

AlignUSER: Human-Aligned LLM Agents via World Models for Recommender System Evaluation

Nicolas Bougie¹, Gian Maria Marconi¹, Xiaotong Ye¹, Narimasa Watanabe¹
{nicolas.bougie,gianmaria.marconi,tony.yip,narimasa.watanabe}@woven.toyota

¹Woven by Toyota

Abstract

Evaluating recommender systems remains challenging due to the gap between offline metrics and real user behavior, as well as the scarcity of interaction data. Recent work explores large language model (LLM) agents as synthetic users, yet they typically rely on few-shot prompting, which yields a shallow understanding of the environment and limits their ability to faithfully reproduce user actions. We introduce ALIGNUSER, a framework that learns *world-model-driven* agents from human interactions. Given rollout sequences of actions and states, we formalize world modeling as a next state prediction task that helps the agent internalize the environment. To align actions with human personas, we generate counterfactual trajectories around demonstrations and prompt the LLM to compare its decisions with human choices, identify suboptimal actions, and extract lessons. The learned policy is then used to drive agent interactions with the recommender system. We evaluate ALIGNUSER across multiple datasets and demonstrate closer alignment with genuine humans than prior work, both at the micro and macro levels.

1 Introduction

Recommender systems (RS) are central to many online services, from e-commerce to media platforms, where they personalize content and drive user engagement (Li et al., 2024). Despite significant progress in user preference modeling, evaluation remains a bottleneck (Yoon et al., 2024). Offline metrics (e.g., nDCG, Recall) computed on static datasets dominate current evaluation practices, yet are often misaligned with online behavior once a model is deployed (Zhang et al., 2019; Jannach and Jugovac, 2019). Besides, these metrics do not translate to business values such as sales or satisfaction (Jannach and Jugovac, 2019). On the other hand, online A/B tests offer more faithful

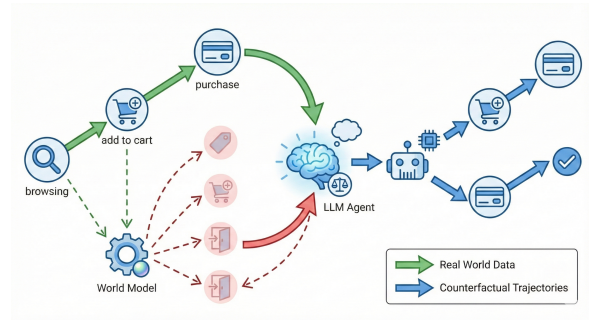


Figure 1: ALIGNUSER learns aligned synthetic users by combining world-model pretraining (a transition model from rollout tuples (s_t, a_t, s_{t+1}^*)) with counterfactual reflection over human trajectories.

feedback but are expensive, slow to iterate, and constrained by ethical and privacy considerations.

A promising alternative is to leverage LLM-based agents as synthetic users that interact with recommender systems in a simulation (Bougie and Watanabe, 2025c). These agents can express rich preferences and feedback in natural language, potentially approximating user-level metrics such as satisfaction or perceived relevance (Hou et al., 2024; Zhang et al., 2023; Huang et al., 2023; Wang et al., 2023c; Yoon et al., 2024). However, most existing approaches rely on few-shot prompting to mimic human behavior. The agent is typically asked to “act like a typical user” given a handful of examples (Wang et al., 2025; Bougie and Watanabe, 2025c), but has no explicit grasp of how the environment evolves in response to its actions. Thus, the agent gains only a superficial understanding of the world and struggles to faithfully reproduce human trajectories, especially when reasoning about long-term consequences (e.g., when to add to cart or exit). Furthermore, without explicit alignment, the agent primarily projects its own intrinsic biases, rather than letting the persona realistically guide its decisions (Salecha et al., 2024; Kaiser et al., 2025; Bisbee et al., 2024).

In this paper, we postulate that agents should understand how the world works in order to faithfully replicate user actions. In light of this, given rollout trajectories (e.g., browsing, searching, adding items to the cart,...), we first pretrain the agent policy on a *world-model* task that predicts the next state from a state–action pair. This task helps the agent internalize environment dynamics: what happens if it clicks on this item, goes to the next page, or decides to leave. To align agents with their human counterparts, we further generate counterfactual trajectories around demonstrations. For each state, we consider alternative actions, roll out their consequences, and prompt the LLM to compare them with the human action, identify suboptimal decisions, and extract insights to guide future choices. This reflection process yields a policy that is explicitly trained to align with human decisions while being aware of environment dynamics. At test time, the learned policy drives the agent’s interactions with recommender systems. We evaluate ALIGNUSER on several datasets and show that it achieves closer alignment with humans than prior LLM-based user agents, both at the micro level, while providing more reliable guidance for RS selection than traditional offline metrics and A/B testing.

2 Related Work

Evaluation of Recommender Systems. Traditional recommender system evaluation predominantly relies on offline metrics such as nDCG, Recall, or RMSE computed on historical logs (Zhang et al., 2019; Jannach and Jugovac, 2019). Although useful for model selection, these metrics do not directly capture user experience or business values, and their correlation with online A/B tests is often weak (Jannach and Jugovac, 2019; Bougie and Watanabe, 2025c). Recent work thus explores interactive and counterfactual evaluation, including bandit simulators, user models, and causal inference techniques (Li et al., 2024).

LLM-based Agents. Recently, LLMs have opened new possibilities for simulating human-like agents in virtual worlds (Park et al., 2023; Li et al., 2023; Wei et al., 2022). LLM-powered agents can reason, plan, and interact through natural language (Wei et al., 2022; Bougie and Watanabe, 2025b; Park et al., 2023; Bougie and Watanabe, 2025a). Several studies harness LLMs as user simulators or conversational agents in recommendation

settings. RecMind (Wang et al., 2023c) and InteRecAgent (Huang et al., 2023) propose planning and reflection mechanisms over tool-augmented agents. Agent4Rec (Zhang et al., 2023) and related work (Hou et al., 2024; Yoon et al., 2024) investigate generative user agents that interact with recommender models and provide ratings or textual feedback. Recently, (Bougie and Watanabe, 2025c) consider image-driven sensing and advanced reasoning modules to align agents with their human counterparts. Although these systems exhibit positive correlations with online a/b tests (Bougie and Watanabe, 2025c), they typically treat the agent policy as a black box mapping from a textual state to an action in a single step, without an explicit model of how actions shape future states. Moreover, the agent is typically instructed to act given its persona, examples, and demographic attributes, which produces behavior reflecting the model’s priors rather than genuine user decision patterns.

World Models World models have a long history in reinforcement learning as predictors of future states and rewards (Ha and Schmidhuber, 2018). Deep latent world models like Dreamer demonstrate that learned dynamics can support imagination and decision-making at scale (Hafner et al., 2019). Recent work extends these ideas to language agents by treating states and actions as text and learning next-state predictors (Kim et al., 2022). Self-reflection strategies such as STaR (Zelikman et al., 2022) and more recent approaches (Wang et al., 2023b) leverage chain-of-thought explanations to improve reasoning and robustness. Most closely related to our work, recent “early experience” method (Zhang et al., 2025) trains LLM agents by generating alternative trajectories and comparing expert actions to alternatives using environment feedback. We present a similar philosophy but target the alignment of user agents with human RS behavior, introduce persona-driven reflection to align persona with actions, and couple world-model-guided counterfactuals with explicit supervision from human trajectories.

Behavior Cloning and Offline RL. Our work is also related to behavior cloning and offline reinforcement learning (Kostrikov et al., 2021; Janner et al., 2021). If one removes both next-state prediction and counterfactual reflection, training reduces to supervised next-action prediction from human trajectories. Offline RL learns from fixed interaction data, but typically optimizes expected return

through reward modeling, value estimation, or planning. ALIGNUSER differs from both paradigms. It does not optimize an explicit reward or value function. Instead, it combines transition modeling with simulator-grounded counterfactual reflection, providing outcome-aware supervision that captures the consequences of alternative actions without relying on value estimation or policy optimization.

