignuser: Human-aligned llm agents via world models for recommender system evaluation. *arXiv preprint arXiv:2601.00930*.

Nicolas Bougie and Narimawa Watanabe. 2025a. Citysim: Modeling urban behaviors and city dynamics with large-scale llm-driven agent simulation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 215–229.

Nicolas Bougie and Narimawa Watanabe. 2025b. Simuser: Simulating user behavior with large language models for recommender system evaluation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 6: Industry Track)*, pages 43–60.

Cheng-Han Chiang and Hung-yi Lee. 2023. Can large language models be an alternative to human evaluations? In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15607–15631.

Konstantina Christakopoulou, Filip Radlinski, and Katja Hofmann. 2016. Towards conversational recommender systems. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 815–824.

Yupeng Hou, Junjie Zhang, Zihan Lin, Hongyu Lu, Ruobing Xie, Julian McAuley, and Wayne Xin Zhao. 2024. Large language models are zero-shot rankers for recommender systems. In *European conference on information retrieval*, pages 364–381. Springer.

Xu Huang, Jianxun Lian, Yuxuan Lei, Jing Yao, Defu Lian, and Xing Xie. 2025. Recommender ai agent: Integrating large language models for interactive recommendations. *ACM Transactions on Information Systems*, 43(4):1–33.

Dietmar Jannach and Michael Jugovac. 2019. Measuring the business value of recommender systems. *ACM Transactions on Management Information Systems (TMIS)*, 10(4):1–23.

Song Jin, Juntian Zhang, Yuhan Liu, Xun Zhang, Yufei Zhang, Guojun Yin, Fei Jiang, Wei Lin, and Rui Yan. 2025. Beyond static testbeds: An interaction-centric agent simulation platform for dynamic recommender systems. In *Proceedings of the 2025 Conference on*

*Empirical Methods in Natural Language Processing*, pages 18914–18931.

Wang-Cheng Kang, Jianmo Ni, Nikhil Mehta, Maheswaran Sathiamoorthy, Lichan Hong, Ed Chi, and Derek Zhiyuan Cheng. 2023. Do llms understand user preferences? evaluating llms on user rating prediction. *arXiv preprint arXiv:2305.06474*.

Yehuda Koren. 2009. Collaborative filtering with temporal dynamics. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 447–456.

Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37.

Wenqiang Lei, Xiangnan He, Yisong Miao, Qingyun Wu, Richang Hong, Min-Yen Kan, and Tat-Seng Chua. 2020. Estimation-action-reflection: Towards deep interaction between conversational and recommender systems. In *Proceedings of the 13th international conference on web search and data mining*, pages 304–312.

Chance Jiajie Li, Jiayi Wu, Zhenze Mo, Ao Qu, Yuhan Tang, Kaiya Zhao, Yulu Gan, Jie Fan, Jiangbo Yu, Jinhua Zhao, and 1 others. 2026. Simulating society requires simulating thought. *Advances in Neural Information Processing Systems*, 38.

Jinming Li, Wentao Zhang, Tian Wang, Guanglei Xiong, Alan Lu, and Gerard Medioni. 2023. Gpt4rec: A generative framework for personalized recommendation and user interests interpretation. *arXiv preprint arXiv:2304.03879*.

Yang Li, Kangbo Liu, Ranjan Satapathy, Suhang Wang, and Erik Cambria. 2024. Recent developments in recommender systems: A survey. *IEEE Computational Intelligence Magazine*, 19(2):78–95.

Yiding Liu, Tuan-Anh Nguyen Pham, Gao Cong, and Quan Yuan. 2017. An experimental evaluation of point-of-interest recommendation in location-based social networks.

John D Mayer, Yvonne N Gaschke, Debra L Braverman, and Temperance W Evans. 1992. Mood-congruent judgment is a general effect. *Journal of personality and social psychology*, 63(1):119.

Tien T Nguyen, F Maxwell Harper, Loren Terveen, and Joseph A Konstan. 2018. User personality and user satisfaction with recommender systems. *Information systems frontiers*, 20(6):1173–1189.

Hao Wang, Zonghu Wang, and Weishi Zhang. 2018. Quantitative analysis of matthew effect and sparsity problem of recommender systems. In *2018 IEEE 3rd International Conference on Cloud Computing and Big Data Analysis (ICCCBDA)*, pages 78–82. IEEE.

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

Xinyuan Wang, Liang Wu, Liangjie Hong, Hao Liu, and Yanjie Fu. 2025b. Llm-enhanced user–item interactions: Leveraging edge information for optimized recommendations. *ACM Transactions on Intelligent Systems and Technology*, 16(5):1–24.

Yancheng Wang, Ziyan Jiang, Zheng Chen, Fan Yang, Yingxue Zhou, Eunah Cho, Xing Fan, Yanbin Lu, Xiaojiang Huang, and Yingzhen Yang. 2024. Recmind: Large language model powered agent for recommendation. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 4351–4364.

Ziyi Wang, Yuxuan Lu, Wenbo Li, Amirali Amini, Bo Sun, Yakov Bart, Weimin Lyu, Jiri Gesi, Tian Wang, Jing Huang, and 1 others. 2025c. Opera: A dataset of observation, persona, rationale, and action for evaluating llms on human online shopping behavior simulation. *arXiv preprint arXiv:2506.05606*.

Jun Xiao, Hao Ye, Xiangnan He, Hanwang Zhang, Fei Wu, and Tat-Seng Chua. 2017. Attentional factorization machines: Learning the weight of feature interactions via attention networks. *arXiv preprint arXiv:1708.04617*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Se-eun Yoon, Zhankui He, Jessica Echterhoff, and Julian McAuley. 2024. Evaluating large language models as generative user simulators for conversational recommendation. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1490–1504.

An Zhang, Yuxin Chen, Leheng Sheng, Xiang Wang, and Tat-Seng Chua. 2024a. On generative agents in recommendation. In *Proceedings of the 47th international ACM SIGIR conference on research and development in Information Retrieval*, pages 1807–1817.

Jizhi Zhang, Keqin Bao, Wenjie Wang, Yang Zhang, Wentao Shi, Wanhong Xu, Fuli Feng, and Tat-Seng Chua. 2024b. Prospect personalized recommendation on large language model-based agent platform. *arXiv preprint arXiv:2402.18240*.

