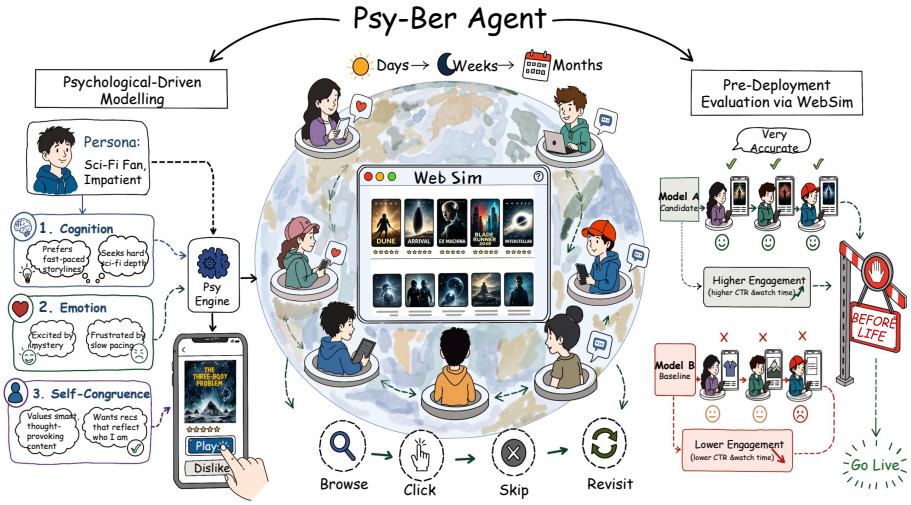
PSYBER-AGENT: A PSYCHOLOGY-DRIVEN USER AGENT FOR RECOMMENDER SYSTEMS

Anonymous authors
Paper under double-blind review



ABSTRACT

Evaluating recommender systems before deployment requires simulators that remain realistic over long, closed-loop interaction cycles. Current LLM-based agents generally function as surface imitators that generate realistic short-term responses but fail to capture the underlying user dynamics such as fatigue, affect, and shifting preferences. We propose **PsyBer-Agent**, a psychology-inspired simulator that models these dynamics through its **Psy-Engine**. This engine represents each user with three evolving, model-defined latent states. To calibrate simulated behavior against real interaction logs, we align log-conditioned reference trajectories with simulator-generated latent-state trajectories via Unbalanced Gromov–Wasserstein optimal transport. This approach provides structured supervision for a lightweight policy without the computational burden of fine-tuning the backbone LLM. We also introduce WebSim and the PsyBer Benchmark, which encompass multimodal scenarios and seven recommender backbones. Our framework evaluates behavioral fidelity, latent-state consistency, robustness, and responses to social-cue perturbations. On the tested horizons, PsyBer-Agent shows lower behavioral discrepancy than the prompt-only simulators while maintaining stable performance under large-scale concurrent simulation for up to 5M queued simulation tasks under bounded concurrency. Code is available in the supplementary material and at:
PsyBer-Agent: `https://anonymous.4open.science/r/CC66F/`,
WebSim Platform: `https://anonymous.4open.science/r/D8EAX/`.

1 INTRODUCTION

Modern recommender systems rely on rapid iteration, yet collecting realistic user feedback through surveys, user studies, or online A/B tests remains costly, slow, and risky Grbovic & Cheng (2018); Gomez-Uribe & Hunt (2015). Offline log replay is cheaper but misses post-deployment counterfactual interaction dynamics Ie et al. (2019). Advances in LLMs and agentic systems make pre-deployment user simulation increasingly plausible Huang et al. (2025); Gur et al. (2023); Zhang et al. (2024b): LLM-based agents can generate trajecto-

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

ries such as clicks, skips, and continued browsing Zhang et al. (2024a); Wang et al. (2025b), providing a richer testbed for recommender evaluation, A/B testing, and robustness analysis.

Table 1: Comparison of agent-based user simulators. "Dyn." (dynamic state modeling), "Psy." (psychological modeling), and "Align." (real-log alignment).

Method	Year	Scale	Dyn.	Psy.	Align.
RecAgent Wang et al. (2025a)	2023	50	✓	✗	▲
Agent4Rec Zhang et al. (2024a)	2024	1K	✓	✗	▲
UserSim Zhang et al. (2025b)	2024	Data	▲	✗	▲
SimUSER Bougie & Watanabe (2025)	2025	1K	✓	▲	▲
RecUserSim Chen et al. (2025)	2025	3K	✓	▲	▲
PUB Ma et al. (2025)	2025	Data	▲	▲	✓
DyTA4Rec Wanyan et al. (2025)	2025	6K	✓	✗	✓
RecInter Jin et al. (2025)	2025	1K	✓	✗	✓
PsyBer-Agent	**2026**	**5M**	✓	✓	✓

✓: explicit; ▲: partial/implicit; ✗: no explicit support.

However, current LLM-based user simulators remain unreliable as pre-deployment evaluation proxies Seshadri et al. (2026); Chen et al. (2026). Existing methods improve local realism through personas, memory, feedback distillation, or latent reasoning, but many remain primarily prompt-driven: they generate plausible local actions without an explicit, controllable state-transition mechanism connecting history to subsequent behavior. As a result, local errors may accumulate into behavioral drift, making these simulators brittle across environments and long-horizon interactions Chaney et al. (2018); Nakash et al. (2025). More specifically, existing systems generally do not expose a unified computational mechanism for representing how curiosity-, fatigue-, social satisfaction-, and self-congruence-related signals change under repeated exposure Knijnenburg et al. (2012). This limits their ability to represent cumulative preference dynamics in a controlled and interpretable state space. Practical recommender evaluation also requires scalable and reproducible multi-user simulation for model comparison, A/B discrimination, and robustness analysis, while existing systems remain mostly restricted to single-agent or small-scale settings Gilotte et al. (2018). Table 1 summarizes these differences.

To address these limitations, we propose **PsyBer-Agent**, a psychology-inspired user simulator with an explicit model-defined latent-state engine. PsyBer-Agent builds on `Psy-Engine` to model three evolving user-state dimensions: cognition, emotional/social satisfaction, and self-congruence. These model-defined states interact with personas, memory, multimodal item representations, so behavior is generated through explicit state transitions rather than prompt-only role-playing. We further introduce a psycho-behavioral alignment framework based on Unbalanced Gromov-Wasserstein (UGW) optimal transport Séjourné et al. (2021), which converts offline expert trajectories into structured supervision for a lightweight online policy without fine-tuning the backbone LLM. The main workflow of PsyBer-Agent can be seen in Figure 1.

Built on PsyBer-Agent, we introduce `PsyBer` Benchmark and the `WebSim` Platform for simulated-user A/B testing and recommender model evaluation. The benchmark evaluates behavior fidelity, latent-state tracking, decision robustness, and efficiency under controlled closed-loop rollouts. Across WebSim evaluations, PsyBer-Agent more faithfully matches real interaction logs, better matches model-defined latent-state trajectories, and remains stable under perturbations and stress. By distilling expensive backbone reasoning into a lightweight online policy, it further supports industrial-scale concurrent simulation at up to 5M queued simulation tasks under bounded concurrency, with bounded latency growth, high success rates, and explicit cost accounting. Together, these results establish PsyBer-Agent as a practical and scalable proxy for pre-deployment recommender evaluation. Our contributions are:

- We propose **PsyBer-Agent**, a psychology-inspired user simulator with explicit, model-defined latent-state dynamics for controllable long-horizon recommender simulation.
- We introduce a psycho-behavioral alignment framework that uses **Psy-Engine** and UGW alignment to distill observed offline behavior into efficient online policies without backbone fine-tuning.
- We build **PsyBer Benchmark** and **WebSim**, a unified testbed for closed-loop A/B testing, behavior-fidelity evaluation, robustness analysis, and million-scale simulation.

2 Problem Formulation and Notation

Given implicit interaction logs $\mathcal{D} = \{(u, i, t)\}$, where u denotes a user, i denotes an interacted item, and t denotes the interaction timestamp. We reconstruct each user's expert sequence $\tau_u^{\text{exp}} =$

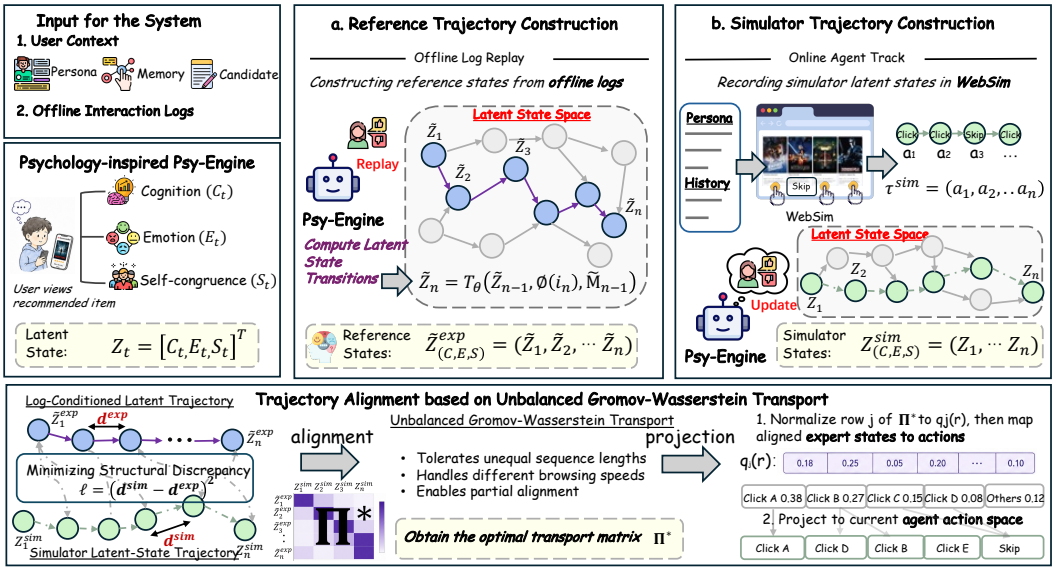


Figure 1: The workflow of PsyBer-Agent. PsyBer-Agent is mainly composed of **Psy-Engine** and **Psycho-behavioral Alignment**.

$(i_{u,1}, \ldots, i_{u,T_u})$. At step t, the simulator observes a top-K slate $\mathcal{L}_{u,t} = \{\ell^{(1)}_{u,t}, \ldots, \ell^{(K)}_{u,t}\}$ and selects an action $a_{u,t} \in \{1, \ldots, K, \texttt{skip}\}$ from the available action set.

Each item is represented by $\phi(i) = (\mathbf{v}_i, n_i, c_i, p_i)$, where $\mathbf{v}_i$ is the semantic embedding, and n_i, c_i, p_i denote identity gap, cognitive load, and social proof. Cross-domain feature construction is detailed in Appendix A.1. The users' psychological state is $\mathbf{z}_{u,t} = [C_{u,t}, E_{u,t}, S_{u,t}]^\top \in [0,1]^3$, with persona ψ_u and memory $\mathcal{M}_{u,t}$, and evolves as

$$\mathbf{z}_{u,t+1} = \mathcal{T}_\Theta(\mathbf{z}_{u,t}, \mathcal{L}_{u,t}, a_{u,t}, \mathcal{M}_{u,t}, \psi_u). \tag{1}$$

To align $\mathbf{z}_{u,t}$with $\tau^{\exp}_u$, we train the fast policy π_θ, we align simulated and expert trajectories via UGW, and obtain soft action targets $\hat{\mathbf{p}}_{u,t}$:

$$\mathcal{L}_{\text{policy}} = \sum_u \sum_t \text{CE}(\hat{\mathbf{p}}_{u,t}, \pi_\theta(\cdot \mid \mathbf{z}_{u,t}, \mathcal{L}_{u,t})) \,. \tag{2}$$

Problem Statement

Given implicit logs $\mathcal{D}$, can aligning observed user behaviors with psychology-inspired, model-defined state transitions yield a reliable behavioral simulator for recommendation systems?

3 PsyBer-Agent Framework

Overview. PsyBer-Agent couples psychology-inspired latent-state modeling with psycho-behavioral alignment: *Psy-Engine* defines user state, transition dynamics, and action evaluation, while *PsyBer-Agent* constructs multimodal item features, reconstructs log-conditioned latent reference trajectories, and learns a lightweight, fast policy via UGW-based alignment.

The C/E/S states are computational constructs motivated by psychological principles, so CES-MSE measures internal consistency within this model-defined state space rather than validated correspondence to independently measured human psychological states. PsyBer-Agent is implemented based on Camel-AI Li et al. (2023).