3 Problem Formulation

We model the environment as a Markov decision process $\mathcal{M} = (\mathcal{S}, \mathcal{A}, T)$, where states $s \in \mathcal{S}$ are textual representations of pages (e.g., search results, product details, cart), T is the transition function, and $a \in \mathcal{A}$ are actions such as [SEARCH], [CLICK], [ADD_TO_CART], [PURCHASE], [RATE], or [EXIT]. We assume a dataset of n human trajectories:

$$\mathcal{D}_{\text{human}} = \{(s_t^{(n)}, a_t^{(n)}, \hat{s}_{t+1}^{(n)}, p^{(n)})\}_t^n, \quad (1)$$

collected from real user sessions, where $a_t^{(n)}$ denotes the human action at time t , and $\hat{s}_{t+1}^{(n)}$ the subsequent state. Each demonstrator is also associated with a persona p . Our goal is to learn a policy $\pi_\phi(a | s, p)$ parametrized by ϕ , such that the trajectories it induces when interacting with the environment resemble human trajectories at both micro (step-wise action) and macro (session outcome) levels. We further assume a dataset $\mathcal{D}_{\text{rollout}}$ of experience collected either via random interactions or following a curiosity-driven strategy (Bougie and Ichise, 2020), $\{(s_t^{(n)}, a_t^{(n)}, \hat{s}_{t+1}^{(n)}, \dots)\}$.

4 Method

At its core, ALIGNUSER gains an understanding of the world by predicting next states from state-action pairs and aligns with human behaviors by comparing human actions with counterfactual examples. Following this pre-training step, the agent interacts with the recommender system. Figure 1 illustrates the overall architecture.

4.1 World Modeling

We first train our LLM-based policy π_ϕ to approximate the environment transition dynamics. In our study, states are represented entirely in natural language, allowing us to model next-state prediction as a standard next-token prediction objective. Inspired by prior studies on training LLMs as world models, we use next states from the rollout set $\mathcal{D}_{\text{rollout}}$ as direct training signals for the language

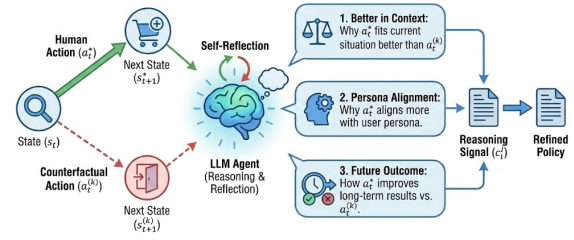


Figure 2: Counterfactual reflection from counterfactual trajectories.

agent’s policy π_ϕ . Note that throughout this paper, we use the term *world model* to denote a learned transition model over observable states.

Given a state-action pair (s_t, a_t) from $\mathcal{D}_{\text{rollout}}$, the model predicts the next state \hat{s}_{t+1} as a sequence of tokens: $\hat{s}_{t+1} \sim \pi_\phi(\cdot | s_t, a_t)$, and train π_ϕ to maximize the likelihood of the human next state s_{t+1}^* :

$$\mathcal{L}_{\text{wm}}(\phi) = - \sum_{(s_t, a_t, \hat{s}_{t+1}) \in \mathcal{D}_{\text{rollout}}} \log p_\phi(\hat{s}_{t+1} | s_t, a_t). \quad (2)$$

For example, when browsing an e-commerce site, the model may learn to predict that clicking on a product leads to a detailed product page, whereas submitting an empty search query results in a “no results” state. These natural-language page descriptions act as next-state supervision, enabling the model to internalize how different user actions transform the shopping session without requiring any handcrafted supervision.

4.2 Human Alignment via Counterfactual Reasoning

To align the policy with human decisions, we compare human trajectories with counterfactuals (Figure 2). For each human transition $(s_t, a_t, \hat{s}_{t+1}, p) \in \mathcal{D}_{\text{human}}$, we sample alternative actions $\{a_t^{(1)}, \dots, a_t^{(K)}\}$ that the current policy π_ϕ considers plausible yet deviate from the demonstrated action. Given the state s_t , we first draw a pool of K candidate actions $\{a_t^{(1)}, \dots, a_t^{(K)}\}$ from $\pi_\phi(\cdot | s_t, p)$ such that the generated action differs from the ground truth:

$$a_t^{(k)} \sim \pi_\phi(\cdot | s_t, p) \quad \text{s.t.} \quad a_t^{(k)} \neq a_t \quad (3)$$

This ensures the exploration of actions that the model currently believes to be plausible, and there-

fore most likely to cause misalignment if left uncorrected.

We then let the agent reason on the counterfactual states, by comparing them with human state-action pairs. Given next states, we prompt the LLM to explain (1) why the human choice is better in the current context, (2) why the human choice is more aligned with its persona and preferences, (3) how the human action improves future outcomes compared to the alternatives. These explanations provide richer, transferable supervision than expert actions alone, leveraging the LLM’s strength in processing language to internalize decision principles that generalize across tasks. In practice, we prompt the model to generate a chain-of-thought c_t^j explaining why the human action a_t is preferable to the alternative a_t^j based on the differences between their resulting states \hat{s}_{t+1} and s_t^j . The prompt is designed to elicit natural language reasoning that highlights potential limitations or inefficiencies in a_t^j , grounded in the actual state transitions observed. This reflection is used for both environment-driven actions (e.g., click, search) and item-centric actions (e.g., like, rate). The lessons are stored in \mathcal{D}_{CR} . We then train the agent to jointly predict the chain-of-thought and the expert action conditioned on the state s_t , using a next-token prediction loss over the target sequence (c_t^j, a_t) :

$$\mathcal{L}_{\text{CR}} = - \sum_{(s_t, a_t, c_t^j, p) \in \mathcal{D}_{\text{CR}}} \log p_\phi(c_t^j, a_t \mid s_t, p), \quad (4)$$

where p_ϕ denotes the language model’s output distribution, aligned with the agent’s policy π_ϕ .

The overall optimization problem that is solved for learning the language agent can be expressed as:

$$\mathcal{L}(\phi) = \lambda_{\text{wm}} \mathcal{L}_{\text{wm}}(\phi) + \lambda_{\text{CR}} \mathcal{L}_{\text{CR}}(\phi), \quad (5)$$

where λ_{wm} and λ_{CR} are scalars that balance world-model and counterfactual terms.

4.3 Interacting with Recommender Systems

Once pre-trained, ALIGNUSER uses the learned policy π_ϕ to act as a synthetic user. Given its persona p , the agent interacts with the recommender system until it either purchases items or decides to terminate the session. Each agent is equipped with an episodic memory that stores its interactions with the RS. The memory is initially populated with the user’s viewing and rating history. When the agent executes a new action or rates an item, the

corresponding interaction is added to the episodic memory.

Namely, at every step, the policy π_ϕ receives a natural-language description of the current state s_t (e.g., a page of recommended items), and we prompt π_ϕ to internally reason about the situation and output an action, as shown in the pseudo-prompt below:

Action Selection Prompt Structure

```
[STATE]
s_t
[PERSONA]
p
[RECENT_HISTORY]
H
[POSSIBLE_ACTIONS]
a_1, a_2, ..., a_M
Instruction: Think step by step about what a careful user with this persona would do next, considering their goals, preferences, and the future consequences of each action.
End with a single line of the form:
BEST-ACTION: <action_token>
RATIONALE: <rationale>
```

The selected action is then executed in the environment (e.g., clicking an item, going to the next page, or exiting), and the process repeats until a terminal action is selected.

To further enhance the ability of the agent to reason on items, we compare our vanilla ALIGNUSER with ALIGNUSER+, which integrates a graph memory (Bougie and Watanabe, 2025c) and causal reasoning. Namely, the agent stores its preferences in a graph-based memory and retrieves evidence to decide whether it likes or dislikes an item. Besides, following the initial action selection a_{tent} , we introduce a *causal reasoning* step where agents generate questions ($Q = \pi_\phi(a_{\text{tent}}, H, p, P_{\text{causal}})$) to validate tentative actions given recent history H and prompt P_{causal} . For each counterfactual scenario (e.g., “What would happen if I exit now?”), the agent estimates outcomes and adjusts its final action based on cause-effect consistency.