Junjie Zhang, Yupeng Hou, Ruobing Xie, Wenqi Sun, Julian McAuley, Wayne Xin Zhao, Leyu Lin, and Ji-Rong Wen. 2024c. Agentcf: Collaborative learning with autonomous language agents for recommender systems. In *Proceedings of the ACM Web Conference 2024*, pages 3679–3689.

Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. 2019. Deep learning based recommender system: A survey and new perspectives. *ACM computing surveys (CSUR)*, 52(1):1–38.

Kesen Zhao, Shuchang Liu, Qingpeng Cai, Xiangyu Zhao, Ziru Liu, Dong Zheng, Peng Jiang, and Kun Gai. 2023. Kuaisim: A comprehensive simulator for recommender systems. *Advances in Neural Information Processing Systems*, 36:44880–44897.

Yong Zheng, Bamshad Mobasher, and Robin Burke. 2015. Carskit: A java-based context-aware recommendation engine. In *2015 IEEE international conference on data mining workshop (ICDMW)*, pages 1668–1671. IEEE.

## A Experimental Setup

Experiments are conducted on four datasets spanning diverse recommendation domains: MovieLens-1M (movies), AmazonBook (books), Steam (video games), and OPeRA (Wang et al., 2025c) (EC site). Following prior work, we filter out users and items with fewer than 20 interactions. We then split interactions into training, validation, and test sets using an 80/10/10 time-based split, so that all test interactions strictly occur after the validation and training interactions. This reflects the temporal distribution shift encountered in real deployments and ensures that simulated agents cannot access future information. In a few experiments, we also report results for CONTEXTSIM(FS), where agents are not pretrained but instead condition their decisions on chain-of-thoughts exemplars appended to prompts as structured *context*. This enables a direct comparison between pretrained agents and those that rely purely on prompt interpretation and in-context reasoning.

**Agent Initialization.** For each user, we instantiate one ContextSim agent. The agent profile $p$ is initialized via the persona-matching procedure described in Section 3.1. Concretely, we provide the LLM a few interactions from its history and prompt it to generate $K = 5$ candidate personas. A separate scoring prompt evaluates the consistency between each candidate persona and the interaction history; the persona with the highest consistency score is selected and fixed for all subsequent simulations. Episodic memory is initialized with the user's training interactions and preference descriptions, while values in the emotional memory are initialized as neutral and updated during simulations.

**Thought Synthesis.** We construct two reasoning datasets, $\mathcal{D}_{\text{PD}}$ (item disentanglement) and $\mathcal{D}_{\text{TA}}$ (trajectory alignment), from the training splits of all datasets. For $\mathcal{D}_{\text{PD}}$, we sample rated/liked/purchased items for each user (up to 50 per user) and prompt the model to explain why the observed action is consistent with the persona $p$, history $H$, and item attributes. For $\mathcal{D}_{\text{TA}}$, we use state-action pairs and alternative navigation actions $A_t$, and ask the model to justify why the historical action $a_t$ is preferable, grounding the explanation in $p$, $H$, and the current state $s_t$. We fine-tune a Qwen3-8B model on $\mathcal{D}_{\text{PD}} \cup \mathcal{D}_{\text{TA}}$. Unless otherwise stated, we train for 5 epochs with a batch size of 16, learning rate $1\times10^{-5}$, AdamW optimizer, and a maximum sequence length of 4,096 tokens. LoRA was applied to all linear modules, while LoRA rank is set to 8 and LoRA alpha is set to 16.

**Life Simulation.** The life simulation module is instantiated using the personas inferred from the training split and is applied independently to each agent. Prompts follow CitySim framework (Bougie and Watanabe, 2025a). For CONTEXTSIM(SIM), we simulate day-by-day schedules during evaluation, discretizing the day into 30-minute slots and conditioning the generated activities on persona attributes, day type (weekday vs. weekend), and sampled external factors (season, weather, local events). At each slot, the module determines whether an RS interaction is likely to occur; if so, it produces a contextual scenario $c = (c_t, c_l, c_s, c_g, c_b)$, which is passed to the interaction policy. For CONTEXTSIM(SUM), we run the life simulation for 30 days per agent before evaluation and summarize contextual statistics (e.g., typical time-of-day and location of usage, recurring goals and constraints) into a short text description that is appended to the persona. All subsequent interactions for CONTEXTSIM(SUM) are conditioned on this summary instead of running the life simulation online.

**Tasks and Metrics.** For rating prediction tasks, we ask agents to assign ratings to held-out user–item pairs and compute RMSE and MAE on the test split. For temporal-pattern analysis, we group interactions into four time-of-day bands (Morning, Afternoon, Evening, Night) using the original timestamps and report click-through rates and Spearman correlations against real data. For action-alignment experiments on OPeRA, we follow the protocol described in Table 14, requiring exact match of action parameters and reporting accuracy and F1 metrics over action types, click subtypes, and session outcomes.

**Baselines.** We compare ContextSim to RecAgent, Agent4Rec, and SimUSER as LLM-powered user simulators, and to RecMind when results are available. All LLM-based baselines are implemented with GPT-4o-mini as in their original papers and use the same training/validation/test splits as ContextSim. For rating prediction, we also include standard RS baselines: Matrix Factorization (MF) and Attentional Factorization Machines (AFM). GPT-4o serves as an automatic judge for persona–action consistency, human-likeness scores, and context-consistency labels. For the offline A/B testing experiments, we use a proprietary dataset of 55 historical experiments from a large-scale food recom-

mendation platform; for each strategy, we run the simulator on the same item pools and compute the Spearman correlation between simulated and real metrics.