3.1 Simulation Experiment Platform Construction

We build **WebSim**, an end-to-end platform for multimodal asset preparation, offline alignment/policy training, and browser-level interactive simulation (details in Appendix C.5). For each dataset, WebSim creates a multimodal semantic view and a psycho-behavioral view for Psy-Engine: each item i is encoded by $\mathbf{v}_i$ with features (n_i, c_i, p_i), and each user is initialized with persona anchors $(\mathbf{u}_{\text{base}}, \mathbf{u}_{\text{ideal}}, \mathbf{u}_{\text{real}})$. During simulation, a web service and browser controller execute click/pagination actions, use the trained fast policy when available or a constrained vision-language fallback otherwise, and log traces, psychological states, and run summaries for reproducible evaluation.

Persona Anchor Construction. For each user u, WebSim constructs three pre-window persona anchors: $\mathbf{u}_{\text{base}}$ encodes available profile and metadata, $\mathbf{u}_{\text{ideal}}$ summarizes long-term preferences, and $\mathbf{u}_{\text{real}}$ summarizes the most recent K temporally ordered interactions. Item representations are pooled with a masked mean and projected into a shared persona space; missing fields and empty histories use fixed cold-start vectors with an availability mask. These anchors condition the simulator and are not interpreted as direct measurements of psychological traits. Information from the simulation window and later periods is excluded.

3.2 PSY-ENGINE MODELING

We design Psy-Engine as a psychology-inspired latent-state engine for history-dependent user behavior. At each step t, the simulator maintains $\mathbf{z}_t = [C_t, E_t, S_t]^\top \in [0,1]^3$, with coordinates motivated by cognitive reserve C_t, emotional satisfaction E_t, and self-congruence S_t, respectively. These variables serve as model-defined latent states for behavior generation, with their psychological interpretations providing modeling priors.

State Transition Model. Given the consumed item i_t with features $\phi(i_t) = (\mathbf{v}_t, n_t, c_t, p_t)$ and memory bank $\mathcal{M}_t$, here, n_t, c_t, and p_t represent incongruence, a cognitive-load proxy, and social proof. Psy-Engine computes novelty ν_t, semantic saturation σ_t, semantic fit β_t, serendipitous interest ρ_t, and self-conflict γ_t. Detailed definitions are provided in Appendix A. We use $\text{Proj}_{[0,1]}(x) = \min(1, \max(0, x))$ to keep the state coordinates bounded.

Psy-Engine organizes these signals into an attractiveness proxy A_t, a fatigue proxy F_t, and two interaction-dependent targets, emotional satisfaction E_t and self-congruence S_t, here, $E_t^\star$ and $S_t^\star$ denote target states computed from the current item.:

$$A_t = f_A(\nu_t, n_t), \quad F_t = f_F(c_t, \sigma_t), \quad E_t^\star = f_E(\beta_t, \rho_t, p_t), \quad S_t^\star = f_S(\beta_t, \gamma_t). \tag{3}$$

The inputs to these functions are motivated by exploratory interest Loewenstein (1994); Silvia (2008), cognitive load and satiation Sweller (1988); Redden (2008), affective satisfaction and social influence Kotkov et al. (2016); Cialdini & Goldstein (2004), and self-congruence Higgins (1987); Sirgy (1982). We instantiate these functions as fixed, bounded aggregation rules; their exact definitions are provided in Appendix A. Their specific forms are computational choices used to operationalize the corresponding modeling motivations, rather than uniquely determined psychological laws.

The resulting state is updated through a bounded relaxation operator:

$$\mathbf{z}_{t+1} = \text{Proj}_{[0,1]^3}\left(\mathbf{z}_t + \alpha \mathbf{d}_t\right), \quad \mathbf{d}_t = \begin{bmatrix} A_t - F_t \\ E_t^\star - E_t \\ S_t^\star - S_t \end{bmatrix}, \tag{4}$$

where $\mathbf{z}_t = [C_t, E_t, S_t]^\top$, $\text{Proj}_{[0,1]^3}$ keeps each coordinate in $[0,1]$, and $\alpha \in [0,1]$ controls the relaxation rate. Thus, the cognitive coordinate responds to the net attractiveness–fatigue signal, whereas the affective and self-congruence coordinates move toward interaction-dependent targets. We use a shared update rate to keep the parameterization compact, rather than to assume identical psychological time constants. We set $\alpha = 0.35$ following the sensitivity analysis in Appendix E.2. This bounded transition is a parsimonious computational hypothesis for history-dependent behavior; its coordinate-wise expansion is provided in Appendix A.2. For subsequent trajectory construction, we denote the complete Psy-Engine transition by

$$\mathbf{z}_{t+1} = \mathcal{T}_\Theta\left(\mathbf{z}_t, \phi(i_t), \mathcal{M}_t\right), \tag{5}$$

where $\mathcal{T}_\Theta$ summarizes the bounded state update defined above and Θ denotes the fixed Psy-Engine parameters. Implementation details are provided in Appendix B.1.

3.3 PSYCHO-BEHAVIORAL ALIGNMENT

Offline logs lack annotations of users' latent decision states. PsyBer-Agent replays logged actions in Psy-Engine to construct model-defined latent-state trajectories and derive state-conditioned soft supervision, enabling policy imitation under comparable computational states without treating them as direct psychological measurements. Details are provided in Appendix B.2.

Model-Defined State Sequence Construction. Offline browsing logs record clicks but do not provide annotations of the latent states underlying those decisions. We therefore use the logged clicks as forced inputs to the deterministic Psy-Engine transition and reconstruct a model-defined reference trajectory for subsequent alignment. Let $\tau^{\text{exp}} = (i_1, \ldots, i_n)$ denote an expert click

sequence, where i_r is the r-th clicked item. For comparison, let $\mathbf{Z}^{\mathrm{sim}} = (\mathbf{z}_1^{\mathrm{sim}}, \ldots, \mathbf{z}_m^{\mathrm{sim}})$ denote the simulator-generated trajectory, with $\mathbf{z}_j^{\mathrm{sim}} \in [0,1]^3$ encoding cognitive reserve (C), emotional satisfaction (E), and self-congruence (S). Starting from the persona-conditioned initial state $\tilde{\mathbf{z}}_0$ and replay memory $\tilde{\mathcal{M}}_0$, we apply teacher forcing by treating each logged expert item as the consumed item and recursively obtain the model-defined pseudo-expert trajectory $\tilde{\mathbf{Z}}^{\mathrm{exp}} = (\tilde{\mathbf{z}}_1, \ldots, \tilde{\mathbf{z}}_n)$:

$$\tilde{\mathbf{z}}_r = \mathcal{T}_\Theta\left(\tilde{\mathbf{z}}_{r-1}, \phi(i_r), \tilde{\mathcal{M}}_{r-1}\right), \quad r = 1, \ldots, n. \tag{6}$$

The resulting state $\tilde{\mathbf{z}}_r$ serves as the model-defined reference state at the r-th expert interaction for trajectory alignment and state-conditioned supervision. Here, $\mathcal{T}_\Theta$ denotes the item-level Psy-Engine transition defined above, $\phi(i_r) = (\mathbf{v}_r, n_r, c_r, p_r)$ is the psycho-behavioral feature tuple of item i_r, and $\tilde{\mathcal{M}}_{r-1}$ stores the normalized embeddings of previously replayed expert items. Detailed construction procedures are provided in Appendix B.1.

Prior-Guided Latent-State Trajectory Alignment. We align the model-defined expert trajectory and the simulator trajectory using an Unbalanced Gromov–Wasserstein (UGW) transport plan Séjourné et al. (2021). Because the expert trajectory contains clicks but no latent-state annotations, and may differ from the simulator trajectory in length and browsing rhythm, we first construct a soft state prior from expert prefixes:

$$\hat{\mathbf{z}}_r = g_\varphi\left(\psi, \tau_{1:r}^{\mathrm{exp}}, \phi(i_r), \frac{r}{n}\right), \tag{7}$$

where ψ denotes the user persona and $\tau_{1:r}^{\mathrm{exp}} = (i_1, \ldots, i_r)$ is the expert prefix. The regressor g_φ maps each prefix into the simulator state space, detailed descriptions are provided in Appendix B.2. For simulator step j and expert position r, we define an initialization cost $c_{j,r}^{\mathrm{init}}$ that combines state and local-dynamics discrepancies:

$$c_{j,r}^{\mathrm{init}} = \lambda_z \left\|\mathbf{z}_j^{\mathrm{sim}} - \hat{\mathbf{z}}_r\right\|_2^2 + \lambda_\Delta \left\|\Delta\mathbf{z}_j^{\mathrm{sim}} - \Delta\hat{\mathbf{z}}_r\right\|_2^2, \tag{8}$$

where $\mathbf{z}_j^{\mathrm{sim}}$ denotes the model-defined latent state generated by the simulator at step j, while $\hat{\mathbf{z}}_r$ denotes the latent state predicted by the regressor from the expert trajectory prefix up to position r, $\Delta\mathbf{z}_j^{\mathrm{sim}} = \mathbf{z}_j^{\mathrm{sim}} - \mathbf{z}_{j-1}^{\mathrm{sim}}$ and $\Delta\hat{\mathbf{z}}_r = \hat{\mathbf{z}}_r - \hat{\mathbf{z}}_{r-1}$, with $\Delta\mathbf{z}_1^{\mathrm{sim}} = \Delta\hat{\mathbf{z}}_1 = \mathbf{0}$. The nonnegative coefficients λ_z and λ_Δ control the contributions of the two discrepancies; further details are provided in Appendix B.2

Let $\Pi = (\Pi_{j,r})$ denote the transport plan, where $\Pi_{j,r}$ is the soft matching weight between simulator step j and expert position r. We initialize Π from the cost $c_{j,r}^{\mathrm{init}}$ as

$$\Pi_{j,r}^{(0)} \propto \exp\left(-\frac{c_{j,r}^{\mathrm{init}}}{\tau_0}\right), \qquad \Pi^\star = \arg\min_{\Pi \geq 0} \mathcal{L}_{\mathrm{UGW}}(\Pi), \tag{9}$$

where $\tau_0 > 0$ controls the sharpness of the soft matching. Starting from $\Pi^{(0)}$, UGW minimizes $\mathcal{L}_{\mathrm{UGW}}(\Pi)$ over nonnegative transport plans to obtain the optimized plan $\Pi^\star$. The detailed UGW alignment process can be found in Appendix B.2.

Alignment-guided Fast Policy Learning and Inference. Because trajectory alignment may leave some simulator steps without direct expert actions, we distill $\Pi^\star$ into soft action targets for a lightweight fast policy used as a fallback during online simulation. The optimized transport plan $\Pi^\star$ is row-normalized into step-wise soft supervision, where $q_j(r)$ denotes the alignment weight between simulator step j and expert position r. The aligned expert-item weights are projected onto the executable action set $\mathcal{A}_j = \mathcal{L}_j \cup \{\texttt{skip}\}$, where matched items contribute to their corresponding slate actions and residual unmatched mass is assigned to `skip`, yielding the soft action label $\hat{p}_j$ over $\mathcal{A}_j$. The fast policy is trained by

$$q_j(r) = \frac{\Pi_{j,r}^\star}{\sum_{r'=1}^{n} \Pi_{j,r'}^\star + \delta},\ \delta > 0, \quad \mathcal{L}_{\mathrm{policy}} = \sum_j \mathrm{CE}\Big(\hat{p}_j, \pi_\theta(\cdot \mid \mathbf{z}_j^{\mathrm{sim}}, \mathcal{A}_j)\Big). \tag{10}$$

At inference, the fast policy selects an action from the same executable action set:

$$a_j^{\mathrm{fast}} = \arg\max_{a \in \mathcal{A}_j} \pi_\theta\left(a \mid \mathbf{z}_j^{\mathrm{sim}}, \mathcal{A}_j\right), \quad \mathcal{A}_j = \mathcal{L}_j \cup \{\texttt{skip}\}. \tag{11}$$

The selected action is used directly or as a top-k hint for PsyBer-Agent, with the backbone LLM invoked only when needed. The fast policy uses latent-state, persona, memory, and candidate-item features. It falls back to the LLM/VLM when its confidence is insufficient or the action space is unsupported, and uses random selection only if that route also fails. This enables efficient online inference from aligned psycho-behavioral supervision.

4 PSYBER BENCHMARK

4.1 BENCHMARK COVERAGE

The PsyBer Benchmark consists of two core tasks: **(1.)** Multi-scenario user online simulation. **(2.)** Evaluate the performance of the recommendation model. The Figure 3 shows detailed data statistics. Detailed descriptions of the task definitions are provided in Appendix C.3.