5 Experiments

Baselines We compare ALIGNUSER against RecAgent (Wang et al., 2023a), Agent4Rec (Zhang et al., 2023), and SimUSER (Bougie and Watanabe, 2025c) which represent the closest comparable methods. Some experiments involve two versions of our method: ALIGNUSER and ALIGNUSER+, in order to isolate the effects of pretraining and

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
AlignUSER (1:1)	0.8203	0.8372	0.7969	0.8166	0.8432	0.8427	0.8179	0.8301	0.8138	0.8421	0.8263	0.8340
AlignUSER (1:3)	0.7994	0.8423	0.5987	0.6999	0.6843	0.7784	0.6380	0.7014	0.7641	0.8339	0.6118	0.7058
AlignUSER (1:9)	0.7061	0.8438	0.4273	0.5663	0.6648	0.7827	0.3892	0.5198	0.7284	0.7964	0.3379	0.4745
AlignUSER+ (1:1)	0.8317	0.8483	0.8075	0.8274	0.8546	0.8533	0.8292	0.8416	0.8269	0.8549	0.8376	0.8462
AlignUSER+ (1:3)	0.8119	0.8532	0.6113	0.7121	0.6985	0.7915	0.6511	0.7145	0.7781	0.8461	0.6254	0.7190
AlignUSER+ (1:9)	0.7195	0.8524	0.4451	0.5848	0.6787	0.7935	0.4042	0.5356	0.7413	0.8079	0.3528	0.4922

Table 1: User preference alignment across MovieLens, AmazonBook, and Steam datasets. All improvements are statistically significant ($p < 0.05$). Bold: best results for each type (1:1), (1:3) (1:9).

few-shot prompting. When possible, we also report the results of RecMind (Wang et al., 2023c), an agent-based RS.

5.1 Implementation Details

We employ Qwen3-8B as the backbone LLM of our framework. During policy training, we generate $K = 3$ counterfactual actions per state and obtain their predicted next states from the environment. The policy is trained using a weighted combination of world-model and reflection loss, following Sec. 4.2. When persona is not available, we estimate persona attributes via persona-matching, as done in SimUSER (Bougie and Watanabe, 2025c). We investigate four real-world datasets: MovieLens-1M (Harper and Konstan, 2015), Steam (Kang and McAuley, 2018), and AmazonBook (McAuley et al., 2015), OPeRA (Wang et al., 2025).

5.2 Preference Alignment

In order to appropriately respond to recommendations, synthetic users must possess a clear understanding of their own preferences. Thereby, we query the agents to classify items based on whether their human counterparts have interacted with them or not. We randomly assigned 20 items to each of 1,000 agents, with varying ratios (1:m where $m \in \{1, 3, 9\}$) of items users had interacted with to non-interacted items. We treat this as a binary classification task. Table 1 shows ALIGNUSER agents accurately identified items aligned with their tastes, significantly outperforming baselines across all distractor levels (paired t-tests, 95% confidence, $p < 0.05$). These improvements can be directly at-

Methods	MovieLens		AmazonBook		Steam	
	RMSE	MAE	RMSE	MAE	RMSE	MAE
MF	1.2142	0.9971	1.2928	0.9879	1.3148	1.0066
AFM	1.1762	0.8723	1.3006	1.1018	1.2763	0.9724
RecAgent	1.1021	0.7632	1.2587	1.1191	1.0766	0.9598
RecMind-SI (few-shot)	1.0651	0.6731	1.2139	0.9434	0.9291	0.6981
Agent4Rec	0.7612	0.7143	0.8788	0.6712	0.7577	0.6880
SimUSER	0.5020	0.4460	0.5676	0.4210	0.5866	0.5323
ALIGNUSER	0.4693	0.4151	0.5130	0.3992	0.5344	0.5006
ALIGNUSER+	0.4292	0.3871	0.4649	0.3741	0.4970	0.4829

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

tributed to the reflection step, which allows the LLM to understand how personas relate to the agent’s actions and preferences. Further gains stem from the knowledge-graph memory, as observed by comparing ALIGNUSER with ALIGNUSER+.

5.3 Rating Items

A central component of recommender-system interactions is the ability to judge whether a user would like or dislike an item. We evaluate our method on this task by comparing several LLM-based agents with standard baselines, including matrix factorization (MF) (Koren et al., 2009) and Attentional Factorization Machines (AFM) (Xiao et al., 2017). Results are reported in Table 2. Across datasets, our agent consistently achieves lower error than other baselines. Other LLM approaches tend to produce larger deviations, especially on long-tail or sparsely observed items, reflecting their tendency to hallucinate plausible but incorrect ratings. In contrast, ALIGNUSER explicitly aligns the agent’s behavior with its persona during the world-model pretraining phase. This provides auxiliary signals about

	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(persona)	4.41 ± 0.16	3.99 ± 0.18	4.02 ± 0.23	4.05 ± 0.20
ALIGNUSER	<u>4.53 ± 0.15*</u>	<u>4.19 ± 0.17*</u>	<u>4.17 ± 0.21*</u>	<u>4.31 ± 0.19*</u>
ALIGNUSER+	4.58 ± 0.14*	4.25 ± 0.16*	4.22 ± 0.20*	4.34 ± 0.18*

Table 3: Human-likeness score evaluated by GPT-4o across recommendation domains (higher is better). *Significant improvements over best baseline ($p < 0.05$).

preference consistency and item relationships, enabling the agent to form a more coherent internal preference state before issuing a rating.

5.4 Human Likelihood

To assess how closely agent trajectories resemble real user behavior, we adopt GPT-4o as an automatic evaluator, following prior evidence that LLM judges achieve reliability comparable to human annotators (Chiang and Lee, 2023). For each interaction sequence, the evaluator assigns a score on a 5-point Likert scale, where higher values indicate stronger alignment with human-like reasoning and behavioral patterns. As shown in Table 3, our method achieves substantially higher human-likeness scores across all datasets. Our world modeling task reduces inconsistent behaviors and encourages the agent to evaluate how a human would behave under alternative situations. In contrast, baseline LLM agents, such as Agent4Rec, exhibit patterns that the evaluator reliably flags as non-human, including premature [EXIT] actions and erratic ratings for items with similar attributes.

5.5 Reasoning and Persona Consistency

We further measure how agents reproduce human-like reasoning and session dynamics on the OPeRA dataset (Wang et al., 2025), which features state-action pairs, and rationales. First, we report thought–action consistency, where GPT-4o compares LLM-generated and genuine rationales as *coherent*, *partially coherent*, or *contradictory*. The consistency score is the proportion of steps labeled coherent. Second, we measure persona–behavior consistency, whether the actions are consistent with the stated shopping style and preferences. This targets whether the agent maintains a stable, individualized behavior pattern rather than drifting toward a generic shopper. Finally, analyze session-level metrics: number of pages visited, and the purchase rate gap, defined as the absolute difference (%) between human and agent purchase frequencies.

As shown in Table 4, counterfactual reflec-

Method	Thought–Action Consistency (%) ↑	Persona–Behavior Consistency (%) ↑	Pages/Session ≈ human	Purchase Rate Gap (abs., %) ↓
Human (OPeRA)	–	–	5.3	–
Random	38.7	36.1	2.4	22.8
RecAgent	49.5	46.7	3.5	16.3
Agent4Rec	55.8	52.4	4.0	12.1
SimUSER	64.3	61.5	4.6	9.9
ALIGNUSER	<u>86.7</u>	<u>82.4</u>	<u>5.1</u>	<u>2.5</u>
ALIGNUSER+	89.3	85.6	5.1	2.1

Table 4: Thought and persona consistency on OPeRA-test, together with session-level statistics. **Bold**: best result; underlined: second best among synthetic agents.

Method	Thought–Action Consistency (%) ↑	Persona–Behavior Consistency (%) ↑	Pages/Session ≈ human	Purchase Rate Gap (abs., %) ↓
Human (OPeRA)	–	–	5.3	–
Random	38.7	36.1	2.4	22.8
RecAgent	49.5	46.7	3.5	16.3
Agent4Rec	55.8	52.4	4.0	12.1
SimUSER	64.3	61.5	4.6	9.9
AlignUSER	86.7	82.4	5.1	2.5
AlignUSER+	89.3	85.6	5.1	2.1
AlignUSER(image)	88.6	83.5	5.3	2.3
AlignUSER+(image)	90.1	86.5	5.2	1.9

Table 5: Image-conditioned world modeling on OPeRA-test.

tion substantially improves internal coherence. ALIGNUSER raises thought, action consistency compared to baselines. A similar trend appears for persona, behavior consistency, indicating that the policy not only reproduces local decisions but also preserves a stable shopping style over entire sessions. Session statistics also move closer to human behavior. While RecAgent and Agent4Rec tend to under-explore the site — fewer pages than humans, and often over-purchase compared to human shoppers, our method produces more faithful browsing sessions.