**Interactions with RS.** Following persona initialization and thought synthesis, agents interact with the recommender system in a page-by-page manner until a terminal action is selected. In recommendation domains (MovieLens, Steam, Amazon-Book), sessions terminate via `[EXIT]`; while in the web-shopping domain (OPeRA), it may include purchase-related decisions before `[EXIT]`. To improve accuracy, they first assess items on the current pages by assigning to each a `[WATCH]`/`[SKIP]` intention. Next, the agent infers internal state, including fatigue, curiosity, and boredom, from the ongoing session, recent actions, and current context. It then selects an action from the environment action space (e.g., navigating pages, clicking an item, searching, rating, or exiting). Action selection is accompanied by an internal thought that explicitly weighs the available actions against its persona and context $c$, as well as the agent's preferences and prior interactions. After executing the action, the simulator transitions to the next state $s_{t+1}$, and the new interaction is appended to episodic memory. Following each step, the agent performs a short self-reflection that summarizes the rationale behind the action (and any expressed tastes), storing this concise explanation in the episodic memory.

## B Discussion

Our study shows that grounding LLM-based user agents in a realistic daily-life context, along with explicit thought synthesis, yields synthetic users that more closely match human preferences and interaction dynamics, and provides more reliable evaluation signals for recommender systems. Despite these improvements, several limitations remain.

First, the current framework represents context and interface states primarily through text. While this abstraction enables a simplified implementation across domains, it may omit fine-grained cues that drive real decisions, such as item thumbnails or user experience. Extending ContextSim to multimodal observations and structured interface representations (e.g., interface screenshot) could improve both the realism of user actions.

Second, our thought-synthesis supervision is derived from historical interactions and their associated outcomes, which provide only partial coverage of the space of plausible contexts and behaviors. Although the life simulation module increases diversity by exposing agents to a broader range of temporal, spatial, and situational conditions, the induced contexts are still generated by a model and may fail to capture rare events, abrupt preference shifts, or hard-to-capture constraints present in real life (e.g., unexpected schedule changes, social commitments). Improving the realism of simulated contexts remains an important research direction.

Third, our persona representation includes Big Five traits to capture relatively stable behavioral tendencies that may affect how users engage with recommendations beyond their item preferences. Prior work has linked personality to recommender-system behavior (Nguyen et al., 2018), and recent user simulators similarly incorporate personality attributes (Bougie and Watanabe, 2025b). For example, openness may influence willingness to explore unfamiliar items, while conscientiousness may affect how carefully users browse and compare alternatives. However, such traits are rarely directly observed in recommendation datasets and must therefore be inferred from interaction histories. We evaluate this inference procedure on the Personality 2018 MovieLens dataset in Appendix F.1.

Fourth, our life simulation module currently summarizes context over a fixed horizon and uses either online generation (CONTEXTSIM(SIM)) or an offline summary (CONTEXTSIM(SUM)). While the summary variant is computationally efficient and performs well empirically, it can blur short-term fluctuations that affect decisions at the session level (e.g., late-night fatigue or time-limited shopping). Conversely, the online variant increases fidelity but introduces additional variance and may require more interactions to faithfully reproduce human action distributions.

Fifth, some evaluations rely on an LLM as a judge, which may introduce evaluator-specific bias or noise (Table 6). We partially mitigate this limitation by providing explicit behavioral evidence to the judge and, for human-likeness, validating LLM-based rankings against human pairwise preferences (Section 4.8). Future work should extend this validation with larger-scale human annotations and multiple evaluator models.

Finally, our evaluation demonstrates strong improvements on domains with moderate temporal depth and session-like interactions. Deploying ContextSim in longer-horizon settings, such as continu-

| Evaluation | Section | Metric | Basis | LLM judge |
|---|---|---|---|---|
| Preference Alignment | 4.1 | Acc. / Prec. / Rec. / F1 | Real interactions (binary labels) | No |
| Rating Prediction | 4.2 | RMSE / MAE | Held-out ratings | No |
| Thought Consistency | 4.3 | Consistency / Contradiction | GPT-4o judgment | Yes |
| Human-likeness | 4.4 | Likert score | GPT-4o judgment | Yes |
| Offline A/B Testing | 4.5 | Spearman correlation | Real A/B test outcomes (55 experiments) | No |
| RS Parameter Optimization | 4.6 | Engagement metrics | Real deployment results | No |
| Action Alignment (OPeRA) | App. F.5 | Accuracy / F1 | Ground-truth actions (exact match) | No |
| Context Consistency | App. F.6 | Alignment labels | GPT-4o judgment | Yes |
| Temporal Patterns | App. F.8 | CTR / Correlation | Real timestamps | No |
| Human Preferences | App. F.12 | Preferences | Human judgment | No |
| Rating Distribution | App. F.7 | Distribution alignment | Ground-truth rating distribution | No |
| Prompt Robustness | App. F.15 | RMSE / Action F1 / Temporal corr. | Held-out ratings, ground-truth actions, and real timestamps | No |

Table 6: Summary of the evidence used in each evaluation.

ous media feeds or mobile app usage, may require additional components, including persistent preference evolution or hierarchical goal management. This direction is especially important if the simulator is used not only to compare fixed recommenders, but also to optimize adaptive policies that influence users over time.

### B.1 Running Time and Cost

We compare the running time of CONTEXTSIM and SIMUSER for 1,000 user interactions. As reported in SIMUSER, it performs API calls to GPT-4o-mini and requires 10.1h for 1,000 interactions without parallelization, corresponding to approximately \$16–\$21 in API cost under current pricing. In contrast, CONTEXTSIM primarily performs inference with a locally served Qwen3-8B policy (vLLM). Using 4-GPU, this yields an estimated runtime of ≈1.3h for 1,000 interactions, corresponding to roughly \$13–\$16 in GPU time. Overall, CONTEXTSIM is faster than GPT-based simulators and offers competitive or lower cost while avoiding per-call API overhead. In addition, our method is privacy-preserving as all interactions are simulated locally without transmitting user data to external APIs.

## C Datasets

**MovieLens-1M.** MovieLens-1M is a widely adopted benchmark for recommender-systems research. It contains approximately one million explicit ratings on a 1–5 star scale, collected from 6,040 users across 3,706 movies. The dataset also provides auxiliary information, including movie titles and genre annotations, as well as basic user demographics such as age, gender, and occupation.
**Steam.** The Steam dataset comprises user–game interaction records from the Steam platform. It includes user and game identifiers together with English-language user reviews, and provides game-level metadata such as titles.

**AmazonBook.** AmazonBook is a subset of the Amazon product reviews corpus limited to the Books category. It consists of user–item interactions in the form of ratings and textual reviews, accompanied by item metadata including book titles and category labels.