4.2 TASK EVALUATION

WebSim Experiment Platform. We introduce **WebSim**, a Flask-based closed-loop simulator that decouples user interaction from recommender inference and uses random fallback for unavailable or empty model outputs, enabling consistent cross-model and cross-dataset evaluation. Additional implementation details are provided in Appendix C.5.

Recommender Models and Datasets. To support recommendation simulation across diverse scenarios, WebSim is built on a set of representative recommendation backbones, including PopRec Cremonesi et al. (2010), BPR-MF Rendle et al. (2012), LightGCN He et al. (2020), Multi-VAE Liang et al. (2018), GRU4Rec Hidasi et al. (2015), SASRec Kang & McAuley (2018), and BERT4Rec Sun et al. (2019). To construct PsyBer-Test, a multi-scene and multi-modal benchmark, we integrate recommendation datasets from several representative application scenarios, including MM-ML-1M Zhang et al. (2026b) for movie viewing, the Amazon Review Hou et al. (2024) for shopping, and MicroLens Ni et al. (2025) and NineRec Zhang et al. (2025a) for image-text information flow and short-video browsing scenarios. Detailed data statistics are provided in Appendix C.2 and C.4.

Evaluation Scope. We evaluate PsyBer-Agent from four aspects: Online A/B test evaluation, behavior fidelity, baseline simulator comparison, and scalability and cost. Detailed metrics design is provided in the Appendix C.6.

5 EXPERIMENTS

5.1 EXPERIMENTAL SETTINGS

Backbone Recommender Models. We evaluate the proposed PsyBer-Agent on representative baseline models from the PsyBer Benchmark. Detailed descriptions and core mechanisms of these models are provided in Appendix D.2.

User Simulator Baselines. We compare representative agent-based simulators covering social, conversational, RL-based, and platform-oriented recommendation settings. Details and the protocol-compatibility and reproducibility criteria for quantitative inclusion are provided in Appendix D.3.

Dataset & Hardware Setting. We split each user's interactions chronologically into 70/10/20 train/-validation/test sets and evaluate on held-out users. Hardware details are reported in Appendix D.5.

5.2 EVALUATION AND ANALYSIS

We evaluate PsyBer-Agent through three main experiments and three diagnostic analyses. The main experiments examine: (**I.**) behavior fidelity at both population and individual levels, (**II.**) baseline simulator comparison, and (**III.**) scalability and token cost under high-concurrency simulation. The diagnostic analyses include (**IV.**) component ablation, (**V.**) information-channel ablation, and (**VI.**) an offline backbone sanity check. Additional analyses of group behavior, parameter sensitivity, robustness, unit cost, and fallback usage are reported in Appendices E.1, E.2, E.3, and D.5.

PsyBer-Agent Online A/B Test Evaluation. We compare 500 human users with 500 simulated users under the same recommendation interface and fixed backend, with the decision-maker as the only varying factor. HR10, NDCG10, and CTR are computed using the same protocol, held-out relevance targets, and replayed candidate slates for both groups. The full PsyBer-Agent with Qwen3-VL-235B closely matches human behavior, with absolute metric differences of 0.0030, 0.0016, and 0.0040, respectively. Removing memory increases these differences to 0.0160, 0.0106, and 0.0220,

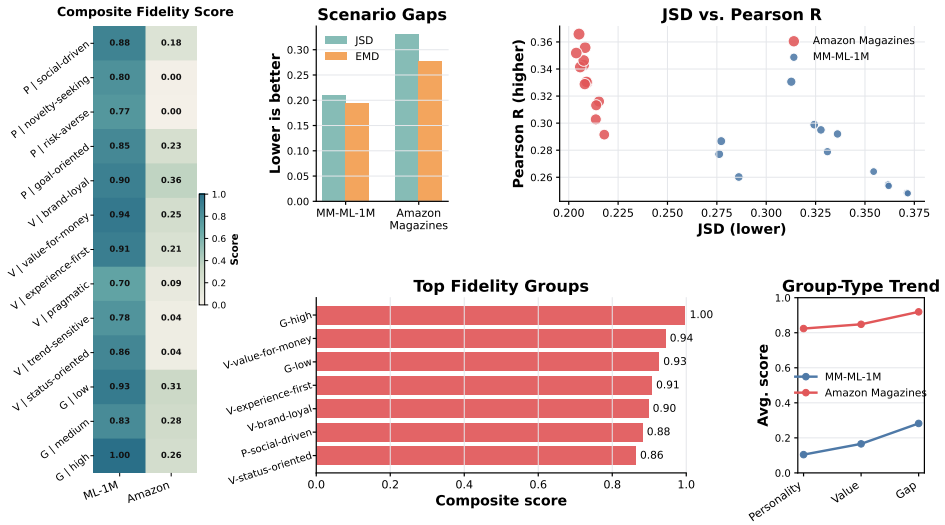


Figure 2: PsyBer-Agent simulation experiments and profile-conditioned population analysis.

while additionally removing the user profile increases them to 0.0400, 0.0246, and 0.0500. These results indicate that memory and user-profile modeling are important for behavioral agreement under a fixed backend. They do not establish user-satisfaction prediction or the ranking of alternative recommenders in a live A/B test. The fixed-backend agreement protocol is detailed in Appendix D.6.

Table 2: Performance comparison of PsyBer-Agent and Human Users across different configurations.

Perspective	Method	PsyBer-Agent			Human Users		
		HR@10 ↑	NDCG@10 ↑	CTR ↑	HR@10 ↑	NDCG@10 ↑	CTR ↑
Backbone Scaling	PsyBer-Agent + Qwen3-VL-235B	0.2350	0.1150	0.2960	0.2380	0.1166	0.3000
Mechanism Validation	PsyBer-Agent	0.2350	0.1150	0.2960	0.2380	0.1166	0.3000
	w/o Memory	0.2220	0.1060	0.2780	0.2380	0.1166	0.3000
	w/o User Profile	0.1980	0.0920	0.2500	0.2380	0.1166	0.3000

Findings 1

PsyBer-Agent's predictions show low discrepancy with human behavior under the tested fixed-backend protocol, and both memory and user-profile modeling are key to preserving this agreement.

Agent Behavior Fidelity Evaluation.

We assess PsyBer-Agent's behavioral realism across scenarios and recommender backbones using three complementary metrics: JSD for population-level interaction-distribution mismatch, Pearson correlation (R) for item-level agreement with real interaction frequencies, and EMD for downstream behavioral discrepancy after matched prefixes. As shown in Table 3 and Figure 2, PsyBer-Agent preserves realistic behavioral patterns across scenarios and backbones, with stronger alignment on Amazon Magazine Subscriptions than on MM-ML-1M. Amazon Magazine Subscriptions forms compact low-JSD/high-R/low-EMD clusters, while MM-ML-1M is more dispersed, suggesting that narrower e-commerce preferences are easier to simulate than semantically diverse movie-consumption dynamics.

Table 3: Behavioral realism results across scenarios and recommender backbones.

Scenarios	Rec Models	JSD↓	Pearson R↑	EMD↓
MM-ML-1M	BPR-MF	0.3304	0.1615	0.3711
	LightGCN	0.5603	0.1867	0.1649
	Multi-VAE	0.3249	0.1217	0.3673
	GRU4Rec	0.3551	0.1755	0.3643
	SASRec	0.6515	0.1803	0.4602
	BERT4Rec	0.3292	0.1796	0.3683
Amazon Magazine Subscriptions	BPR-MF	0.2085	0.5164	0.2315
	LightGCN	0.2126	0.5466	0.2245
	Multi-VAE	0.2094	0.5436	0.2333
	GRU4Rec	0.2105	0.5455	0.2386
	SASRec	0.2106	0.5437	0.2357
	BERT4Rec	0.2173	0.5500	0.2491

To further summarize profile-conditioned fidelity, we compute a composite fidelity score in Figure 2 by combining normalized JSD, Pearson R, and EMD into a single diagnostic ranking score. The resulting group-level patterns show that identity-gap and value-oriented groups achieve the highest fidelity, indicating that PsyBer-Agent preserves stable behavioral signatures across heterogeneous profile groups. Overall, the remaining mismatch appears more related to scenario and recommender effects than to unstable persona conditioning. Detailed group-level analysis results corresponding to Figure 2 are provided in Appendix E.5.

Findings 2

PsyBer-Agent preserves group-level behavioral signatures across scenarios, and the remaining fidelity gap appears to be driven more by scenario complexity and recommender exposure than by unstable persona conditioning.

Baseline Simulator Comparison. To compare PsyBer-Agent with existing user-simulation baselines under a controlled setting, we conduct a fixed-candidate replay experiment on MM-ML-1M with 500 users, 10 decision steps per user, and five random seeds. At each step, all simulators observe the same user histories, item metadata, and candidate pages generated under LightGCN and SASRec, and are evaluated on decision accuracy, CTR calibration, distributional fidelity, item-level correlation with real logs, and latency. As shown in Table 4, PsyBer-Agent achieves the best overall performance in this controlled comparison, with the highest accuracy and correlation as well as the lowest CTRGap, JSD, and inference latency among all methods. These results suggest that explicit psychological state modeling yields a more faithful user-behavior prior for replay-based recommendation simulation.

Table 4: Basic simulator fidelity comparison.

Simulator	Acc. ↑	CTRGap ↓	JSD ↓	Pearson R ↑	Latency ↓
RecAgent Wang et al. (2025a)	0.212	0.482	0.420	0.264	5.46s
Agent4Rec Zhang et al. (2024a)	0.193	0.489	0.447	0.264	5.74s
UserSim Zhang et al. (2025b)	0.128	0.500	0.605	0.142	3.31s
DyTA4Rec Wanyan et al. (2025)	0.254	0.455	0.335	0.281	5.91s
PsyBer-Agent	**0.275**	**0.406**	**0.286**	**0.305**	**2.53s**

PsyBer-Agent Scalability & Cost Evaluation. We evaluate whether PsyBer-Agent remains efficient under large-scale workloads by running $N \in \{500, 1\text{K}, 100\text{K}, 1\text{M}, 5\text{M}\}$ simulation tasks over 3 runs and reporting throughput, p95 latency/decision, wall-clock time, success rate, and token cost. Detailed metric definitions are provided in Appendix C.6. As shown in Table 5, PSYBER-AGENT scales to 5M tasks with bounded p95 latency growth (2.52s to 3.80s), success rates above 91%, aggregate throughput of approximately 4.1 tasks/s after queue warm-up for workloads of at least $N \geq 100\text{K}$, and explicit wall-clock/token-cost accounting, indicating stable million-scale simulation under external model-call overhead. Hardware settings, unit cost, and fallback statistics are provided in Appendix D.5.

Table 5: Scalability and cost of PsyBer-Agent under large-scale workloads.

Workload Size N	Throughput (tasks/s) ↑	p95 Lat. /Decision (s) ↓	Wall-clock Time ↓	Success Rate ↑	Token Cost ↓
500	1.00	2.52s	8.3 mins	100.00%	4262.7K
1K	0.94	2.65s	17.8 mins	100.00%	10265.3K
100K	4.18	2.74s	398.4 mins	92.75%	983.31M
1M	4.12	3.27s	4058.8 mins	91.34%	10277.83M
5M	4.09	3.80s	20358.9 mins	95.18%	52262.77M

Findings 3

PsyBer-Agent maintains bounded latency and high completion rates up to 5M tasks. Scaling behavior is dominated by predictable growth in wall-clock time and token cost.

5.3 ABLATION STUDY

5.3.1 COMPONENT ABLATION

We first examine the constituent components of PsyBer-Agent by ablating the preprocessing agent, Psy-Engine, unbalanced Gromov–Wasserstein alignment, and fusion memory. As shown in Table 6, the full configuration achieves the lowest CTRGap, Exp-JS, and CES-MSE among the evaluated variants. Removing the preprocessing agent, Psy-Engine, or transport alignment increases one or more discrepancies, suggesting that structured input conditioning, model-defined state transitions, and distribution-level trajectory alignment provide complementary benefits under the tested protocol. Removing fusion memory has a smaller effect on CTRGap than removing several other components, but increases CES-MSE, indicating that memory remains useful for maintaining internal state consistency. These results support the contribution of the individual modules without implying psychological validity of the latent states.

Table 6: Ablation study of PsyBer-Agent components.