5.6 Image-Conditioned World Modeling on OPeRA

Because natural-language page states may miss layout and saliency cues, we additionally study an image-conditioned variant on OPeRA using UI screenshots and Qwen3-VL-8B, while keeping the rest of the training protocol unchanged. Results in Table 5 show that adding images further improves thought–action consistency, persona–behavior consistency, and purchase-rate matching. This suggests that visual signals are helpful in diverse web-shopping environments, where small layout differences can change the semantics of a page.

5.7 Action Alignment

Next, we measure action alignment. We adopt an exact-match criterion: a prediction is counted as correct only if all action parameters match the ground-truth. For click actions, this requires matching the clicked target (e.g., the correct prod-

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
SimUSER-SFT 🚩	39.80	61.50	57.20	70.20
AlignUSER 🚩	51.47	69.81	66.29	78.07
AlignUSER+ 🚩	52.92	71.94	66.88	80.52

Table 6: 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 (%).

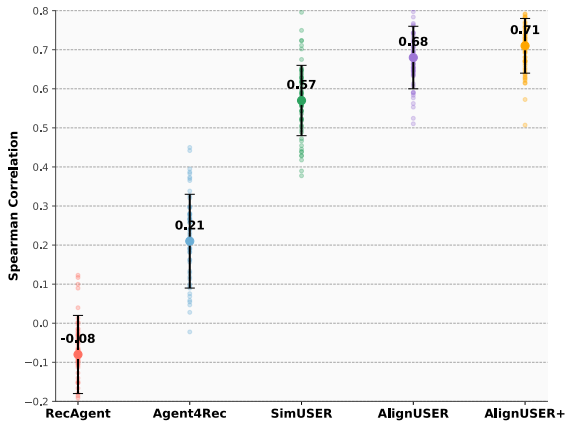


Figure 3: Spearman correlation between estimated and actual engagement metrics. Higher values indicate better alignment with ground-truth metrics.

uct 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 assess 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). Finally, because online shopping is inherently goal-driven, we evaluate the prediction of session outcomes. We measure performance on these terminal actions using accuracy and weighted F1, which reflects how well the model captures users’ eventual decisions

and long-term goals over the course of a session. As shown in Table 6, ALIGNUSER surpasses prior LLM-based simulators, and ALIGNUSER+ yields the strongest results, with particularly large gains in action generation accuracy and session-outcome prediction.

5.8 Offline A/B Testing

We further examine whether ALIGNUSER 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 recommended items. Every test compares multiple recommendation strategies, with the average number of visited pages used as the primary business metric. 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 shown in Figure 3, AlignUSER+ achieves the highest correlation with ground truth, outperforming all other baselines. Statistical tests confirm that the improvements over all baselines are significant ($p < 0.05$), with AlignUSER also clearly outperforming the prior state-of-the-art user simulator, SimUSER.

5.9 Recommender System Evaluation

Understanding the efficacy of various recommendation algorithms is crucial for enhancing user satisfaction. By simulating human proxies, we

	\bar{P}_{view}	\bar{N}_{like}	\bar{P}_{like}	\bar{N}_{exit}	\bar{S}_{sat}
Random	0.295	3.05	0.247	2.80	2.60
Pop	0.388	4.15	0.365	2.95	3.28
MF	0.468	5.72	0.439	3.08	3.70
MultVAE	<u>0.521</u>	5.31	0.452	<u>3.22</u>	<u>3.82</u>
LightGCN	0.552	<u>5.49</u>	<u>0.446</u>	3.26	3.88

Table 7: Evaluation of recommendation strategies on a recommendation task from the MovieLens dataset.

can better predict how users will engage with recommender systems, providing valuable interactive metrics. We compare various recommendation strategies, including most popular (Pop), matrix factorization (MF) (Koren et al., 2009), LightGCN (He et al., 2020), and MultVAE (Liang et al., 2018), using the MovieLens dataset. Upon exiting, agents rated their satisfaction on a scale from 1 to 10. Ratings above 3 were considered indicative of a *like*. Metrics include average viewing ratio (\bar{P}_{view}), average number of likes (\bar{N}_{like}), average ratio of likes (\bar{P}_{like}), average exit page number (\bar{N}_{exit}), and average user satisfaction score (\bar{S}_{sat}). Here, rather than evaluating the proposed framework itself, we use simulated users to examine whether their interactions with recommender systems are coherent with well-established trends in the literature. Table 7 demonstrates that agents exhibit higher satisfaction with advanced recommendations versus random and Pop methods, consistent with observations from real-world systems.

6 Discussion

Our study demonstrates that incorporating an explicit world-modeling task and counterfactual self-reflection yields user agents that more faithfully reproduce human interaction patterns and produce more reliable evaluation signals for recommender systems. Despite these gains, several limitations remain.

While this abstraction enables a unified modeling framework across domains, it omits fine-grained visual, layout, and interaction signals inherent to real e-commerce interfaces. This limitation is particularly evident in OPeRA, where page structures exhibit greater diversity. In our image-conditioned experiments, we observe improvements in both next-state prediction and downstream behavioral alignment, indicating that richer state representations are beneficial. Extending the framework to incorporate multimodal inputs, such as screenshots,

product imagery, or structured DOM representations offers a promising direction to enhance the fidelity of both the world model and the resulting policy.

Training relies on human trajectories, which provide only partial coverage of the action space. While world-model-guided counterfactual rollouts mitigate this limitation by exposing the agent to alternative transitions, the policy may still extrapolate poorly in states that are rarely visited by humans or in sequences that diverge significantly from typical browsing paths.

Our experiments focus on interaction settings with moderate temporal depth (e.g., shopping, books), where sessions typically span a few dozen steps. Deploying ser agents in long-horizon settings such as news reading, continuous mobile app use, or social media feeds may require additional components, such as persistent memory, hierarchical planning, or explicit long-term goal modeling.

Finally, the behavior of the agent inevitably inherits biases and idiosyncrasies of the underlying LLM. Although the reflection mechanism constrains deviations from human demonstrations, residual biases in preference modeling, sentiment, or perception may still surface and influence evaluation outcomes.

7 Conclusion

We introduced ALIGNUSER, a world-model-guided framework for learning user agents from human trajectories. By modeling environment dynamics through next-state prediction and generating counterfactuals to align actions with human decisions and personas, ALIGNUSER brings LLM-based agents closer to real user behavior. Experiments in shopping and recommender system evaluation scenarios demonstrate improvements in action prediction, rating, and preference alignment. Our method also exhibits a positive correlation between simulated and online A/B test outcomes. We believe that synthetic users offer a promising foundation for scalable and privacy-preserving evaluation of recommender systems, and pave the way to more realistic, controllable, and interpretable agent-based simulation frameworks.

8 Limitations

Although ALIGNUSER achieves the highest alignment with human trajectories among the evaluated baselines, several limitations must be acknowl-

edged. First, reproducibility is constrained by the availability of human interaction logs. One dataset used for evaluating alignment is proprietary, limiting full transparency and replication.

Second, our approach inherits the cultural, demographic, and socioeconomic biases present in large language models. Since ALIGNUSER relies on LLM-generated reflections and counterfactual reasoning, any underlying biases in the base model may manifest as skewed interpretations of user motives or preferences. Related to this, we occasionally observe hallucinations in world-model rollouts, for example, predicting implausible next states or misinterpreting page semantics, which can propagate into the reflective policy and yield suboptimal decisions.

Third, the effectiveness of ALIGNUSER is tightly coupled to the strengths and weaknesses of the underlying LLMs. Inconsistencies in reasoning quality, brittleness under distribution shift, and occasional unfounded judgments may degrade the fidelity of simulated users, particularly in sparsely covered regions of the state-action space.

Fourth, when explicit persona annotations are unavailable, ALIGNUSER relies on persona matching. Although this component is reasonably robust in our experiments, mismatches can propagate to downstream simulation behavior, especially for subgroup-level analyses or long-horizon consistency of individual agents. At population scale, such noise may partly average out, but systematic errors in persona matching remain an important limitation.