**OPeRA.** OPeRA is a dataset developed to evaluate large language models for simulating human online shopping behavior. It contains real-world shopping sessions that integrate survey-based persona information, observations of webpage content, fine-grained user actions (e.g., clicks and navigation), and self-reported rationales explaining user decisions.

## D Simulation Environment

Our simulator mirrors real-world recommendation platforms like Netflix and Steam, functioning in a page-by-page manner. Users are initially presented with a list of item recommendations on each page: (i) recommendations for MovieLens, Steam, and AmazonBook, or (ii) a web-shopping page for OPeRA. The recommendation algorithm is structured as a standalone module, allowing including any algorithm. This design features pre-implemented collaborative filtering-based strategies, including random, most popular, Matrix Factorization, LightGCN, and MultVAE.

Namely, in **recommendation domains**, the environment displays a *page* of $M$ recommended items as a single text state $s_t$. For each item, the state includes its title and an item description. The short description is either taken from available domain metadata (when present) or retrieved from the title. If the agent clicks on an item, the simulator reveals a more detailed description for that item in the next state. We format each page as:

**Page Format (Recommendation Domains)**

PAGE {page_number}
CONTEXT: {user_context}
<− {item_title} −> <− History ratings: {item_rating} −> <− Summary: {item_description} −>
<− {item_title} −> <− History ratings: {item_rating} −> <− Summary: {item_description} −>
...

Here, {user_context} is the contextual scenario used by ContextSim (temporal, spatial, situational, goal, and constraint factors). {item_rating} is the agent's own historical rating when available; otherwise, it defaults to a dataset-derived statistic (e.g., global mean rating).

The environment supports the following actions: `[NEXT_PAGE]`: advance to page (`page_number` + 1). `[PREVIOUS_PAGE]`: go back to page (`page_number` − 1) when `page_number` > 1.`[CLICK_ITEM:<item_id>]`: show the detailed description for the selected item in the next state, `[SEARCH]`: given query search for specific items, `[RATE]`, and `[EXIT]`: terminate the session.

In **web-shopping domains** like OPeRA, each state includes (i) page context, (ii) a product list with attributes that appear in the observation, and (iii) a list of interactive elements identified by semantic IDs:

**Page Format (Web-Shopping Domains)**

PAGE {page_number}
USER CONTEXT: {user_context}
CONTEXT: {page_context}
PRODUCTS:
<− {product_title} −> <− Price: {price} −> <− Details: {short_description} −>
<− {product_title} −> <− Price: {price} −> <− Details: {short_description} −>
...
INTERACTIVE ELEMENTS (semantic IDs):
{semantic_id_1}, {semantic_id_2}, ..., {semantic_id_L}

where the page context features the web-shopping information exposed by OPeRA (e.g., the current page type such as search, cart contents, cart price, and other page-specific cues). Actions follow the same action space as defined in the OPeRA dataset (Wang et al., 2025c), extended with the navigation actions described above.

# E Prompts

## E.1 Post-Interview Prompt

After completing an interaction session, each agent is queried using a post-interview prompt designed to assess its overall satisfaction with the recommender system. The exact prompt used for this evaluation is presented below:

**Post-Interview Prompt**

How satisfied are you with the recommender system you recently interacted with?
**### Instructions:**
1. Rating: Provide a rating from 1 to 10.
2. Explanation: Explain the reason for your rating.

**### Response Format:**
- RATING: [integer between 1 and 10]
- REASON: [detailed explanation]

## E.2 Believability of Synthetic User Prompt

To evaluate the believability of synthetic users, as described in Section 4.1, we adapt the post-interview setup by introducing additional instructions focused on prior interactions with recommended items. The modified prompt is provided verbatim below:

**Believably of Synthetic User Prompt**

**### Instructions**
1. Review each {item_type} in the ## Recommended List ##.
2. For each {item_type}, classify if you have already interacted with it ("Interacted") or if you have not ("Not Interacted").

## E.3 LLM Evaluator Prompt

To assess whether interaction traces resemble those of real users or are indicative of AI-generated behavior, we employ an external LLM-based evaluator. This evaluator is prompted as follows:

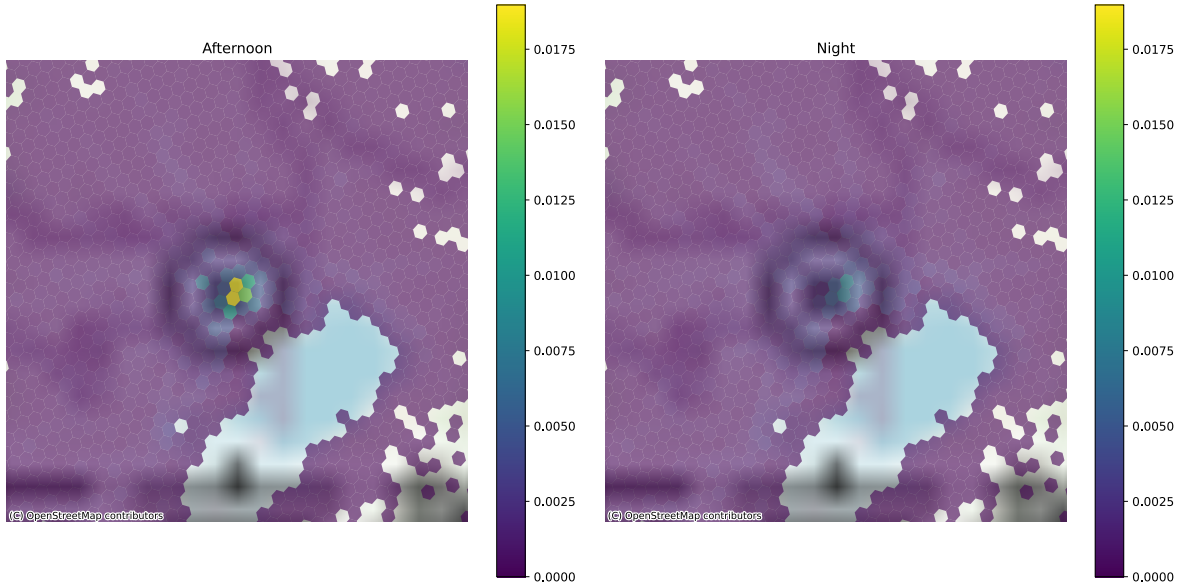


Figure 5: Estimated probability of user interactions with the recommender system across H3 tiles in the Tokyo metropolitan area during the afternoon (left) and night (right). Ocean areas are represented in light blue.