Architecture / Variants	Simulation Fidelity		
	CTRGap (↓)	Exp-JS (↓)	CES-MSE (↓)
Direct Agent w/ Prompt	0.8926	0.7852	0.1654
w/o Preprocessing Agent	0.8652	0.7613	0.1290
w/o Psy-Engine	0.7469	0.7382	0.1154
w/o Optimal Transport	0.6906	0.6817	0.1053
w/o Fusion Memory	0.2985	0.6764	0.1254
PsyBer-Agent (Ours Full)	**0.2608**	**0.4297**	**0.0636**

5.3.2 Information Ablation

Beyond component-level ablations, we evaluate the information channels used by PsyBer-Agent, including item textual and visual information, psychological-change signals, user identity profiles, and scene information. As shown in Table 7, the full configuration obtains the best overall results under the rerun protocol, while removing individual channels increases at least one of the reported discrepancies. The observed changes indicate that item content, dynamic state signals, user information, and scene context provide complementary conditioning information. These ablations measure sensitivity to the available input channels and do not constitute independent evidence that the latent variables correspond to psychologically validated traits. All ablation variants use the same data split, candidate-replay protocol, and random-seed configuration; variants that include the fast policy are retrained under their corresponding ablated state or information setting.

Table 7: Ablation study of information channels in PsyBer-Agent across simulation-fidelity metrics.

Architecture / Variants	Simulation Fidelity		
	CTRGap (↓)	Exp-JS (↓)	CES-MSE (↓)
Direct Agent w/ Prompt	0.8926	0.7852	0.1654
w/o Item Information	0.7824	0.6770	0.1282
w/o Psychological Changes	0.8816	0.6804	0.1254
w/o User Identity Profile	0.6738	0.6761	0.1086
w/o Scene Information	0.5243	0.6783	0.0855
PsyBer-Agent (Ours Full)	**0.2608**	**0.4297**	**0.0636**

6 Related work

User Simulation for Recommender Evaluation. Recommender evaluation is moving from static log replay toward LLM-driven user simulation. Recent methods improve realism through persona and working-memory architectures Chen et al. (2025); Liu et al. (2025); Bernard & Balog (2025), align simulated behavior with human preferences via feedback distillation Wei et al. (2025), leverage latent reasoning signals for recommendation interactions Zhang et al. (2025b), and generate human-camouflaged adversarial users Gu et al. (2025a). However, most simulators remain prompt-dependent and model plausible local actions rather than explicit internal dynamics such as fatigue, curiosity, and self-consistency over long horizons. PsyBer-Agent addresses this gap with an explicit psychological engine for motivation-aligned behavior beyond prompt-only imitation.

Psychological User Modeling and Behavioral Calibration. Psychological and calibration-oriented modeling has been used to improve user and agent realism through Big Five profiles Ma et al. (2025); Zhao et al. (2025), contextual cognitive alignment Feng et al. (2025), broader value and motivation modeling Peng et al. (2025); Gebreegziabher et al. (2026), and co-evolving user-system feedback Kim et al. (2026). Beyond recommendation, generative-agent calibration emphasizes behavior-level alignment, including implicit alignment to target behavioral distributions Wang et al. (2025c), uncertainty-aware decision calibration Solopova et al. (2026), reproducible social simulation Taillandier et al. (2025), and multi-level, context-sensitive alignment criteria Zeng et al. (2025) across mobility Vu et al. (2025), economic-expectation Gusella & Vicario (2025), and public-health simulations Hou et al. (2025). However, existing approaches still under-specify how internal motivation states evolve over time and how they should align with real interaction logs.

Robustness, Safety, and Large-scale Simulation. LLM-centered recommender systems and agent-based simulation introduce new robustness and safety risks, including targeted poisoning Guan et al. (2026), persistent backdoors Ning et al. (2025), fragile explanation modules Zhang et al. (2026a), and membership-inference leakage He et al. (2025). Agentic attacks further increase realism through LLM-optimized manipulation trajectories Ning et al. (2024) and simulator-leveraged shilling Gu et al. (2025b), while LLM-as-recommender systems remain susceptible to preference-oriented biases and exploits Tang et al. (2026). These risks motivate user simulators that support behavior fidelity together with scalable stress testing, robustness analysis, privacy-aware evaluation, and controlled pre-deployment diagnosis.

7 Conclusion

We presented PsyBer-Agent, a psychology-inspired user simulator that combines model-defined latent-state dynamics, offline UGW alignment, and a fast policy with LLM/VLM fallback for closed-loop recommender evaluation. Across 200 runs and 3,981 decisions, it achieved 3.11 s average latency, full fast-hint coverage, and a 99.9% batch success rate. On the PsyBer Benchmark, it reduced behavioral discrepancy and improved model-defined state consistency over prompt-based baselines under fixed replay. These results support its use as a pre-deployment simulation proxy, but do not establish psychological validity or predict live A/B rankings. Future work will add independently validated supervision and broaden evaluation across domains, horizons, and robustness settings.

STATEMENTS

AI USE STATEMENT

In this work, generative AI tools were used for language polishing and minor editorial assistance during manuscript preparation, as well as for coding assistance in the engineering of the WebSim platform. All LLM-generated text and code were reviewed, verified, and tested by the authors. The research ideas, methodology, experimental design, and all reported results were produced and validated by the authors, who take full responsibility for the final content of this work, including any text or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work uses publicly available recommendation datasets and simulated user interactions. In addition, the fixed-backend human–simulator agreement study in Section 5 involved 500 human participants, who were genuine anonymous users of the website. All participants provided informed consent before the experiment and were informed of the purpose, procedure, and duration of the study; participation was voluntary and could be withdrawn at any time without penalty. The experiment consisted solely of browsing recommendation pages and making click/skip decisions, involved no sensitive personal information, and posed minimal risk. The study was approved by the ethics committee of the authors' institution. Collected interaction logs (clicks and skips) are stored in anonymized form, are used only for research evaluation, and will not be linked to personally identifiable information. PsyBer-Agent is intended as a diagnostic proxy for pre-deployment evaluation, not as the sole basis for deployment decisions. Potential risks—including manipulative recommendation strategies, engagement-driven feedback loops, privacy concerns from interaction logs, and fairness issues across user groups—are discussed in the broader-impact statement (Appendix G.2) together with mitigation guidelines such as responsible data handling, fairness checks, human oversight, and staged online validation.

REPRODUCIBILITY STATEMENT

We provide detailed descriptions of the Psy-Engine transition rules and hyperparameters (Appendix A), the UGW alignment objective and its initialization (Appendix B), the WebSim platform implementation (Appendix C.5), all evaluation metrics (Appendix C.6), the hardware and concurrency setup (Appendix D.5), and the fixed-backend human–simulator agreement protocol, including participant recruitment and the unified metric computation. Source code for PsyBer-Agent and the WebSim platform is available at `https://anonymous.4open.science/r/CC66F/` and `https://anonymous.4open.science/r/D8EAX/`, respectively.

REFERENCES

Nolwenn Bernard and Krisztian Balog. Usersimcrs v2: Simulation-based evaluation for conversational recommender systems. *arXiv preprint arXiv:2512.04588*, 2025.

Nicolas Bougie and Narimawa Watanabe. SimUSER: Simulating user behavior with large language models for recommender system evaluation. In Georg Rehm and Yunyao Li (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 6: Industry Track)*, pp. 43–60, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-288-6. doi: 10.18653/v1/2025.acl-industry.5. URL https://aclanthology.org/2025.acl-industry.5/.

Allison JB Chaney, Brandon M Stewart, and Barbara E Engelhardt. How algorithmic confounding in recommendation systems increases homogeneity and decreases utility. In *Proceedings of the 12th ACM conference on recommender systems*, pp. 224–232, 2018.

Jiawei Chen, Ruoxi Xu, Boxi Cao, Ruotong Pan, Yunfei Zhang, Yifei Hu, Yong Du, Tingting Gao, Yaojie Lu, Yingfei Sun, et al. Towards real-world human behavior simulation: Benchmarking large language models on long-horizon, cross-scenario, heterogeneous behavior traces. *arXiv preprint arXiv:2604.08362*, 2026.

Luyu Chen, Quanyu Dai, Zeyu Zhang, Xueyang Feng, Mingyu Zhang, Pengcheng Tang, Xu Chen, Yue Zhu, and Zhenhua Dong. Recusersim: A realistic and diverse user simulator for evaluating conversational recommender systems. In *Companion Proceedings of the ACM on Web Conference 2025*, pp. 133–142, 2025.

Robert B. Cialdini and Noah J. Goldstein. Social influence: Compliance and conformity. *Annual Review of Psychology*, 55:591–621, 2004. doi: 10.1146/annurev.psych.55.090902.142015.

Paolo Cremonesi, Yehuda Koren, and Roberto Turrin. Performance of recommender algorithms on top-n recommendation tasks. In *Proceedings of the fourth ACM conference on Recommender systems*, pp. 39–46, 2010.

Yuanjun Feng, Stefan Feuerriegel, and Yash Raj Shrestha. Contextualizing recommendation explanations with llms: A user study. *arXiv preprint arXiv:2501.12152*, 2025.

Simret Araya Gebreegziabher, Yukun Yang, Charles Chiang, Hojun Yoo, Chaoran Chen, Hyo Jin Do, Zahra Ashktorab, Werner Geyer, Diego Gómez-Zará, and Toby Jia-Jun Li. The behavioral fabric of llm-powered gui agents: Human values and interaction outcomes. *arXiv preprint arXiv:2601.16356*, 2026. URL https://arxiv.org/abs/2601.16356.

Alexandre Gilotte, Clément Calauzènes, Thomas Nedelec, Alexandre Abraham, and Simon Dollé. Offline a/b testing for recommender systems. In *Proceedings of the eleventh ACM international conference on web search and data mining*, pp. 198–206, 2018.

Carlos A Gomez-Uribe and Neil Hunt. The netflix recommender system: Algorithms, business value, and innovation. *ACM Transactions on Management Information Systems (TMIS)*, 6(4):1–19, 2015.

Mihajlo Grbovic and Haibin Cheng. Real-time personalization using embeddings for search ranking at airbnb. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 311–320, 2018.

Shengkang Gu, Jiahao Liu, Dongsheng Li, Guangping Zhang, Mingzhe Han, Hansu Gu, Peng Zhang, Ning Gu, Li Shang, and Tun Lu. Llm-based user simulation for low-knowledge shilling attacks on recommender systems. *arXiv preprint arXiv:2505.13528*, 2025a.

Shengkang Gu, Jiahao Liu, Dongsheng Li, Guangping Zhang, Mingzhe Han, Hansu Gu, Peng Zhang, Ning Gu, Li Shang, and Tun Lu. Llm-based user simulation for low-knowledge shilling attacks on recommender systems. *arXiv preprint arXiv:2505.13528*, 2025b. URL https://arxiv.org/abs/2505.13528.

Guowei Guan, Yurong Hao, Jiaming Zhang, Tiantong Wu, Fuyao Zhang, Tianxiang Chen, Longtao Huang, Cyril Leung, and Wei Yang Bryan Lim. Venomrec: Cross-modal interactive poisoning for targeted promotion in multimodal llm recommender systems. *arXiv preprint arXiv:2602.06409*, 2026. URL https://arxiv.org/abs/2602.06409.

Izzeddin Gur, Hiroki Furuta, Austin Huang, Mustafa Safdari, Yutaka Matsuo, Douglas Eck, and Aleksandra Faust. A real-world webagent with planning, long context understanding, and program synthesis. *arXiv preprint arXiv:2307.12856*, 2023.

Filippo Gusella and Eugenio Vicario. Generative agents and expectations: Do llms align with heterogeneous agent models? *arXiv preprint arXiv:2511.08604*, 2025. URL https://arxiv.org/abs/2511.08604.

Jiajie He, Min-Chun Chen, Xintong Chen, Xinyang Fang, Yuechun Gu, and Keke Chen. Membership inference attacks on llm-based recommender systems. *arXiv preprint arXiv:2508.18665*, 2025. URL https://arxiv.org/abs/2508.18665.

Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 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*, pp. 639–648, 2020.

Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939*, 2015.

E. Tory Higgins. Self-discrepancy: A theory relating self and affect. *Psychological Review*, 94(3): 319–340, 1987.

Abe Bohan Hou, Hongru Du, Yichen Wang, Jingyu Zhang, Zixiao Wang, Paul Pu Liang, Daniel Khashabi, Lauren Gardner, and Tianxing He. Can a society of generative agents simulate human behavior and inform public health policy? a case study on vaccine hesitancy. *arXiv preprint arXiv:2503.09639*, 2025.

Yupeng Hou, Jiacheng Li, Zhankui He, An Yan, Xiusi Chen, and Julian McAuley. Bridging language and items for retrieval and recommendation. *arXiv preprint arXiv:2403.03952*, 2024.