Finally, the framework integrates several interacting components, world modeling, counterfactual generation, and reflection, which can make it difficult to isolate the contribution of each module. While we provide ablation studies to partially disentangle these effects, future work is needed to better understand how different training signals and architectural choices influence alignment outcomes.

9 Ethics Statement

This work introduces a framework for training synthetic users to support the evaluation of recommender systems. While such agents provide clear advantages in terms of scalability, cost-effectiveness, and privacy preservation, the approach raises several ethical considerations.

Synthetic user agents trained on human logs may

inadvertently reproduce or amplify demographic, cultural, or socioeconomic biases present in the underlying LLMs or in the behavioral data used for world-model learning. These biases can manifest in the agent’s simulated preferences or interaction patterns, potentially leading to misleading evaluation signals. In particular, biased reflections or hallucinated counterfactuals may privilege certain user groups or product categories, reinforcing unfairness in downstream recommender systems.

A broader concern lies around the use of realistic synthetic users as proxies for actual individuals. When such agents are used to assess or optimize RS behavior, there is a risk that system designers may over-rely on simulated outcomes, reducing the involvement of real users, domain experts, or impacted stakeholders. This is especially sensitive in domains such as e-commerce, job recommendations, or content consumption, where algorithmic decisions can influence user autonomy and exposure to information.

Finally, the generation of counterfactual trajectories, while valuable for alignment, relies on models that may produce plausible, sounding but factually incorrect predictions. Such inaccuracies could misguide evaluation or optimization if not interpreted with caution.

We emphasize that synthetic users should complement, rather than replace, real human feedback. Responsible deployment requires transparency regarding model limitations, continuous monitoring for bias, and safeguards to prevent misuse or misinterpretation of simulation outcomes.

References

- James Bisbee, Joshua D. Clinton, Cassy Dorff, Brenton Kenkel, and Jennifer M. Larson. 2024. [Synthetic replacements for human survey data? the perils of large language models](#). *Political Analysis*, 32(4):401–416.
- Nicolas Bougie and Ryutaro Ichise. 2020. Skill-based curiosity for intrinsically motivated reinforcement learning. *Machine Learning*, 109:493–512.
- 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. Generative reviewer agents: Scalable simulacra of peer review. In *Proceedings of the 2025 Conference on*

- Empirical Methods in Natural Language Processing: Industry Track*, pages 98–116.
- Nicolas Bougie and Narimawa Watanabe. 2025c. 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? *arXiv preprint arXiv:2305.01937*.
- David Ha and Jürgen Schmidhuber. 2018. Recurrent world models facilitate policy evolution. *Advances in neural information processing systems*, 31.
- Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. 2019. Dream to control: Learning behaviors by latent imagination. *arXiv preprint arXiv:1912.01603*.
- F Maxwell Harper and Joseph A Konstan. 2015. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)*, 5(4):1–19.
- Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 639–648.
- 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. 2023. Recommender ai agent: Integrating large language models for interactive recommendations. *arXiv preprint arXiv:2308.16505*.
- Dietmar Jannach and Michael Jugovac. 2019. Measuring the business value of recommender systems. *ACM Transactions on Management Information Systems (TMIS)*, 10(4):1–23.
- Michael Janner, Qiyang Li, and Sergey Levine. 2021. Offline reinforcement learning as one big sequence modeling problem. *Advances in neural information processing systems*, 34:1273–1286.
- Carolin Kaiser, Jakob Kaiser, Vladimir Manewitsch, Lea Rau, and Rene Schallner. 2025. Simulating human opinions with large language models: Opportunities and challenges for personalized survey data modeling. In *Adjunct Proceedings of the 33rd ACM Conference on User Modeling, Adaptation and Personalization*, page 82–86. Association for Computing Machinery.
- Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *2018 IEEE international conference on data mining (ICDM)*, pages 197–206. IEEE.
- Minsoo Kim, YeonJoon Jung, Dohyeon Lee, and Seungwon Hwang. 2022. Plm-based world models for text-based games. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1324–1341. Association for Computational Linguistics.
- Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37.
- Ilya Kostrikov, Ashvin Nair, and Sergey Levine. 2021. Offline reinforcement learning with implicit q-learning. *arXiv preprint arXiv:2110.06169*.
- Guohao Li, Hasan Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. 2023. Camel: Communicative agents for "mind" exploration of large language model society. *Advances in Neural Information Processing Systems*, 36:51991–52008.
- 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.
- Dawen Liang, Rahul G Krishnan, Matthew D Hoffman, and Tony Jebara. 2018. Variational autoencoders for collaborative filtering. In *Proceedings of the 2018 world wide web conference*, pages 689–698.
- Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. 2015. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*, pages 43–52.
- Joon Sung Park, Joseph C O’Brien, Carrie J Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. 2023. Generative agents: Interactive simulacra of human behavior. *arXiv preprint arXiv:2304.03442*.
- Aadesh Salecha, Molly E Ireland, Shashanka Subrahmanya, João Sedoc, Lyle H Ungar, and Johannes C Eichstaedt. 2024. Large language models show human-like social desirability biases in survey responses. *arXiv preprint arXiv:2405.06058*.
- Lei Wang, Jingsen Zhang, Xu Chen, Yankai Lin, Ruihua Song, Wayne Xin Zhao, and Ji-Rong Wen. 2023a. Recagent: A novel simulation paradigm for recommender systems. *arXiv preprint arXiv:2306.02552*.
- Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023b. Self-consistency improves chain of thought reasoning in language models. In *Proceedings of the International Conference on Learning Representations*.

- Yancheng Wang, Ziyang Jiang, Zheng Chen, Fan Yang, Yingxue Zhou, Eunah Cho, Xing Fan, Xiaojiang Huang, Yanbin Lu, and Yingzhen Yang. 2023c. Recmind: Large language model powered agent for recommendation. *arXiv preprint arXiv:2308.14296*.
- Ziyi Wang, Yuxuan Lu, Wenbo Li, Amirali Amini, Bo Sun, Yakov Bart, Weimin Lyu, Jiri Gesi, Tian Wang, Jing Huang, and 1 others. 2025. Opera: A dataset of observation, persona, rationale, and action for evaluating llms on human online shopping behavior simulation. *arXiv preprint arXiv:2506.05606*.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.
- 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*.
- Se-eun Yoon, Zhankui He, Jessica Maria Echterhoff, and Julian McAuley. 2024. Evaluating large language models as generative user simulators for conversational recommendation. *arXiv preprint arXiv:2403.09738*.
- Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. 2022. Star: Bootstrapping reasoning with reasoning. *Advances in Neural Information Processing Systems*, 35:15476–15488.
- An Zhang, Leheng Sheng, Yuxin Chen, Hao Li, Yang Deng, Xiang Wang, and Tat-Seng Chua. 2023. On generative agents in recommendation. *arXiv preprint arXiv:2310.10108*.
- Kai Zhang, Xiangchao Chen, Bo Liu, Tianci Xue, Zeyi Liao, Zhihan Liu, Xiyao Wang, Yuting Ning, Zhaorun Chen, Xiaohan Fu, and 1 others. 2025. Agent learning via early experience. *arXiv preprint arXiv:2510.08558*.
- 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.

A Experimental Setup

Experimental Settings. We separate the dataset into training, validation, and test sets (80/10/10%), using a time-based split. This ensures to reflect the temporal distribution shift that may be observed in the real world. To address privacy concerns, the name and gender are omitted. Moreover, for the sake of generality, we do not utilize user-specific information available in these datasets, relying instead on the personas identified via persona-matching (Bougie and Watanabe, 2025c). To obtain rollouts $\mathcal{D}_{\text{rollout}}$ that cover a large space of the environment, we collect rollouts using a decaying ϵ -greedy exploration policy. At episode t , the behavior policy selects a random action with probability ϵ_t and otherwise follows the greedy action of the current policy. We linearly anneal ϵ_t from 0.3 to 0.05 over 100,000 episodes. $\mathcal{D}_{\text{rollout}}$ was augmented with human transitions and their counterfactual transitions. This ensures that the world model covers regions not visited by humans. Matrix factorization (MF) is utilized as the recommender model unless specified otherwise. In our simulator, agents are presented with four items $n = 4$ per page and allowed to interact by viewing and rating items based on their preferences. When the output of the LLM deviated from the desired format, resulting in errors, the LLM was re-prompted with the following instruction: You have one more chance to provide the correct answer.