| History length | Average MAE ↓ | Average Pearson $r$ ↑ |
|---|---|---|
| 10 interactions | 0.274 | 0.33 |
| 25 interactions | 0.201 | 0.47 |
| 50 interactions | **0.152** | **0.61** |

Table 7: Accuracy of inferred Big Five traits at different interaction-history lengths on MovieLens.

| Configuration | Accuracy (1:1) ♦ | RMSE ♦ | Consistency ♠ | Temporal Correlation ♦ |
|---|---|---|---|---|
| ContextSim (full) | 0.824 | 0.451 | 84.1% | 0.94 |
| - Life Simulation | 0.798 | 0.489 | 63.8% | 0.31 |
| - Thought Synthesis | 0.812 | 0.468 | 46.4% | 0.88 |

Table 8: Component ablation study. We report user preference accuracy, rating RMSE, thought consistency, and temporal correlation, on MovieLens ♦ and OPeRA ♠.

**LLM Evaluator Prompt**

Please evaluate the following interactions of an agent with a recommender system, and determine whether it is generated by a Large Language Model (LLM) AI or a real human:
{interaction logs}

Please rate on a scale of 1 to 5, with 1 being most like an AI and 5 being most like a human.

# F Additional Experiments

## F.1 Persona Inference Validation

Although personality traits provide relatively stable behavioral signals that can complement item preferences, we do not assume that they can be perfectly recovered from recommendation histories. We therefore evaluate our persona-inference procedure using the Personality 2018 MovieLens dataset, which contains user interaction histories along with Big Five personality scores. We vary the number of historical interactions available to the inference procedure and compare the inferred traits with the questionnaire scores. Table 7 reports MAE and Pearson correlation averaged across the five personality dimensions. As the available interaction history increases, trait inference becomes substantially more accurate.

## F.2 Ablation Studies

We report ablation results in Table 8, evaluating preference accuracy, rating error, consistency, and temporal alignment. Removing the life simulation module leads to a sharp drop in temporal correlation, indicating that realistic daily schedules are essential for reproducing time-of-day interaction patterns. In contrast, ablating the thought-synthesis module preserves overall accuracy and RMSE but substantially degrades internal consistency, showing that explicit reasoning is critical for generating coherent, persona-aligned trajectories. Together, these results demonstrate that both components contribute complementary aspects of

| Backbone | MovieLens | AmazonBook | Steam | OPeRA |
|---|---|---|---|---|
| ContextSim (Llama-3.2-3B) | 4.15 ± 0.18 | 4.00 ± 0.19 | 4.02 ± 0.21 | 3.96 ± 0.20 |
| ContextSim (Qwen-2.5-7B) | 4.35 ± 0.16 | 4.10 ± 0.18 | 4.12 ± 0.20 | 4.03 ± 0.19 |
| ContextSim (Llama-3.1-8B) | 4.47 ± 0.15 | 4.20 ± 0.17 | 4.24 ± 0.20 | 4.12 ± 0.18 |
| ContextSim (Qwen3-8B) | **4.60 ± 0.14** | **4.28 ± 0.16** | **4.31 ± 0.19** | **4.21 ± 0.18** |

Table 9: Human-likeness scores of CONTEXTSIM with different backbone LLMs, evaluated by GPT-4o across four recommendation domains.

| Context Factors | Rating error | Temporal correlation |
|---|---|---|
| None | 0.489 | 0.23 |
| Time only | 0.472 | 0.71 |
| Time + Location | 0.464 | 0.82 |
| Time + Location + Situation | 0.458 | 0.89 |
| Full context | 0.451 | 0.94 |

Table 10: Impact of individual context factors on MovieLens, with rating error measured by RMSE.

fidelity: life simulation governs *when* users act, whereas thought synthesis governs *how* they behave.

### F.3 Interaction Location

Next, we examine *where* users are likely to engage with recommender systems. We simulate life in the Tokyo area and construct spatial probability maps over H3 tiles, indicating the likelihood that a user will engage with the RS. We report results in Figure 5. Afternoon interactions (12–18) are sharply concentrated around the central business district, reflecting well-known commuting patterns and the higher density of workplaces. In contrast, nighttime interactions (00–06) remain centered in Tokyo but also spread outward into suburban and residential areas. This shift highlights that user attention is shaped not only by temporal context but also by the mobility structure of daily life, impacting their potential needs and time constraints.

### F.4 LLM Backbone Choice

To study the impact of the backbone LLM, the base model is swapped while keeping the rest of CONTEXTSIM unchanged. As shown in Table 9, all backbones achieve high human-likeness scores ($\geq$ 4), indicating that the proposed life-context grounding and thought synthesis are robust to the choice of underlying model. Qwen3-8B yields the strongest results overall, but the performance gaps to Llama-3.1-8B and Qwen-2.5-7B are modest, suggesting that our approach is reasonably robust to the backbone choice.

| Context factor | Aligned | Partially aligned | Contradictory |
|---|---|---|---|
| Temporal $c_t$ | 0.69 | 0.23 | 0.08 |
| Spatial $c_l$ | 0.64 | 0.27 | 0.09 |
| Situational $c_s$ | 0.55 | 0.32 | 0.13 |
| Goal $c_g$ | 0.72 | 0.21 | 0.07 |
| Constraint $c_b$ | 0.58 | 0.30 | 0.12 |

Table 11: LLM-based consistency between simulated contexts and OPeRA survey answers. Results are averaged over 5,000 interaction samples.