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

Eugene Ie, Chih-wei Hsu, Martin Mladenov, Vihan Jain, Sanmit Narvekar, Jing Wang, Rui Wu, and Craig Boutilier. Recsim: A configurable simulation platform for recommender systems. *arXiv preprint arXiv:1909.04847*, 2019.

Song Jin, Juntian Zhang, Yuhan Liu, Xun Zhang, Yufei Zhang, Guojun Yin, Fei Jiang, Wei Lin, and Rui Yan. Beyond static testbeds: An interaction-centric agent simulation platform for dynamic recommender systems. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 18903–18920, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.956. URL https://aclanthology.org/2025.emnlp-main.956/.

Wang-Cheng Kang and Julian McAuley. Self-attentive sequential recommendation. In *2018 IEEE international conference on data mining (ICDM)*, pp. 197–206. IEEE, 2018.

Sein Kim, Sangwu Park, Hongseok Kang, Wonjoong Kim, Jimin Seo, Yeonjun In, Kanghoon Yoon, and Chanyoung Park. Self-evolverec: Self-evolving recommender systems with llm-based directional feedback. *arXiv preprint arXiv:2602.12612*, 2026. URL https://arxiv.org/abs/2602.12612.

Bart P Knijnenburg, Martijn C Willemsen, Zeno Gantner, Hakan Soncu, and Chris Newell. Explaining the user experience of recommender systems. *User modeling and user-adapted interaction*, 22(4): 441–504, 2012.

Denis Kotkov, Shuaiqiang Wang, and Jari Veijalainen. A survey of serendipity in recommender systems. *Knowledge-Based Systems*, 111:180–192, 2016. ISSN 0950-7051. doi: https://doi.org/10.1016/j.knosys.2016.08.014. URL https://www.sciencedirect.com/science/article/pii/S0950705116302763.

Guohao Li, Hasan Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. Camel: Communicative agents for" mind" exploration of large language model society. *Advances in neural information processing systems*, 36:51991–52008, 2023.

Dawen Liang, Rahul G Krishnan, Matthew D Hoffman, and Tony Jebara. Variational autoencoders for collaborative filtering. In *Proceedings of the 2018 world wide web conference*, pp. 689–698, 2018.

Fei Liu, Xinyu Lin, Hanchao Yu, Mingyuan Wu, Jianyu Wang, Qiang Zhang, Zhuokai Zhao, Yinglong Xia, Yao Zhang, Weiwei Li, et al. Recoworld: Building simulated environments for agentic recommender systems. *arXiv preprint arXiv:2509.10397*, 2025.

George Loewenstein. The psychology of curiosity: A review and reinterpretation. *Psychological bulletin*, 116(1):75, 1994.

Chenglong Ma, Ziqi Xu, Yongli Ren, Danula Hettiachchi, and Jeffrey Chan. Pub: an llm-enhanced personality-driven user behaviour simulator for recommender system evaluation. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 2690–2694, 2025. doi: 10.1145/3726302.3730238.

Itay Nakash, George Kour, Guy Uziel, and Ateret Anaby Tavor. Breaking react agents: Foot-in-the-door attack will get you in. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 6484–6509, 2025.

Jianmo Ni, Jiacheng Li, and Julian McAuley. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pp. 188–197, 2019.

Yongxin Ni, Yu Cheng, Xiangyan Liu, Junchen Fu, Youhua Li, Xiangnan He, Yongfeng Zhang, and Fajie Yuan. A content-driven micro-video recommendation dataset at scale. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, pp. 6486–6491, 2025.

Liang-bo Ning, Shijie Wang, Wenqi Fan, Qing Li, Xin Xu, Hao Chen, and Feiran Huang. Cheatagent: Attacking llm-empowered recommender systems via llm agent. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 2284–2295, 2024. doi: 10.1145/3637528.3671837. URL https://dl.acm.org/doi/abs/10.1145/3637528.3671837.

Liangbo Ning, Wenqi Fan, and Qing Li. Exploring backdoor attack and defense for llm-empowered recommendations. *arXiv preprint arXiv:2504.11182*, 2025. URL https://arxiv.org/abs/2504.11182.

Qiyao Peng, Hongtao Liu, Hua Huang, Qing Yang, and Minglai Shao. A survey on llm-powered agents for recommender systems. *arXiv preprint arXiv:2502.10050*, 2025.

Joseph P. Redden. Reducing satiation: The role of categorization level. *Journal of Consumer Research*, 34(5):624–634, 02 2008. ISSN 0093-5301. doi: 10.1086/521898. URL https://doi.org/10.1086/521898.

Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618*, 2012.

Thibault Séjourné, François-Xavier Vialard, and Gabriel Peyré. The unbalanced gromov wasserstein distance: Conic formulation and relaxation. *Advances in Neural Information Processing Systems*, 34:8766–8779, 2021.

Preethi Seshadri, Samuel Cahyawijaya, Ayomide Odumakinde, Sameer Singh, and Seraphina Goldfarb-Tarrant. Lost in simulation: Llm-simulated users are unreliable proxies for human users in agentic evaluations. *arXiv preprint arXiv:2601.17087*, 2026.

Paul J Silvia. Interest—the curious emotion. *Current directions in psychological science*, 17(1): 57–60, 2008.

M. Joseph Sirgy. Self-concept in consumer behavior: A critical review. *Journal of Consumer Research*, 9(3):287–300, 1982.

Veronika Solopova, Viktoria Skorik, Maksym Tereshchenko, Alina Haidun, and Ostap Vykhopen. Llms as strategic actors: Behavioral alignment, risk calibration, and argumentation framing in geopolitical simulations. *arXiv preprint arXiv:2603.02128*, 2026. URL https://arxiv.org/abs/2603.02128.

Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pp. 1441–1450, 2019.

John Sweller. Cognitive load during problem solving: Effects on learning. *Cognitive science*, 12(2): 257–285, 1988.

Patrick Taillandier, Jean Daniel Zucker, Arnaud Grignard, Benoit Gaudou, Nghi Quang Huynh, and Alexis Drogoul. Integrating llm in agent-based social simulation: Opportunities and challenges. *arXiv preprint arXiv:2507.19364*, 2025. URL https://arxiv.org/abs/2507.19364.

Zichen Tang, Zirui Zhang, Qian Wang, Zhenheng Tang, Bo Li, and Xiaowen Chu. Is your llm-as-a-recommender agent trustable? llms' recommendation is easily hacked by biases (preferences). *arXiv preprint arXiv:2603.17417*, 2026.

Trung-Dung Vu, Benoit Gaudou, and Kamaldeep Singh Oberoi. Modeling realistic human behavior using generative agents in a multimodal transport system: Software architecture and application to toulouse. *arXiv preprint arXiv:2510.19497*, 2025. URL https://arxiv.org/abs/2510.19497.

Lei Wang, Jingsen Zhang, Hao Yang, Zhi-Yuan Chen, Jiakai Tang, Zeyu Zhang, Xu Chen, Yankai Lin, Hao Sun, Ruihua Song, Xin Zhao, Jun Xu, Zhicheng Dou, Jun Wang, and Ji-Rong Wen. User behavior simulation with large language model-based agents. *ACM Trans. Inf. Syst.*, 43 (2), January 2025a. ISSN 1046-8188. doi: 10.1145/3708985. URL https://doi.org/10.1145/3708985.

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

Yunzhe Wang, Gale Lucas, Burcin Becerik-Gerber, and Volkan Ustun. Implicit behavioral alignment of language agents in high-stakes crowd simulations. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 30669–30686, 2025c.

Xinye Wanyan, Danula Hettiachchi, Chenglong Ma, Ziqi Xu, and Jeffrey Chan. Temporal-aware user behaviour simulation with large language models for recommender systems. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, CIKM '25, pp. 5335–5339, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400720406. doi: 10.1145/3746252.3760878. URL https://doi.org/10.1145/3746252.3760878.

Tianjun Wei, Huizhong Guo, Yingpeng Du, Zhu Sun, Huang Chen, Dongxia Wang, and Jie Zhang. Mirroring users: Towards building preference-aligned user simulator with user feedback in recommendation. *arXiv preprint arXiv:2508.18142*, 2025.

Wei Zeng, Hengshu Zhu, Chuan Qin, Han Wu, Yihang Cheng, Sirui Zhang, Xiaowei Jin, Yuhuai Shen, Zifei Wang, Furong Zhong, and Hui Xiong. Multi-level value alignment in agentic ai systems: Survey and perspectives. *arXiv preprint arXiv:2506.09656*, 2025. URL https://arxiv.org/abs/2506.09656.

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

Chong Zhang, Xinyi Liu, Zhongmou Zhang, Mingyu Jin, Lingyao Li, Zhenting Wang, Wenyue Hua, Dong Shu, Suiyuan Zhu, Xiaobo Jin, et al. When ai meets finance (stockagent): Large language model-based stock trading in simulated real-world environments. *arXiv preprint arXiv:2407.18957*, 2024b.

Guilin Zhang, Kai Zhao, Jeffrey Friedman, and Xiaowen Chu. Robustexplain: Evaluating robustness of llm-based explanation agents for recommendation. *arXiv preprint arXiv:2601.19120*, 2026a. URL https://arxiv.org/abs/2601.19120.

Jiaqi Zhang, Yu Cheng, Yongxin Ni, Yunzhu Pan, Zheng Yuan, Junchen Fu, Youhua Li, Jie Wang, and Fajie Yuan. Ninerec: A benchmark dataset suite for evaluating transferable recommendation. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 47(07):5256–5267, 2025a.

Wenlin Zhang, Xiangyang Li, Qiyuan Ge, Kuicai Dong, Pengyue Jia, Xiaopeng Li, Zijian Zhang, Maolin Wang, Yichao Wang, Huifeng Guo, et al. Exploring recommender system evaluation: A multi-modal user agent framework for a/b testing. *arXiv preprint arXiv:2601.04554*, 2026b.

Zijian Zhang, Shuchang Liu, Ziru Liu, Rui Zhong, Qingpeng Cai, Xiangyu Zhao, Chunxu Zhang, Qidong Liu, and Peng Jiang. Llm-powered user simulator for recommender system. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 13339–13347, 2025b.

Xiaoyan Zhao, Yang Deng, Wenjie Wang, Hong Cheng, Rui Zhang, See-Kiong Ng, and Tat-Seng Chua. Exploring the impact of personality traits on conversational recommender systems: A simulation with large language models. *arXiv preprint arXiv:2504.12313*, 2025.

Content of Appendix

A Design of Psy-Engine: A Psychological State Engine for the PsyBer-Agent **18**
A.1 Initial Settings: Core State, Input Features, and Persona 18
A.2 Cross-domain feature construction. 18
A.3 Similarity, Projection, and Indicator Operators 19
A.4 Intermediate Drives 20
A.5 State Transition Equations 20

B Design of PsyBer-Agent Psycho-behavioral Alignment **20**
B.1 Model-Defined State Sequence Construction 21
B.2 Transport Initialization and UGW Alignment 21
B.3 Alignment-induced Soft Supervision for Policy Learning 23

C PsyBer Benchmark **24**
C.1 Benchmark Overview 24
C.2 Data Construction 24
C.3 Task Definition 24
C.4 Benchmark Coverage and Evaluation Protocol 25
C.5 WebSim Experiment Platform 25
C.6 Metrics and Evaluation 26

D Additional Details **29**
D.1 Details of the Evaluated Datasets 29
D.2 Details of the Evaluated Recommender Systems 30
D.3 Details of the User Simulator Baselines 30
D.4 Details of the Prompt Design 31
D.5 Details of Experimental Hardware Environment and Parallel Simulation Settings . . . 32
D.6 Details of Fixed-Backend Human–Simulator Agreement Protocol 33

E Additional Experiments & Analysis **33**
E.1 PsyBer-Agent Group Behavior Exploration. 33
E.2 Psy-Engine and PsyBer-Agent Parameter Sensitivity Analysis 33
E.3 PsyBer-Agent Robustness Stress Analysis 35
E.4 Representative Benchmark Performance of Recommendation Models on the Test Platform 37
E.5 Profile-Conditioned Group Fidelity Diagnostics 38
E.6 Offline Backbone Sanity Check 39
E.7 Aggregate Baseline Comparison 40
E.8 Fallback Mechanisms and Usage Statistics 40
E.9 Large-Scale Cost and Execution Analysis 40

F PsyBer-Agent Case Study **42**
F.1 MovieLens-based Web Browsing Scenario 42
F.2 Amazon Magazine Subscriptions Shopping Scenario 43

G Declarations **44**
G.1 Limitations 44
G.2 Broader impacts 45

APPENDIX A DESIGN OF PSY-ENGINE: A PSYCHOLOGICAL STATE ENGINE FOR THE PSYBER-AGENT

A.1 INITIAL SETTINGS: CORE STATE, INPUT FEATURES, AND PERSONA

In this part, we pre-process the information observable to the agent to reduce the burden of online computation. At interaction step t, Psy-Engine maintains a three-dimensional state

$$\mathbf{z}_t = [C_t, E_t, S_t]^\top \in [0,1]^3, \tag{12}$$

where C_t, E_t, and S_t denote cognitive reserve, emotional satisfaction, and self-congruence at time t, respectively. Each candidate item i_t is represented by

$$\phi(i_t) = (\mathbf{v}_t, n_t, c_t, p_t), \tag{13}$$

where $\mathbf{v}_t \in \mathbb{R}^d$ is the item embedding, $n_t \in [0,1]$ measures item-level incongruence or negative-preference strength, $c_t \in [0,1]$ denotes cognitive load, and $p_t \in [0,1]$ is the social-proof score.