Unless otherwise stated, we report results as mean \pm standard deviation over repeated runs with different random seeds. For pairwise comparisons, we use paired t -tests across runs and report significance at $p < 0.05$. For correlation-based analyses, we additionally report bootstrap confidence intervals in the appendix. We also summarize runtime and parameter sensitivity in the appendix, including the effect of the number of counterfactuals K and the loss weights ($\lambda_{\text{wm}}, \lambda_{\text{CR}}$).

Counterfactuals. We generate $K = 3$ counterfactual actions (excluding a) per state and obtain their next states from the environment. During training, we use a batch size of 16 and a learning rate of $1e^{-5}$, and train for 8 epochs. We set the loss weights to $\lambda_{\text{wm}} = 1.0$ and $\lambda_{\text{CR}} = 0.5$. In datasets such as MovieLens that do not include actions (e.g., [CLICK], we only let the agent reflect at the item level by sampling alternative rating actions([1], [2],...), treating different rating values

as distinct choices in the action space. For datasets that provide interaction actions, we also generate counterfactuals at the trajectory level, enabling reflection over alternative sequences of actions and states rather than solely on isolated item-level decisions. For instance, when the supervised signal is a rating decision (e.g., MovieLens/AmazonBook), we additionally treat each discrete rating value in $\{1, 2, 3, 4, 5\}$ as a distinct action and sample alternative rating values as counterfactuals. Actions are executed in the simulation to collect \hat{s}_{t+1} , which is then used for counterfactual reflection.

Preferences. The preferences of each agent are stored in a memory, being initialized from the history of its human counterpart. When a review score for an item is greater than 4, the agent stores a memory entry in the form I liked {item_name} based on my review score of {score}. For a score of 2 or below, the following format is utilized I disliked {item_name} based on my review score of {score}. Neutral scores result in the entry I felt neutral about {item_name} based on my review score of {score}. In all the experiments, items rated ≥ 4 are considered as liked by the user, while items ≤ 2 are considered as disliked. These interactions are stored both as plain text in the episodic memory and as relationships in the knowledge graph memory. The knowledge-graph memory utilizes the same retrieval implementation and parameters as done in SimUSER (Bougie and Watanabe, 2025c). The top- k_2 items, their attributes, and paths are returned to condition decision making (see state/page prompt). Namely, the titles and ratings of retrieved items are concatenated to the prompt.

Persona. To lay a reliable foundation for the generative agent’s subsequent interactions and evaluations, each agent has its own persona p . A persona p encompasses a set of features that characterize the user: **age**, **personality**, and **occupation**. Personality traits are defined by the Big Five personality traits: *Openness*, *Conscientiousness*, *Extraversion*, *Agreeableness*, and *Neuroticism*, each measured on a scale from 1 to 3. Along with attributes extracted from its historical data: $p \cup \{\text{pickiness, habits}\}$. *pickiness* level is sampled in $\{\text{not picky, moderately picky, extremely picky}\}$ based on the user’s average rating. Habits account for user tendencies in engagement, conformity, and variety (Zhang et al., 2023). Namely, given the average rating \bar{R} of a user: $\bar{R} = \frac{1}{N} \sum_{i=1}^N r_{ui}$, the pickiness level $P(\bar{R})$

of a user was determined based on the following thresholds:

$$P(\bar{R}) = \begin{cases} P_1 & \text{if } \bar{R} \geq 4.5 \\ P_2 & \text{if } 3.5 \leq \bar{R} < 4.5 \\ P_3 & \text{if } \bar{R} < 3.5 \end{cases}$$

where P_1 corresponds to *not picky*, P_2 corresponds to *moderately picky*, and P_3 corresponds to *extremely picky*. Engagement measures the frequency and breadth of a user’s interactions with recommended items, distinguishing highly active users from those interacting with only a few items. Engagement can be mathematically expressed as: $T_{act}^u = \sum_{i \in \mathcal{I}} y_{ui}$, where given a user $u \in \mathcal{U}$ and an item $i \in \mathcal{I}$, the quality of the item is denoted by $R_i = \frac{1}{\sum_{u \in \mathcal{U}} y_{ui}} \sum_{u \in \mathcal{U}} y_{ui} \cdot r_{ui}$. Here, $y_{ui} = 0$ indicates that the user u has not rated the item i and inversely $y_{ui} = 1$ indicates that the user has rated the item with $r_{ui} \in \{1, 2, 3, 4, 5\}$. Conformity captures how closely a user’s ratings align with average item ratings, drawing a distinction between users with unique tastes and those whose opinions closely mirror popular sentiments. For user u , the conformity trait is defined as: $T_{conf}^u = \frac{1}{\sum_{i \in \mathcal{I}} y_{ui}} \sum_{i \in \mathcal{I}} y_{ui} \cdot |r_{ui} - R_i|^2$. Variety reflects the user’s proclivity toward a diverse range of item genres or their inclination toward specific genres. The variety trait for user u is formulated as: $T_{div}^u = |U_{i \in \{y_{ui}=1\}} g_i|$.

Interactions with Recommender Systems. Once pre-trained, ALIGNUSER leverages the learned policy π_ϕ to act as a synthetic user. Given a persona p , the agent interacts with the recommender simulator in a page-by-page manner until it selects a terminal action. Each step consists of an internal [WATCH]/[SKIP] screening over items on the current page to identify candidates consistent with the persona and memory, and selecting one environment action (e.g., navigate, click for details, or exit). The [WATCH]/[SKIP] screening is not an environment action; it is an internal decision to reduce the mental workload on users. During the subsequent action selection, we prompt π_ϕ to internally reason about the situation and output an action. The selected action is executed in the environment (e.g., clicking an item to reveal a more detailed description, moving to the next page, or exiting), and the loop repeats until termination. In recommendation domains (MovieLens, Steam, AmazonBook), sessions terminate via [EXIT]; while in the web-shopping domain (OPeRA), it may in-

clude purchase-related decisions before [EXIT]. ALIGNUSER+ integrates a graph memory and causal validation as in SimUSER (Bougie and Watanabe, 2025c). The agent stores preference evidence in a graph-based memory and retrieves supporting paths to decide whether it likes or dislikes an item. Before executing a selected action, the agent performs a causal reasoning step in which it generates questions ($Q = \pi_\phi(a_{tent}, H, p, P_{causal})$) to validate the tentative action. For each counterfactual scenario (e.g., “Does this rating align with my preferences?”), the agent estimates outcomes and revises its final action based on cause-effect consistency.

B Datasets

MovieLens-1M. The MovieLens-1M dataset is a widely used dataset for recommender-system research. It contains approximately 1 million movie ratings on a 1–5 star scale, provided by 6,040 users over 3,706 movies. In addition to user-item rating interactions, the dataset includes movie metadata such as titles and genre labels, as well as basic user demographic attributes, including age, gender, and occupation.

Steam. The Steam dataset consists of user-game interaction data collected from the Steam platform. The dataset includes user identifiers, game identifiers, and associated English-language user reviews. Game-level metadata such as titles is also provided.

AmazonBook. The AmazonBook dataset corresponds to a subset of the Amazon product reviews corpus restricted to the Books category. It contains user-item interactions in the form of ratings and textual reviews, along with book-level metadata such as titles and category information.

OPeRA. OPeRA is a dataset designed to study and evaluate large language models for simulating human online shopping behavior. It contains real-world shopping session logs that combine user persona information collected via surveys, observations of webpage content, fine-grained user actions (e.g., clicks and navigation events), and self-reported rationales explaining users’ decisions.

C Simulation Environment

Our simulator mirrors real-world recommendation platforms like Netflix, or 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, and (ii) a *web-shopping* pages for OPeRA. The recommendation algorithm is structured as a standalone module, allowing including any algorithm. This design features preimplemented collaborative filtering-based strategies, including random, most popular, Matrix Factorization, LightGCN, and MultVAE.