### F.5 Impact of Contextual Factors

We further measure which contextual signals matter most. Table 10 highlights how each additional factor improves both rating accuracy and temporal alignment (MovieLens). Using only temporal context already yields a substantial gain in temporal. Adding location and mood progressively closes the gap, with full context achieving the best RMSE and temporal correlation. These results indicate that both when users interact (time), where they are (location), and why they act jointly shape realistic interaction patterns. As expected, rating accuracy is primarily driven by user preferences, which explains the comparatively smaller improvements from contextual factors alone. Static attributes mainly capture persistent user preferences, whereas dynamic context is particularly important for modeling when and under what circumstances those preferences are expressed.

### F.6 Context Consistency

We next examine whether the contextual factors are consistent with real-world contexts provided in OPeRA (Wang et al., 2025c). We run the daily-life module for 30 days, and collect the aggregated $c$. For evaluation, we randomly sample 5,000 interaction points across all agents. For each sampled context and each dimension (temporal, spatial, situational, goal, constraint), we provide an LLM judge with: (i) the user's persona, (ii) the textual description of the simulated context, and (iii) the ground truth from OPeRA. The judge is asked to label, independently for each dimension, whether the simulated context is *aligned*, *partially aligned*, or *contradictory* to the survey answers. As illustrated in Table 11, contextual factors are largely coherent with the ground-truth. Temporal and goal-related factors exhibit the highest alignment, while situational and constraint factors show slightly more variability but remain largely consistent.

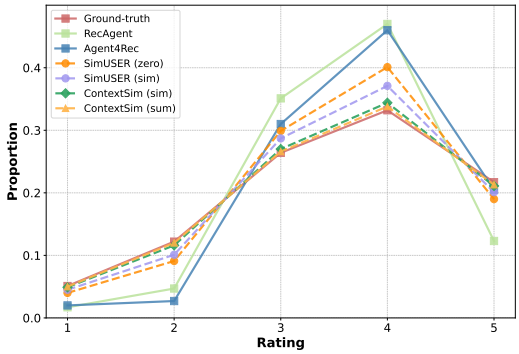


Figure 6: Comparison of rating distributions between ground-truth and human proxies.

| Time Period | Real | Agent4Rec | SimUSER | ContextSim(sim) |
|---|---|---|---|---|
| Morning (6-12) | 0.11 | 0.25 | 0.25 | 0.16 |
| Afternoon (12-18) | 0.21 | 0.25 | 0.25 | 0.24 |
| Evening (18-24) | 0.35 | 0.25 | 0.25 | 0.37 |
| Night (0-6) | 0.28 | 0.25 | 0.25 | 0.23 |

Table 12: Temporal click-through rate patterns.

### F.7 Rating Distribution

Beyond individual rating alignment, human proxies must replicate real-world behavior at the macro level. This implies ensuring that the distribution of ratings generated by the agents matches the distributions observed in the original dataset. Figure 6 presents the rating distribution from the MovieLens dataset and the ratings generated by different simulators. These results reveal a high degree of alignment between the simulated and actual rating distributions, with a predominant number of ratings at 4 and a small number of low ratings (1–2). While Agent4Rec and SimUSER assign fewer low ratings than real users, CONTEXTSIM(SIM) and CONTEXTSIM(SUM) more closely match the true distribution, indicating improved alignment at the population level.

### F.8 Time-Aware Interaction Patterns

In this study, we postulate capturing contextual patterns present in real user behavior but absent from context-free simulators. Using MovieLens interactions with timestamps, we group clicks into four time-of-day bands and compare temporal click-through rates. Table 12 demonstrates that CONTEXTSIM(SIM) accurately captures the evening peak in engagement and reduced night activity, while baseline methods use unrealistic uniform patterns. The correlation between ContextSim and real temporal patterns is higher compared to SimUSER.

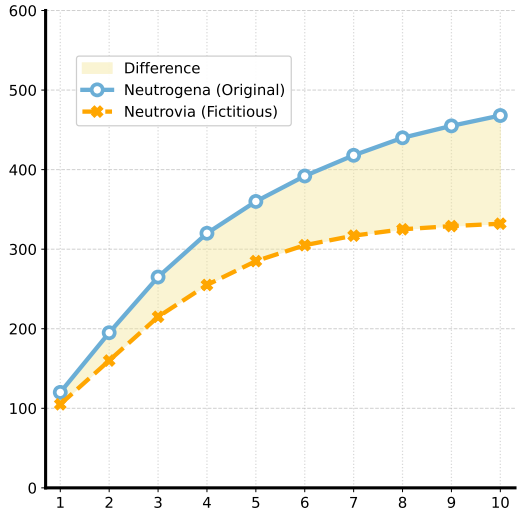


Figure 7: The phenomenon observation of Brand Loyalty.

### F.9 Brand Loyalty

Next, we analyze the proportion of interactions each item received at the final stage of the simulation relative to the total number of interactions, as reported in Figure 7. Because our evaluation includes web-shopping trajectories (OPeRA), where items explicitly expose brand names (e.g., titles and product-page content), brand cues can directly influence agent decisions. We noticed that brand-identified items were significantly more popular than their counterparts. To further examine this effect, we replace the brand name *"Neutrogena"* with a fictitious alternative, *"Neutrovia"*, while keeping all other item attributes unchanged, including recommendation probability and non-brand content. Here, a step corresponds to one interaction round in the simulator. Figure 7 suggests that removing the recognizable brand cue substantially reduces engagement: after 10 steps, the original-brand item accumulates 468 likes, compared to 332 for the fictitious-brand variant.

### F.10 Matthew Effect

We quantify the *Matthew Effect* (Wang et al., 2018) in our simulator by introducing a small early exposure advantage to a single target product and measuring whether this initial advantage compounds over repeated interaction rounds. We select `Neutrogena Make-Up Remover Cleansing Towelettes`, and evaluate two settings: (i) Original, where the product is ranked according to the recommender as usual, and (ii) Boosted, where the same product receives a slight exposure advantage

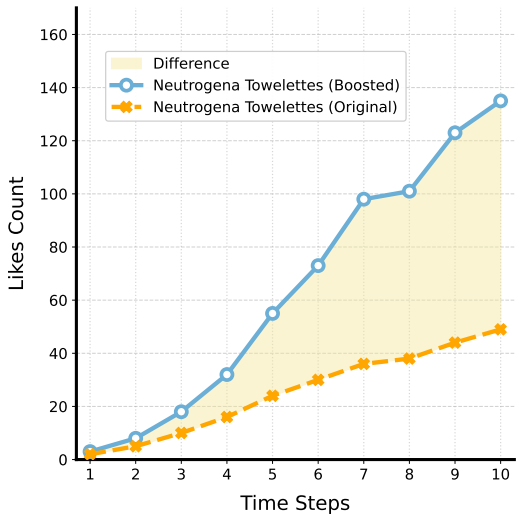


Figure 8: The phenomenon observation of Matthew effect.

only during the first two interaction rounds, after which the recommendation policy is identical to the Original condition. We track the cumulative number of positive interactions ("likes") received by the target product over steps in both conditions. As shown in Figure 8, a small initial exposure boost results in a persistent and widening gap in cumulative likes, even after the boost is removed, consistent with the Matthew effect in simulated interactions.