A.2 CROSS-DOMAIN FEATURE CONSTRUCTION.

For cross-domain recommendation scenarios, we map heterogeneous item metadata into a unified psycho-behavioral feature tuple. Textual fields and visual assets are encoded into normalized embeddings and fused to obtain v_t. The incongruence feature n_t measures the mismatch between the item and the user's current preference or persona anchor, and may incorporate text-image inconsistency as an item-level prior. The cognitive-load feature c_t is estimated from content complexity, including text length, metadata density, and visual complexity. The social-proof feature p_t is computed from observable signals such as ratings, review counts, click/like rates, or platform hotness indicators.

All signals are computed from pre-window item metadata and normalized over the global item set. Social-proof features use only ratings, review counts, popularity, or logged click/like statistics available before the evaluated interaction. All scalar features are normalized to $[0,1]$ over the global item set, ensuring that each item retains stable psycho-behavioral values across interaction steps.

The user persona contains three anchors, $\mathbf{u}_{\text{base}},\ \mathbf{u}_{\text{ideal}},\ \mathbf{u}_{\text{real}} \in \mathbb{R}^d$, and we denote the persona by

$$\psi = (\mathbf{u}_{\text{base}}, \mathbf{u}_{\text{ideal}}, \mathbf{u}_{\text{real}}). \tag{14}$$

Here, $\mathbf{u}_{\text{base}}$ summarizes the user's static profile and metadata, $\mathbf{u}_{\text{ideal}}$ captures long-term preferences from historical interactions, and $\mathbf{u}_{\text{real}}$ captures recent preferences from the latest valid interactions. For completeness, the three anchors are computed as follows. Let $\mathbf{e}(x)$ denote the encoder for a profile field, metadata field, or item, and let $\mathbf{P}_{\text{base}}$, $\mathbf{P}_{\text{ideal}}$, and $\mathbf{P}_{\text{real}}$ project the corresponding representations into a shared d-dimensional persona space.

If $m_{u,j}^{\text{base}} \in \{0,1\}$ indicates availability of field j, then

$$\mathbf{u}_{\text{base}} = \mathbf{P}_{\text{base}} \left(\frac{\sum_{j=1}^{J} m_{u,j}^{\text{base}} \mathbf{e}(x_{u,j})}{\max\left(1, \sum_{j=1}^{J} m_{u,j}^{\text{base}}\right)} \right). \tag{15}$$

Let $\mathcal{H}_u^{\text{long}}$ denote the valid pre-window historical interactions and let $\mathbf{v}_i$ be the representation of item i. With interaction mask $m_{u,i}$, the long-term anchor is

$$\mathbf{u}_{\text{ideal}} = \mathbf{P}_{\text{ideal}} \left(\frac{\sum_{i \in \mathcal{H}_u^{\text{long}}} m_{u,i} \mathbf{v}_i}{\max\left(1, \sum_{i \in \mathcal{H}_u^{\text{long}}} m_{u,i}\right)} \right). \tag{16}$$

Let $\mathcal{H}_u^K = (i_{u,1}, \ldots, i_{u,K})$ be the most recent K valid interactions in temporal order, with mask $m_{u,k}^K$. The recent-interaction anchor is

$$\mathbf{u}_{\text{real}} = \mathbf{P}_{\text{real}} \left(\frac{\sum_{k=1}^{K} m_{u,k}^K \mathbf{v}_{i_{u,k}}}{\max\left(1, \sum_{k=1}^{K} m_{u,k}^K\right)} \right). \tag{17}$$

If no valid field or interaction is available, the corresponding numerator is replaced by a fixed cold-start vector and the availability mask is set to zero. The anchors are used as conditioning representations and are not interpreted as direct measurements of psychological traits.The engine also maintains a bounded memory sequence

$$\mathcal{M}_t = [\mathbf{m}_1, \ldots, \mathbf{m}_{|\mathcal{M}_t|}], \qquad |\mathcal{M}_t| \le M_{\max}, \tag{18}$$

where $M_{\max}$ is the memory capacity and each memory entry $\mathbf{m}$ is a normalized embedding of a previously consumed item. When the memory reaches $M_{\max}$, newly consumed items are inserted using a first-in-first-out update rule. We further denote by $\mathcal{M}_t^{(h)}$ the subsequence consisting of the most recent $\min(h, |\mathcal{M}_t|)$ entries of $\mathcal{M}_t$, where $h \in \mathbb{N}_+$ is the short-term memory horizon. The engine uses a global update rate $\alpha \in [0, 1]$ to control the sensitivity of state transitions.

For each item i_t, we pre-compute item-level features $(\mathbf{v}_t, c_t, p_t)$ during offline preprocessing, while n_t is computed with respect to the current user's preference anchor. The item embedding $\mathbf{v}_t$ is obtained by fusing the item's textual and visual features, and is normalized as

$$\hat{\mathbf{v}}_t = \frac{\mathbf{v}_t}{\max(\|\mathbf{v}_t\|_2, \epsilon)}, \tag{19}$$

where $\epsilon > 0$ is a small smoothing constant for numerical stability. The incongruence score n_t is computed as the semantic distance between the item and the user's current preference anchor:

$$n_t = 1 - \max(0, \cos(\hat{\mathbf{v}}_t, \mathbf{u}_t^{\text{pref}})), \tag{20}$$

where $\mathbf{u}_t^{\text{pref}}$ denotes the current preference anchor, instantiated by $\mathbf{u}_{\text{base}}$ unless otherwise specified.

The cognitive-load score c_t is estimated from the intrinsic complexity of item content and normalized to $[0, 1]$:

$$c_t = \text{Norm}_{\mathcal{I}}\big(\text{Comp}(i_t)\big), \tag{21}$$

where $\text{Comp}(i_t)$ denotes the raw content-complexity score, computed from signals such as text length, metadata density, and visual complexity. The social-proof score p_t is computed from observable social feedback and is also normalized to $[0, 1]$:

$$p_t = \text{Norm}_{\mathcal{I}}\big(\text{Social}(i_t)\big), \tag{22}$$

where $\text{Social}(i_t)$ is instantiated by aggregate feedback signals such as item rating, popularity, review count, or, in WebSim, other agents' like and click rates. Here, $\text{Norm}_{\mathcal{I}}(\cdot)$ denotes min–max normalization over the global item set $\mathcal{I}$, rather than over the current candidate set, so that the same item has a stable normalized score across different interaction steps.

A.3 Similarity, Projection, and Indicator Operators

We use cosine similarity

$$\cos(\mathbf{a}, \mathbf{b}) = \begin{cases} \dfrac{\mathbf{a}^\top \mathbf{b}}{\|\mathbf{a}\|_2 \|\mathbf{b}\|_2}, & \|\mathbf{a}\|_2 \|\mathbf{b}\|_2 > 0, \\ 0, & \text{otherwise}, \end{cases} \tag{23}$$

and the projection operator

$$\text{Proj}_{[0,1]}(x) = \min(1, \max(0, x)). \tag{24}$$

We also use the indicator function

$$\mathbf{1}[\mathcal{E}] = \begin{cases} 1, & \text{if the event or condition } \mathcal{E} \text{ is true}, \\ 0, & \text{otherwise}. \end{cases} \tag{25}$$

For example, $\mathbf{1}[i_r = y]$ equals 1 when the r-th expert item is y and 0 otherwise. It is used to aggregate the alignment weights associated with the same item identity.

A.4 Intermediate Drives

Given item i_t, novelty ν_t and semantic saturation σ_t are defined as

$$\nu_t = \begin{cases} \mathrm{Proj}_{[0,1]}(1 - \max_{\mathbf{m}\in\mathcal{M}_t} \cos(\hat{\mathbf{v}}_t, \mathbf{m})), & |\mathcal{M}_t| > 0, \\ 1, & |\mathcal{M}_t| = 0, \end{cases} \tag{26}$$

and

$$\sigma_t = \begin{cases} \mathrm{Proj}_{[0,1]}\left(\dfrac{1}{|\mathcal{M}_t^{(h)}|} \sum_{\mathbf{m}\in\mathcal{M}_t^{(h)}} \max\big(0, \cos(\hat{\mathbf{v}}_t, \mathbf{m})\big) \right), & |\mathcal{M}_t^{(h)}| > 0, \\ 0, & |\mathcal{M}_t^{(h)}| = 0. \end{cases} \tag{27}$$

The resulting drive factors are

$$A_t = \frac{\nu_t + n_t}{2}, \qquad F_t = \frac{c_t + \sigma_t}{2}, \tag{28}$$

These factors provide a compact decomposition of the psychological drives underlying user behavior.

In equation 28, A_t and F_t denote attentional attractiveness and cognitive fatigue, whose difference forms the net cognitive flux. The design of A_t follows the intuition that novelty and preference inconsistency stimulate exploratory interest Loewenstein (1994); Silvia (2008), while F_t captures the dual depletion caused by intrinsic cognitive load and repeated exposure to semantically similar items Sweller (1988); Redden (2008).

We further compute the semantic-fit and serendipitous-interest signals as $\beta_t = \max(0, \cos(\hat{\mathbf{v}}_t, \mathbf{u}_{\text{base}}))$ and $\rho_t = \nu_t \beta_t$, respectively. The affective target is then given by $E_t^\star = (\beta_t + \rho_t + p_t)/3$, combining semantic fit, novelty-driven serendipity, and social proof Kotkov et al. (2016); Cialdini & Goldstein (2004). Finally, the self-conflict signal is defined as $\gamma_t = \mathrm{Proj}_{[0,1]}(\max(0, \cos(\hat{\mathbf{v}}_t, \mathbf{u}_{\text{real}}) - \cos(\hat{\mathbf{v}}_t, \mathbf{u}_{\text{ideal}})))$, and the self-alignment target is $S_t^\star = \mathrm{Proj}_{[0,1]}(\beta_t - \gamma_t)$, following the rationale that self-fit enhances self-congruence while self-discrepancy or guilt weakens it Higgins (1987); Sirgy (1982). Together, these quantities replace multi-term hand-crafted penalties with a small set of interpretable drives that remain stable across scenarios.

A.5 State Transition Equations

The three state variables are updated from the current state $\mathbf{z}_t$ using the projection operator $\mathrm{Proj}_{[0,1]}$, consistent with Eq. (4) in the main text:

$$C_{t+1} = \mathrm{Proj}_{[0,1]}(C_t + \alpha(A_t - F_t)), \tag{29}$$

$$E_{t+1} = \mathrm{Proj}_{[0,1]}((1-\alpha)E_t + \alpha E_t^\star), \tag{30}$$

$$S_{t+1} = \mathrm{Proj}_{[0,1]}((1-\alpha)S_t + \alpha S_t^\star). \tag{31}$$

The cognitive state responds to the net cognitive flux $A_t - F_t$, while emotional satisfaction and self-congruence move toward their interaction-dependent targets $E_t^\star$ and $S_t^\star$. This transition is a model-defined computational hypothesis; it does not imply independent psychological validity. The fixed Psy-Engine parameters are collectively denoted as Θ, including α, h, $M_{\max}$, the feature weights used in the drive functions, and the fixed feature-normalization statistics.

Appendix B Design of PsyBer-Agent Psycho-behavioral Alignment

In PsyBer-Agent psycho-behavioral alignment, we describe how PsyBer-Agent aligns offline expert behavior with the simulator-side psychological state sequence. For clarity, we omit the user subscript and consider one expert sequence and one simulator sequence at a time.