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 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}
<- {item_title} -> <- History ratings: {item_rating} -> <- Summary: {item_description} -> <- Similar items: {similar_items} ->
<- {item_title} -> <- History ratings: {item_rating} -> <- Summary: {item_description} -> <- Similar items: {similar_items} ->
...

```

Here, $\{item_rating\}$ is the agent’s own historical rating when available, otherwise a dataset-derived statistic (i.e., global mean rating). $\{similar_items\}$ lists retrieved neighbors from the agent’s memory graph in the form ***title (rating/5)***, and is only displayed for SimUSER and ALIGNUSER+.

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>]: reveal 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:

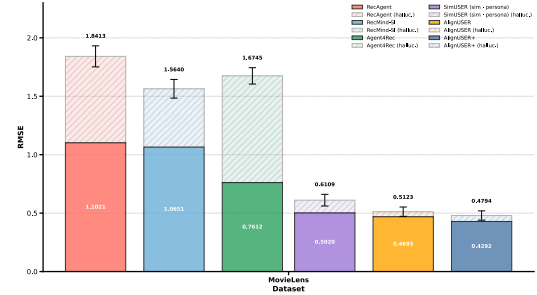


Figure 4: Comparison of RMSE values for original (dark colors) and hallucination-affected (light colors) models for the rating task on MovieLens.

Page Format (OPeRA)