### F.11 Generalization to Unfamiliar Items

In this experiment, we target items that are likely unfamiliar to the backbone LLM, in order to evaluate whether thought synthesis improves generalization beyond the training distribution. Following the rating evaluation in Table 2, including a few-shot prompting version of our framework, CONTEXTSIM(FS). Concretely, we query the backbone LLM to classify each movie into one of the dataset genres; items whose predicted genre does not match the ground-truth label are treated as unfamiliar and included in the subset, while correctly classified items are excluded. Figure 10 shows that RMSE increases for all methods under hallucination, as expected. However, CONTEXTSIM is the most robust, exhibiting the smallest degradation relative to its original performance. In contrast, the few-shot variant, CONTEXTSIM(FS), degrades more sharply and becomes comparable to SIMUSER on unfamiliar items, consistent with the brittleness of few-shot prompting — when the model cannot anchor decisions in stable learned reasoning. Overall, these results highlight that pretraining improves generalization by teaching the policy to discover preference-relevant attributes and how preferences align with its persona.

| Prompt setting | MovieLens RMSE | OPeRA Action F1 | MovieLens Temporal corr. |
|---|---|---|---|
| Original prompts | 0.451 | 64.22 | 0.94 |
| Mean over prompt variants | 0.459 | 63.61 | 0.91 |
| Worst prompt variant | 0.470 | 62.77 | 0.89 |

Table 13: Prompt robustness analysis. The model is fixed, only prompt wording is varied.

### F.12 Impact of Situational Context

We further investigate how situational context, specifically mood and recent activities, influences user engagement with recommendations. Using MovieLens, we report the average rating conditioned on each contextual state. As reported in Figure 9, both factors significantly affect predicted ratings. Positive moods (happy, relaxed) increase ratings compared to negative states (stressed, sad), consistent with evaluation documented in psychology literature (Mayer et al., 1992). Bored users exhibit moderately high ratings, likely reflecting heightened receptivity to novel entertainment. Similarly, leisure-oriented activities (social gatherings, rest) yield higher ratings than cognitively demanding tasks (work, chores), suggesting users are more receptive to recommendations after relaxation. These findings demonstrate that ContextSim captures nuanced psychological effects that static preference models ignore. By grounding agents in a realistic situational context, our framework enables more accurate simulation of context-dependent user behavior, which is critical for evaluating RS performance across diverse real-world scenarios.

### F.13 Prompt Robustness

Since several components of CONTEXTSIM are prompt-driven, we examine whether the results depend on a single hand-crafted prompt design. We keep the fine-tuned policy and vary only the wording of prompts used by the persona initialization, life simulation, action selection, and reflection modules. Specifically, we create ten prompt variants that preserve the same task definition but differ in instruction order, phrasing, and the amount of explanation requested from the agent. For each variant, we rerun the simulator on the same subset of agents and recommendation states, using the same random seeds. We then report performance variation across prompt variants. As shown in Table 13, performance varies only moderately across

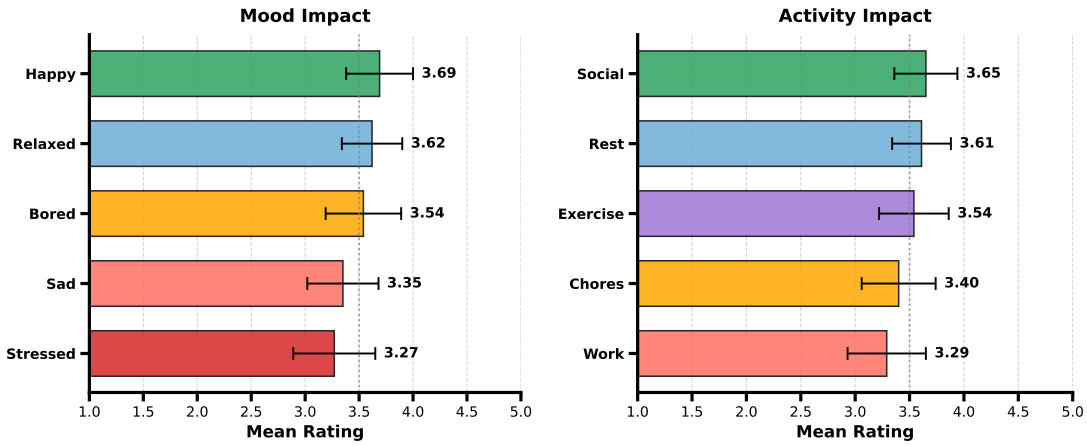


Figure 9: Impact of situational context on predicted ratings on MovieLens.

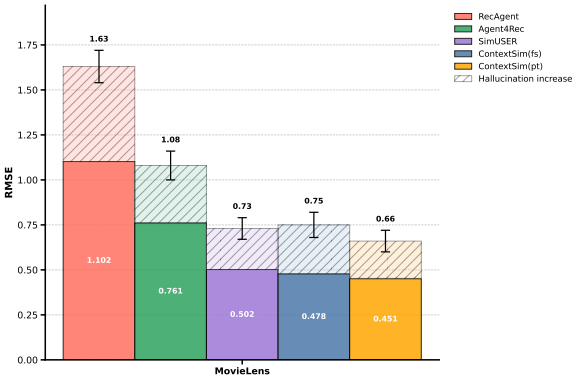


Figure 10: Comparison of RMSE values for the standard rating task (dark bars) and the hallucination subset (dark+light stacked bars) on MovieLens.

prompt variants. This suggests that the gains are not driven by a single fragile prompt template. At the same time, prompt design remains a source of variance, especially for modules that generate context and reflections. We therefore release all prompt templates and report this analysis to make the prompt dependence of the framework explicit.