Let the expert sequence be

$$\tau^{\text{exp}} = (i_1, \ldots, i_n), \tag{32}$$

where i_r denotes the clicked item at expert position r. Let the simulator-side psychological sequence be

$$\mathcal{Z}^{\mathrm{sim}} = (\mathbf{z}_1, \ldots, \mathbf{z}_m), \tag{33}$$

where

$$\mathbf{z}_j = [C_j, E_j, S_j]^\top \in [0,1]^3 \tag{34}$$

is the three-dimensional psychological state defined by Psy-Engine, with C_j, E_j, and S_j denoting cognition, emotional satisfaction, and self-congruence, respectively.

B.1 Model-Defined State Sequence Construction

The offline log records expert item clicks but does not contain annotations of the latent states underlying those decisions. Because the observed clicks alone cannot identify a unique psychological trajectory, we use them as externally supplied actions and reconstruct a model-defined reference trajectory under the same persona-conditioned Psy-Engine used by the simulator. This reconstruction is used for trajectory alignment and policy supervision, rather than treated as a direct measurement of human psychology.

Starting from the persona-conditioned initial state $\tilde{\mathbf{z}}_0$ and replay memory $\tilde{\mathcal{M}}_0$, we replay the expert sequence $\tau^{\mathrm{exp}} = (i_1, \ldots, i_n)$ using teacher forcing, where each logged item is supplied as the consumed item at its corresponding step. Consistent with the transition operator in the main text, we recursively compute

$$\tilde{\mathbf{z}}_r = \mathcal{T}_\Theta\Big(\tilde{\mathbf{z}}_{r-1}, \phi(i_r), \tilde{\mathcal{M}}_{r-1}\Big), \qquad r = 1, \ldots, n. \tag{35}$$

where $\mathcal{T}_\Theta$ denotes the item-level Psy-Engine transition, $\phi(i_r) = (\mathbf{v}_r, n_r, c_r, p_r)$ is the psycho-behavioral feature tuple of item i_r, and Θ collects the fixed Psy-Engine parameters. The term $\tilde{\mathcal{M}}_{r-1}$ denotes the replay memory before step r, storing normalized embeddings of previously replayed expert items. Thus, $\tilde{\mathbf{z}}_r$ is the model-defined latent reference state associated with the r-th logged interaction; it is not an independently observed psychological label.

After replaying i_r, the memory is updated by appending its normalized embedding and truncating the sequence to the most recent $M_{\max}$ entries. This update preserves the same bounded-memory convention used during online simulation, so that the reconstructed expert-side dynamics and simulator-side dynamics are evaluated under comparable state-transition conditions. This yields the model-defined reference sequence

$$\tilde{\mathbf{Z}}^{\mathrm{exp}} = (\tilde{\mathbf{z}}_1, \ldots, \tilde{\mathbf{z}}_n), \qquad \tilde{\mathbf{z}}_r \in [0,1]^3. \tag{36}$$

The resulting sequence is subsequently aligned with the simulator trajectory to transfer information from observed expert actions into state-conditioned soft supervision. In this process, the expert clicks determine the replay inputs, while the Psy-Engine determines the corresponding latent reference states.

For any state sequence $\mathbf{x}_{1:T} = (\mathbf{x}_1, \ldots, \mathbf{x}_T)$, we define its backward increments as

$$\Delta\mathbf{x}_k = \begin{cases} \mathbf{x}_k - \mathbf{x}_{k-1}, & 2 \le k \le T, \\ \mathbf{0}, & k = 1. \end{cases} \tag{37}$$

This convention is applied to the simulator, reconstructed expert-reference, and regressed expert pseudo-state sequences.

B.2 Transport Initialization and UGW Alignment

A direct uniform initialization of the transport matrix is too weak, because the expert item sequence and the simulator psychological sequence are heterogeneous and may have different lengths. To improve alignment stability, we discard sequences shorter than a minimum length threshold and truncate overly long sequences to a fixed maximum horizon during preprocessing.

We therefore first learn an offline regression prior that maps expert-side prefixes into the same three-dimensional state space as the simulator. Specifically, we train a regression function g_φ to predict the model-defined state prior:

$$\hat{\mathbf{z}}_r = g_\varphi\Big(\psi,\, \tau^{\mathrm{exp}}_{1:r},\, \phi(i_r),\, \frac{r}{n}\Big) \tag{38}$$

where ψ denotes the user persona, $\tau_{1:r}^{\exp} = (i_1, \ldots, i_r)$ is the expert prefix up to position r, φ denotes the regression parameters, r/n represents the relative position of the current interaction r within the full log sequence of length n, and $\hat{\mathbf{z}}_r$ is the regressed pseudo state aligned to the simulator state space. Let $\mathcal{D}_{\exp}$ denote the offline collection of expert browsing sequences. Here, $\tilde{\mathbf{z}}_r$ denotes the model-defined reference state reconstructed by replaying the expert sequence through Psy-Engine. The regression parameters are learned offline by minimizing

$$\varphi^\star = \arg\min_{\varphi} \sum_{\tau^{\exp} \in \mathcal{D}_{\exp}} \sum_{r=1}^{n} \Big\| g_\varphi\Big(\psi, \tau_{1:r}^{\exp}, \phi(i_r), \frac{r}{n}\Big) - \tilde{\mathbf{z}}_r \Big\|_2^2 . \tag{39}$$

Here, $\varphi^\star$ denotes the parameter value that minimizes the regression objective; the superscript $\star$ marks the solution obtained by optimization. Using the increment rule in Eq. (37), we define the pseudo-state increments $\Delta\hat{\mathbf{z}}_r$ accordingly.

Given the simulator state sequence $\mathcal{Z}^{\mathrm{sim}} = (\mathbf{z}_1, \ldots, \mathbf{z}_m)$ and the regressed expert pseudo-state sequence $(\hat{\mathbf{z}}_1, \ldots, \hat{\mathbf{z}}_n)$, we define the initialization cost and use it to induce the initial transport weights as follows:

$$c_{j,r}^{\mathrm{init}} = \lambda_z \|\mathbf{z}_j - \hat{\mathbf{z}}_r\|_2^2 + \lambda_\Delta \|\Delta\mathbf{z}_j - \Delta\hat{\mathbf{z}}_r\|_2^2 \qquad \Pi_{j,r}^{(0)} \propto \exp\Big(-\tfrac{c_{j,r}^{\mathrm{init}}}{\tau_0}\Big), \tag{40}$$

Here, $c_{j,r}^{\mathrm{init}}$ is the prior matching cost, $\lambda_z \geq 0$ and $\lambda_\Delta \geq 0$ weight the state and local-change discrepancies, respectively, and $\tau_0 > 0$ controls the sharpness of the soft matching distribution. We then rescale $\Pi^{(0)}$ using the prescribed marginal distributions $a \in \mathbb{R}_+^m$ and $b \in \mathbb{R}_+^n$; unless otherwise specified, both marginals are uniform.

After initialization, we optimize Π by Unbalanced Gromov-Wasserstein (UGW) alignment. On the expert side, let $\mathbf{v}_r$ denote the fused semantic embedding contained in $\phi(i_r)$. We define the expert-side internal distance $d_{\exp}$ as

$$d_{\exp}(r, r') = \lambda_v\big(1 - \cos(\mathbf{v}_r, \mathbf{v}_{r'})\big), \tag{41}$$

where $\lambda_v \geq 0$ controls semantic discrepancy. On the simulator side, the internal distance between two psychological states is defined as

$$d_{\mathrm{sim}}(j, j') = \lambda_s \|\mathbf{z}_j - \mathbf{z}_{j'}\|_2 + \lambda_\Delta^{\mathrm{sim}} \|\Delta\mathbf{z}_j - \Delta\mathbf{z}_{j'}\|_2, \tag{42}$$

where $\lambda_s \geq 0$ controls state discrepancy and $\lambda_\Delta^{\mathrm{sim}} \geq 0$ controls increment discrepancy. We assume a positive structural correspondence between the two spaces: simulator-state pairs that are farther apart should, on average, correspond to expert-item pairs that are more semantically dissimilar. Accordingly, we penalize the difference between their weighted distances. The structural discrepancy between relations (j, j') and (r, r') is

$$\ell\big((j, j'), (r, r')\big) = \big(d_{\mathrm{sim}}(j, j') - d_{\exp}(r, r')\big)^2 . \tag{43}$$

Accordingly, the UGW objective is

$$\mathcal{L}_{\mathrm{UGW}}(\Pi) = \sum_{j,j',r,r'} \ell\big((j, j'), (r, r')\big)\Pi_{j,r}\Pi_{j',r'} - \varepsilon H(\Pi) + \tau_{\mathrm{ugw}}\mathrm{KL}(\Pi\mathbf{1}_n \,\|\, a) + \tau_{\mathrm{ugw}}\mathrm{KL}(\Pi^\top\mathbf{1}_m \,\|\, b), \tag{44}$$

Here, $\Pi_{j,r}$ and $\Pi_{j',r'}$ denote the transport masses assigned to the two simulator–expert correspondences (j, r) and (j', r'), respectively. Their product $\Pi_{j,r}\Pi_{j',r'}$, an entry of $\Pi \otimes \Pi$, weights the structural comparison between these correspondences. where

$$H(\Pi) = -\sum_{j,r} \Pi_{j,r}\big(\log \Pi_{j,r} - 1\big) \tag{45}$$

is the entropy regularizer, and $\varepsilon > 0$ is its weight. With this sign convention, the term $-\varepsilon H(\Pi)$ encourages a smoother transport plan during optimization.

The coefficient $\tau_{\mathrm{ugw}} > 0$ controls marginal relaxation, $\mathbf{1}_n$ and $\mathbf{1}_m$ denote all-ones vectors of dimensions n and m, respectively, and $\mathrm{KL}(\cdot\|\cdot)$ denotes the Kullback–Leibler divergence. The term $\tau_{\mathrm{ugw}}\mathrm{KL}(\Pi\mathbf{1}_n \,\|\, a)$ penalizes deviations of the simulator-side marginal from a, while

$\tau_{\mathrm{ugw}}\mathrm{KL}(\Pi^\top \mathbf{1}_m \,\|\, b)$ penalizes deviations of the expert-side marginal from b. Starting from the initialization $\Pi^{(0)}$, we obtain the final transport matrix by solving

$$\Pi^\star = \arg\min_{\Pi \geq 0} \mathcal{L}_{\mathrm{UGW}}(\Pi). \tag{46}$$

Because the optimization is unbalanced, the learned transport plan naturally tolerates different sequence lengths, browsing speeds, and locally unmatched segments.

B.3 Alignment-induced Soft Supervision for Policy Learning

The offline expert logs contain clicked items but do not specify which action should be taken on the simulator's current candidate slate. After reconstructing the model-defined expert trajectory and aligning it with the simulator trajectory in Appendix B.2, the optimized transport matrix $\Pi^\star$ provides a step-wise correspondence between the two trajectories. We use this correspondence to construct state-conditioned soft action supervision for the fast policy.

The optimized transport matrix $\Pi^\star$ specifies which expert positions are most compatible with each simulator step. We first convert it into normalized matching weights:

$$q_j(r) = \frac{\Pi^\star_{j,r}}{\sum_{r'=1}^{n} \Pi^\star_{j,r'} + \delta}, \tag{47}$$

where $\delta > 0$ is a small constant for numerical stability. Based on $q_j(r)$, we define the item-level soft target distribution

$$\hat{q}_j(y) = \sum_{r=1}^{n} q_j(r)\mathbf{1}[i_r = y]. \tag{48}$$

Here, $\hat{q}_j(y)$ represents the probability mass assigned to item identity y under the alignment-induced supervision at simulator step j. This distribution reflects which expert items are most behaviorally compatible with the current simulator psychological state $\mathbf{z}_j$.

At simulator step j, let the current candidate slate be $\mathcal{L}_j = (\ell_{j,1}, \ldots, \ell_{j,K_s})$, where $\ell_{j,k}$ is the k-th candidate item and K_s is the slate size. The corresponding action set is $\mathcal{A}_j = \{\ell_{j,1}, \ldots, \ell_{j,K_s}, \mathrm{skip}\}$.

Here, skip denotes the no-click or continue-browsing action, implemented as page continuation in WebSim. Since an expert item may be absent from the current candidate slate, we project the item-level target onto $\mathcal{A}_j$:

$$\hat{p}_j(a) = \begin{cases} \hat{q}_j(\ell_{j,k}), & a = \ell_{j,k}, \quad k = 1, \ldots, K_s, \\ 1 - \displaystyle\sum_{k=1}^{K_s} \hat{q}_j(\ell_{j,k}), & a = \mathrm{skip}. \end{cases} \tag{49}$$

The residual mass assigned to skip accounts for aligned expert items that are not available in the current slate, as well as unmatched trajectory mass. Therefore, the projection converts expert-side alignment information into a valid probability distribution over the actions currently executable by the simulator.