```

PAGE {page_number}
CONTEXT: {page_context}
PRODUCTS:
<- {product_title} -> <- Price: {price}
-> <- Availability: {availability} -> <-
Details: {short_description} ->
<- {product_title} -> <- Price: {price}
-> <- Availability: {availability} -> <-
Details: {short_description} ->
...
INTERACTIVE ELEMENTS (semantic
IDs):
{semantic_id_1}, {semantic_id_2}, ..., {se-
mantic_id_L}

```

Actions follow the same action space as described in OPeRA dataset (Wang et al., 2025), extended with the navigation actions described above.

D Additional Experiments

D.1 Rating Items under Hallucination

In this experiment, we specifically target items that are unfamiliar to the LLM, seeking to evaluate the ability of our trained agents to mitigate hallucination through their memory and alignment modules. Similarly to Section 5.3, users are asked to rate movies (MovieLens), but we exclusively include items that are detected as unknown to the LLM. These items i are identified by querying the LLM to classify each movie into one of 18 genres. If the LLM’s genre classification matches the actual category g_i , it indicates that the LLM is familiar with the item, and such movies are excluded from the experiment. From Figure 4, it is evident that while the RMSE values for all methods increase under hallucination, ALIGNUSER and ALIGNUSER+

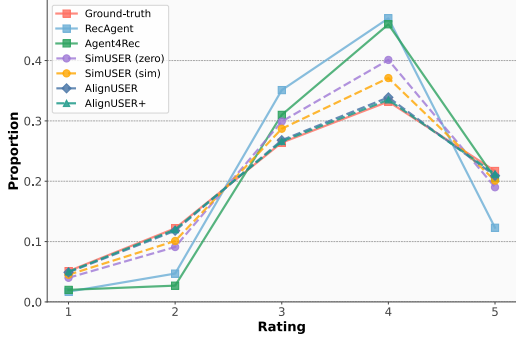


Figure 5: Comparison of rating distributions between ground-truth and human proxies.

Model Variant	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(persona)	4.41 ± 0.16	3.99 ± 0.18	4.02 ± 0.23	4.05 ± 0.20
ALIGNUSER−WM	4.27 ± 0.17	4.03 ± 0.16	4.06 ± 0.22	4.11 ± 0.18
ALIGNUSER−CR	4.12 ± 0.18	3.91 ± 0.17	3.87 ± 0.21	3.98 ± 0.19
ALIGNUSER−Persona	4.21 ± 0.15	3.95 ± 0.16	3.92 ± 0.20	4.02 ± 0.18
ALIGNUSER	4.53 ± 0.15*	4.19 ± 0.17*	4.17 ± 0.21*	4.31 ± 0.19*
ALIGNUSER+	4.58 ± 0.14*	4.25 ± 0.16*	4.22 ± 0.20*	4.31 ± 0.18*

Table 8: Ablation study on human-likeness task. *Significant improvements over best baseline ($p < 0.05$).

are the most robust overall. This relative robustness can be attributed to the combination of reflection and KG memory: by leveraging relationships between users, movies, and ratings from previous interactions, the agents can compare an unfamiliar movie with similar, well-known ones and anchor their predictions in familiar contexts.

D.2 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 aligns closely with the distributions observed in the original dataset. Figure 5 presents the rating distribution from the MovieLens-1M 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 RecAgent and Agent4Rec assign fewer low ratings than real users, SimUSER reduces this mismatch, and ALIGNUSER and ALIGNUSER+ come closest to the true distribution.

Backbone	MovieLens	AmazonBook	Steam	OPeRA
ALIGNUSER+ (Llama-3.2-3B)	4.32 ± 0.18	4.05 ± 0.19	4.07 ± 0.21	4.01 ± 0.20
ALIGNUSER+ (Qwen-2.5-7B)	4.44 ± 0.16	4.13 ± 0.18	4.14 ± 0.20	4.08 ± 0.19
ALIGNUSER+ (Llama-3.1-8B)	4.52 ± 0.15	4.21 ± 0.17	4.20 ± 0.20	4.15 ± 0.18
ALIGNUSER+ (Qwen3-8B)	4.58 ± 0.14	4.25 ± 0.16	4.22 ± 0.20	4.31 ± 0.18

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

D.3 Ablation Studies

To understand the contribution of each component in our method, we perform ablations on the human-likeness evaluation across four datasets. Specifically, we remove: (1) the world-model objective (−WM), which prevents the agent from learning environment dynamics; (2) counterfactual reasoning (−CR), disabling contrastive alignment with human decisions; and (3) persona grounding (−Persona), which removes persona information from the policy input. Results are reported in Table 8.

Removing the world-model objective leads to a consistent drop in human-likeness, indicating that understanding environment dynamics is crucial for generating coherent interaction patterns. Eliminating counterfactual reasoning produces the sharpest decline, confirming that reflection on human−counterfactual gaps is essential for behavior alignment. Finally, ablating persona grounding reduces variability and expressiveness in simulated behavior, particularly on MovieLens and Steam, where personal preferences strongly influence item choices. The full models, ALIGNUSER and ALIGNUSER+, outperform all ablations, highlighting the importance of jointly training on world-model dynamics, persona grounding, and counterfactual reflection.

D.4 LLM Backbone Choice

We further study the impact of the backbone LLM by swapping the base model while keeping the rest of ALIGNUSER+ unchanged. As shown in Table 9, all backbones achieve high human-likeness scores, indicating that the proposed world-model learning and counterfactual alignment 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 remain modest, suggesting that most of the alignment gains stem from the ALIGNUSER architecture rather than from the scale of the backbone model.

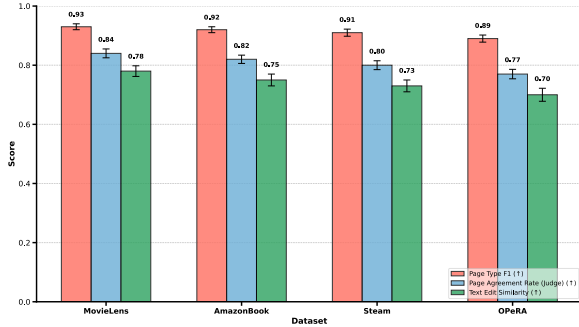


Figure 6: Next-state prediction performance of ALIGNUSER. Error bars indicate variation across 10 runs.

D.5 Running Time Analysis

We compare the running time of AlignUSER, SimUSER, and Agent4Rec for 1,000 user interactions. While Agent4Rec and SimUSER perform API calls to GPT-4o, AlignUSER primarily performs inference with a locally served Qwen2.5 policy (vLLM). Without parallelization, Agent4Rec and SimUSER require 9.3h and 10.1h, respectively, whereas AlignUSER requires ≈ 0.6 h using 4 GPUs. In addition, using current GPT-4o pricing, 1,000 interactions costs around \$16–\$21, whereas AlignUSER costs about \$6–\$8 in GPU time.

D.6 Next-State Prediction

This ablation evaluates whether the next-state prediction task captures basic environment dynamics. For each held-out transition (s_t, a_t, s_{t+1}) , we compare the predicted next state with the real one. Since next states differ in both page structure and textual content, we use three complementary metrics. (i) Page Type F1 (↑): F1 score over coarse page categories (e.g., browse, cart), (ii) page agreement rate (↑): a judge model assesses whether the predicted and ground-truth states describe the same page; and (iii) text edit similarity. Similarity between canonicalized text representations of the two page states, based on normalized edit distance. As shown in Figure 6, AlignUSER predicts the next page state reliably. The Judge Agreement Rate is slightly lower, which is expected because two states can be “coherent” even when some optional details differ (e.g., small differences in shown items). Besides, the small drop in OPeRA can be explained by the increasing state diversity and noisier content inherent in web-shopping websites.

D.7 Sensitivity to Number of Counterfactuals

We now study the sensitivity of ALIGNUSER to the number of counterfactual actions sampled per human step during reflection. All settings are identical to Sec. 4.2 except that we vary $K \in \{0, 1, 2, 3, 5\}$, where $K=0$ disables counterfactual reflection (world-model pretraining only). We report next-action prediction metrics on OPeRA-test (Table 10) following the same protocol as Table 6. Results indicate that increasing K consistently improves next-action prediction, with the largest gains occurring from $K=0$ to $K=2$. Performance saturates beyond $K=3$, suggesting diminishing returns from additional counterfactual supervision, likely because it explores implausible actions.

D.8 Persona Matching Accuracy

As personas are a central component to simulate diverse and heterogeneous users, we evaluate the effectiveness of the self-consistent persona-matching technique, being utilized in this study. Utilizing the MovieLens1 dataset, we predict the agent and occupation of users based on their interaction history. Experimental results are summarized in Table 11. Overall, persona matching turns out to be reasonably robust for enriching simulated agents with detailed backgrounds, including domains where explicit demographic data is not readily provided.

D.9 Sensitivity to Loss Weights

We analyze the impact of balancing world modeling and counterfactual reflection by varying the loss weights in Eq 5. When $\lambda_{CR}=0$, counterfactual reflection is disabled and the training objective reduces to world-model pretraining only. We report next-action prediction metrics on OPeRA-test using the same protocol as Table 6. As shown in Table 12, introducing counterfactual reflection ($\lambda_{CR}>0$) yields substantial gains over world-model-only training. Performance is relatively stable around the default setting (1.0, 0.5).

D.10 Counterfactual Reflection in World Model

Our default training jointly optimizes world modeling and counterfactual reflection, where next states are obtained by rolling out actions in the simulator. Here, we assess whether a two-stage alternative that replaces simulator rollouts during reflection with *imagined* next states predicted by the learned world model. Concretely, we (i) first train the policy with the world-model objective to predict next

Model	Action Gen. (Accuracy)	Action Type (Macro F1)	Click Type (Weighted F1)	Session Outcome (Weighted F1)
AlignUSER ($K=0$)	37.26	58.41	54.66	69.92
AlignUSER ($K=1$)	44.53	63.72	60.11	74.35
AlignUSER ($K=2$)	48.98	67.10	63.89	76.84
AlignUSER ($K=3$)	51.47	69.81	66.29	78.07
AlignUSER ($K=5$)	<u>52.08</u>	<u>70.12</u>	66.10	<u>78.66</u>
AlignUSER+ ($K=0$)	39.41	60.07	56.20	71.58
AlignUSER+ ($K=1$)	46.00	65.31	61.35	75.32
AlignUSER+ ($K=2$)	50.41	69.02	64.92	78.11
AlignUSER+ ($K=3$)	52.92	71.94	66.88	80.52
AlignUSER+ ($K=5$)	53.31	72.10	67.05	80.87

Table 10: Sensitivity to the number of counterfactual actions K used for reflection training on OPeRA. All metrics are percentages (%). $K=0$ disables reflection.

Metric	Age	Occupation
Accuracy	0.7184	0.6691
Precision	0.7512	0.6875
Recall	0.7863	0.7386
F1 Score	0.7683	0.7120

Table 11: Performance of persona matching in predicting age and occupation utilizing MovieLens-1M.

states, then (ii) freeze the world model and perform counterfactual reflection by sampling $K=3$ alternative actions per step and using the policy to obtain their consequences $\tilde{s}_{t+1} = \pi_\phi(s_t, a)$ instead of the environment state \hat{s}_{t+1} . As shown Table 13, using imagined next states yields lower scores than environment-based reflection, which is consistent with the fact that world-model predictions are accurate but not perfect. Nevertheless, imagined reflection remains substantially stronger than world-model-only training ($K=0$), indicating that counterfactual supervision is still beneficial even when consequences are provided by the learned dynamics model.

D.11 Image-Conditioned Next-State Prediction

We further evaluate whether screenshots improve next-state prediction on OPeRA. Using the same protocol as Sec. 6, we train an image-conditioned variant with Qwen3-VL-8B and compare it against the text-only setting. Results in Table 14 show that image inputs mainly improve judge agreement, indicating that visual layout and saliency cues are

useful for disambiguating diverse shopping pages.

E Prompts

E.1 Post-Interview Prompt

The prompt presented to each agent for post-interview is as follows:

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

In Section 5.2, the rating prompt is modified with the following instructions:

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”).

Model	Action Gen. (Accuracy)	Action Type (Macro F1)	Click Type (Weighted F1)	Session Outcome (Weighted F1)
AlignUSER ($\lambda_{wm}=1.0, \lambda_{CR}=0$)	37.26	58.41	54.66	69.92
AlignUSER ($\lambda_{wm}=1.0, \lambda_{CR}=0.25$)	49.92	67.92	64.98	77.41
AlignUSER ($\lambda_{wm}=1.0, \lambda_{CR}=0.5$)	51.47	69.81	66.29	78.07
AlignUSER ($\lambda_{wm}=1.0, \lambda_{CR}=1.0$)	<u>51.88</u>	<u>70.03</u>	66.14	<u>78.29</u>
AlignUSER ($\lambda_{wm}=2.0, \lambda_{CR}=0.5$)	51.21	69.55	<u>66.33</u>	78.01
AlignUSER+ ($\lambda_{wm}=1.0, \lambda_{CR}=0$)	39.41	60.07	56.20	71.58
AlignUSER+ ($\lambda_{wm}=1.0, \lambda_{CR}=0.25$)	51.02	69.64	65.37	79.04
AlignUSER+ ($\lambda_{wm}=1.0, \lambda_{CR}=0.5$)	52.92	71.94	66.88	80.52
AlignUSER+ ($\lambda_{wm}=1.0, \lambda_{CR}=1.0$)	<u>53.12</u>	<u>72.01</u>	66.74	80.61
AlignUSER+ ($\lambda_{wm}=2.0, \lambda_{CR}=0.5$)	52.81	71.76	<u>66.92</u>	<u>80.50</u>

Table 12: Sensitivity to loss weights ($\lambda_{wm}, \lambda_{CR}$) on OPeRA. All metrics are percentages (%).

Model	Action Gen. (Accuracy)	Action Type (Macro F1)	Click Type (Weighted F1)	Session Outcome (Weighted F1)
AlignUSER ($K=0$; WM only)	37.26	58.41	54.66	69.92
AlignUSER	51.47	69.81	66.29	78.07
AlignUSER (imagined CF; two-stage)	48.30	67.28	64.54	76.96
AlignUSER+ ($K=0$; WM only)	39.41	60.07	56.20	71.58
AlignUSER+	52.92	71.94	66.88	80.52
AlignUSER+ (imagined CF; two-stage)	48.91	70.60	65.75	78.02

Table 13: Model-based counterfactual reflection on OPeRA.

Method	Page Type F1 ↑	Judge Agree. ↑	Text Edit Sim. ↑
AlignUSER	0.89	0.77	0.70
AlignUSER(image)	0.92	0.81	0.73
AlignUSER+(image)	0.93	0.82	0.75

Table 14: OPeRA next-state prediction with image-conditioned world models.

acted”).

E.3 LLM Evaluator Prompt

The prompt below was employed to distinguish between humans and AI-generated interactions:

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 Reproducibility

All prompts used in the framework, including those for world modeling and counterfactual alignment, are provided in the repository at https://github.com/simul-ci/align_user. Detailed instructions for data preprocessing, hyperparameter settings, and prompts are also included.