### F.14 Action Alignment

Using the OPeRA dataset, action alignment is measured with an exact-match criterion: a prediction is counted as correct only if all action parameters match the ground-truth specification. For `click` actions, this requires matching the clicked target (e.g., the correct product or button). For `input` actions, the model must identify both the appropriate input field and the exact text entered by the user. We also assess how well each approach classifies action types. We report F1 scores for the high-level action categories `click`, `input`, and `terminate`. To probe fine-grained behavior, we further compute weighted F1 over `click` subtypes, capturing whether the model can distinguish between different click intents (e.g., `review`, `product_link`, `purchase`). Given that online shopping is inherently goal-driven, we evaluate the prediction of session outcomes. Each session terminates either with a `purchase-related click` or a `terminate` action. Performance on these terminal actions is assessed using accuracy and weighted F1, reflecting how well the model captures users' final decisions and long-term goals over the course of a session. As presented in Table 14, CONTEXTSIM substantially outperforms the baselines, with especially large gains in action generation accuracy and session-outcome prediction.

### F.15 Quality of Thought-Synthesis Data

Since the thought-synthesis rationales are LLM-generated and no ground-truth rationale is available, we evaluate the quality of the generated supervision. We leverage Gemini 3.5 as an evaluator from a different model family than GPT, which generates the rationales. For each rationale, the evaluator assigns a score from 1 to 5 along four dimensions: alignment with the persona and interaction history, grounding in the provided evidence, specificity, and justification of the observed action. For trajectory-level rationales, the last dimension additionally considers whether the rationale explains why the observed action is preferred over the available alternatives. As depicted in Table 15, both item and trajectory-level rationales receive consistently high scores across all dimensions.

| Model | Action Gen. (Accuracy) | Action Type (Macro F1) | Click Type (Weighted F1) | Session Outcome (Weighted F1) |
|---|---|---|---|---|
| GPT-4.1 | 21.51 | 48.78 | 44.47 | 47.54 |
| w/o persona | 22.06 | 45.55 | 43.45 | 58.47 |
| w/o rationale | 21.28 | 34.93 | 42.63 | 51.17 |
| Claude-3.7 | 10.75 | 31.58 | 27.27 | 43.52 |
| w/o persona | 10.75 | 25.33 | 22.76 | 43.10 |
| w/o rationale | 10.08 | 26.06 | 20.29 | 43.10 |
| Llama-3.3 | 8.31 | 24.29 | 19.99 | 36.64 |
| w/o persona | 8.31 | 23.69 | 18.59 | 33.21 |
| w/o rationale | 8.76 | 23.60 | 19.22 | 34.19 |
| RecAgent | 22.71 | 49.18 | 45.25 | 54.12 |
| Agent4Rec | 23.09 | 50.05 | 46.37 | 56.70 |
| SimUSER | 24.21 | 52.44 | 48.68 | 59.63 |
| ContextSim | **35.37** | **64.22** | **61.03** | **73.74** |

Table 14: Evaluation of next-action prediction. We report four metrics: action generation accuracy, action type macro F1, click type weighted F1, and session outcome weighted F1. "Claude-3.7" denotes Claude-3.7-Sonnet; "Llama-3.3" denotes Llama-3.3-70B-Instruct. All metrics are percentages (%).

| Quality dimension (1–5 ↑) | Item-level | Trajectory-level |
|---|---|---|
| Alignment | 4.5 | 4.3 |
| Evidence grounding | 4.6 | 4.4 |
| Specificity | 4.2 | 4.0 |
| Action justification | 4.4 | 4.2 |

Table 15: Quality of generated thought-synthesis supervision, evaluated by Gemini 3.5.

| Variant | MovieLens RMSE ↓ | OPeRA Action-Type Macro F1 ↑ |
|---|---|---|
| Base Qwen3-8B (no SFT) | 1.374 | 25.83 |
| Action-only SFT | 0.468 | 52.19 |
| ContextSim w/o life context | 0.489 | 57.68 |
| ContextSim | **0.451** | **64.22** |

Table 16: Impact of thought synthesis and life context using the same Qwen3-8B backbone.

### F.16 Impact of Thought Synthesis

We further isolate the effect of thought synthesis using Qwen3-8B for all variants. In addition to the base model without fine-tuning, we train an action-only variant using the same observed trajectories and training budget, but without rationale supervision. We also report ContextSim without life context. Table 16 shows that action-only SFT substantially improves over the base model, but remains below ContextSim, particularly for action prediction. This suggests that learning both the observed action and its rationale provides a stronger supervision signal than action prediction alone.

### F.17 Backbone Comparison

Finally, as in most prior studies, the primary comparison uses different LLM backbones across methods: the baselines use GPT-4o-mini, while CONTEXTSIM uses Qwen3-8B. To control for this factor, we additionally evaluate all methods with Qwen3-8B while preserving their original prompt, persona, and memory designs. As shown in Table 17, CONTEXTSIM remains strongest on both rating prediction and action alignment, suggesting that its gains are not solely due to backbone choice.

| Method | Backbone | MovieLens RMSE ↓ | OPeRA Action-Type Macro F1 ↑ |
|---|---|---|---|
| RecAgent | Qwen3-8B | 1.318 | 28.57 |
| Agent4Rec | Qwen3-8B | 1.041 | 31.35 |
| SimUSER | Qwen3-8B | 0.713 | 35.83 |
| ContextSim | Qwen3-8B | **0.451** | **64.22** |

Table 17: Comparison of user simulators using a shared Qwen3-8B backbone.

## G Reproducibility

For transparency and reproducibility, we release all prompts used in our framework, including those related to persona initialization, life simulation, decision-rationale synthesis, interaction, and evaluation prompts, in a public repository at `https://github.com/simul-ci/context_sim`. The repository further contains comprehensive documentation covering data preprocessing procedures, hyperparameter configurations, and prompt specifications.