The resulting $\hat{p}_j$ provides dense, state-conditioned supervision even though the offline logs contain only sparse click observations. We use it to define the imitation objective for the online fast policy:

$$\mathcal{L}_{\mathrm{imit}} = \sum_j \mathrm{CE}\big(\hat{p}_j, \pi_\theta(\cdot \mid \mathbf{z}_j^{\mathrm{sim}}, \mathcal{A}_j)\big), \tag{50}$$

where $\pi_\theta(\cdot \mid \mathbf{z}_j^{\mathrm{sim}}, \mathcal{A}_j)$ is the policy distribution over the current simulator state and executable action set, parameterized by θ, and $\mathrm{CE}(\cdot, \cdot)$ denotes cross-entropy. This objective transfers psycho-behaviorally aligned supervision into the lightweight online policy, rather than relying on sparse click replay alone.

APPENDIX C PSYBER BENCHMARK

C.1 BENCHMARK OVERVIEW

PsyBer Benchmark is a closed-loop benchmark for simulated-user evaluation in recommender systems. It is designed to evaluate whether a user simulator can (i) produce realistic and scenario-consistent interaction trajectories under online recommendation exposure, and (ii) support reliable evaluation of target recommendation models in controlled simulation environments. To this end, the benchmark standardizes four components: datasets, recommendation scenarios, target recommender backbones, and evaluation protocols. All experiments are conducted on the WebSim platform, which decouples user-interaction serving from recommender inference and enables consistent cross-model and cross-dataset comparisons. Detailed platform implementation and metric definitions are provided in Appendices C.5 and C.6, respectively.

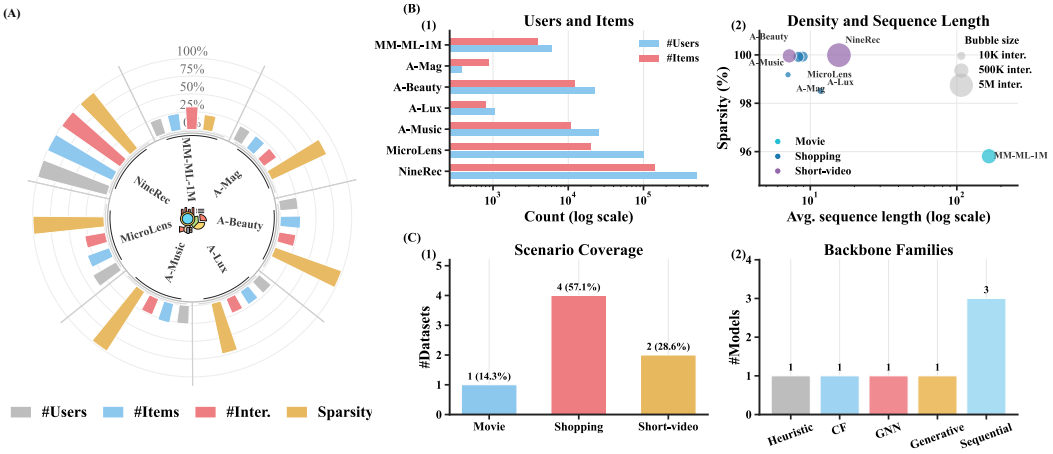


Figure 3: Data Statistics for the PsyBer Benchmark. **(A.)** Dataset footprint; **(B.)** scale and density; **(C.)** scenario and backbone coverage.

C.2 DATA CONSTRUCTION

We construct the benchmark from seven public datasets spanning movie, e-commerce, short-video, and cross-domain recommendation. Specifically, we include **MM-ML-1M** Zhang et al. (2026b), a multimodal extension of MovieLens-1M, to evaluate recommendation settings with rich item-side modalities. To test cross-domain robustness in shopping scenarios, we further adopt four re-edited subsets from the 2018 Amazon Reviews collection Ni et al. (2019): **Musical_Instruments**, **Magazine_Subscriptions**, **Beauty**, and **Luxury_Beauty**. In addition, we include **MicroLens-100K** Ni et al. (2025), a short-video recommendation dataset, and **NineRec-Bili500K** Zhang et al. (2025a), a large-scale cross-domain recommendation dataset with multimodal item metadata.

To ensure fair comparisons across domains, all datasets are processed under a unified protocol for model training and evaluation, including standardized interaction formatting, scenario-specific item metadata construction, and chronological splitting for training, validation/test evaluation. This unified preprocessing allows the same simulator and recommender evaluation pipeline to be applied across heterogeneous recommendation settings.

C.3 TASK DEFINITION

The PsyBer Benchmark consists of two core tasks. **(1)** Multi-scenario user online simulation. **(2)** Recommendation model evaluation.

Task 1: Multi-scenario user online simulation. This task studies whether a simulator can produce realistic user behavior under closed-loop recommendation exposure. Given a recommendation environment, PsyBer-Agent sequentially interacts with the served slates and generates user feedback, including clicks, skips, and continued browsing. The resulting trajectories are used to evaluate

whether the simulator preserves realistic interaction patterns, stable long-horizon behavior, and scenario-consistent responses across different recommendation domains.

Task 2: Recommendation model evaluation. This task evaluates target recommendation models using PsyBer-Agent as a simulated user proxy. On the WebSim platform, under a fixed scenario setting, PsyBer-Agent interacts with each target recommender and generates closed-loop trajectories, which are then used to assess recommendation quality and system behavior. We report online recommendation metrics such as CTR and NDCG@K, along with calibration and efficiency metrics such as CTRGap and latency per decision, where K is specified by the evaluation setting for each experiment.

C.4 Benchmark Coverage and Evaluation Protocol

To support broad and reproducible evaluation, the PsyBer Benchmark covers multiple recommendation backbones and application scenarios. The target recommender set includes representative methods from popularity-based, matrix-factorization-based, graph-based, variational, sequential, and transformer-based recommendation, including PopRec Cremonesi et al. (2010), BPR-MF Rendle et al. (2012), LightGCN He et al. (2020), Multi-VAE Liang et al. (2018), GRU4Rec Hidasi et al. (2015), SASRec Kang & McAuley (2018), and BERT4Rec Sun et al. (2019). The covered recommendation scenarios include movie viewing, shopping, image-text information flow, and short-video browsing. Dataset statistics and coverage of application scenarios and recommender backbones are summarized in Figure 3.

All benchmark evaluations follow a unified closed-loop protocol. For each dataset and recommender setting, WebSim serves recommendation slates to the simulator, collects the simulator's feedback, and updates the interaction history stepwise. This design allows benchmark results to reflect both the quality of local actions and long-horizon interaction dynamics. At a high level, the benchmark evaluates PsyBer-Agent from four complementary perspectives: behavior fidelity, baseline simulator comparison, scalability and cost, and group behavior. Detailed metric definitions and evaluation procedures are provided in Appendix C.6.

C.5 WebSim Experiment Platform

Design goals. We design WebSim as a controllable and reproducible platform for interactive recommendation simulation. In its current version, WebSim adopts a grid-based interaction protocol and separates three concerns: interaction serving, model-agnostic recommendation inference, and dataset-aware metadata and poster delivery. This modularization allows datasets and recommender backbones to vary without modifying the core serving pipeline.

Interaction state management. WebSim maintains a lightweight per-session state in Flask sessions together with an auxiliary in-memory table for interaction statistics. The session state stores recent interaction history, cached recommendation IDs, the current page index, the selected model, the dataset identifier, and a simulation identifier, while the statistics table records item-level impressions and clicks. At initialization, the service resets the session and returns four seed-controlled sampled items. Subsequent clicks update the interaction history, refresh the recommendation cache, and return the next batch of candidates, while pagination traverses the cached list deterministically. In the current implementation, the history length is capped at 20, the page size is fixed to 4, and the recommendation cache is constructed with top-$K = 200$.

Recommendation layer and model abstraction. A unified `MovieRecommender` wrapper dispatches inference to up to seven backbones: PopRec, BPR-MF, GRU4Rec, BERT4Rec, SASRec, LightGCN, and Multi-VAE. For each dataset, only backbones with available checkpoints are instantiated. All models share a normalized item-ID interface and return ranked item IDs; the wrapper then filters unavailable IDs, removes duplicates, and maps the remaining candidates to display cards through `MovieCatalog`. If a requested model is unavailable, the system falls back to a default model; random candidate sampling is used only when no model can be loaded or when inference returns no valid items.

Data and asset handling. `MovieCatalog` unifies metadata loading for MovieLens-style and Amazon-style datasets in the current implementation and resolves image resources from dataset-local posters and images, as well as legacy fallback paths. When recommendation models are available, the catalog is restricted to items covered by the loaded artifacts, preventing dangling item IDs during serving. If a poster file is missing, the platform generates an SVG placeholder on the fly, ensuring stable rendering across datasets with heterogeneous media completeness.

Reproducibility and deployment. WebSim supports script-based launch and shutdown, service readiness probing, and environment-configurable dataset and checkpoint paths, enabling reproducible local and batch evaluation. The service port is configurable, and the current Flask implementation binds to `127.0.0.1` by default. Because dataset paths and model artifacts are configured outside the serving logic, the same codebase can be reused across domains and model variants with minimal engineering overhead.

System architecture

WebSim adopts a lightweight client–server architecture with a browser-based client that supports click and pagination feedback on a grid page. The UML class diagram is shown in Figure 4. The backend is implemented in Flask and provides a compact serving layer for session initialization, item selection, recommendation pagination, session cleanup, poster rendering, and health monitoring. By centralizing session management and recommendation dispatch, the platform keeps the frontend lightweight while simplifying deployment and experiment orchestration.

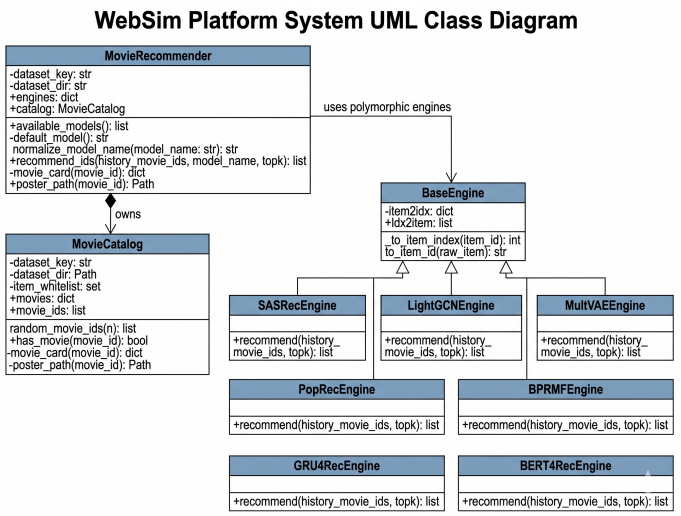


Figure 4: UML class diagram of the WebSim platform.

C.6 Metrics and Evaluation

We use common notation for the metrics reported across scenarios. Let $\mathcal{U}_{\text{eval}}$ denote the evaluated users, $\mathcal{I}$ the item set, and K the ranking cutoff used for top-K evaluation. For each user u, $\mathcal{R}_{u,K}$ denotes the top-K recommended list and $\mathcal{G}_u$ denotes the held-out relevant item set. Let P_{sim} and P_{exp} be the simulated and expert-reference item interaction distributions, respectively. For matched simulator–expert sequence pairs, let $\mathcal{M}$ denote the matching set. For each matched pair $(u, u') \in \mathcal{M}$, let $T_{u,u'}$ denote the matched sequence length, and let $a_{u,t}^{\text{sim}}$ and $a_{u',t}^{\text{exp}}$ denote the simulated and expert-reference actions at step t, respectively. We denote the model-defined C/E/S latent state at step t as

$$\mathbf{z}_{u,t} = [C_{u,t}, E_{u,t}, S_{u,t}]^\top \in [0,1]^3.$$

When distinguishing simulated and expert-reference trajectories, we use $\mathbf{z}_{u,t}^{\text{sim}}$ and $\mathbf{z}_{u',t}^{\text{exp}}$, respectively, where $\mathbf{z}_{u',t}^{\text{exp}}$ is reconstructed from offline interaction logs through Psy-Engine teacher forcing.

Behavior Fidelity. We measure whether simulated agents reproduce expert-reference behavior at both population and trajectory levels.

- **Interaction Jensen–Shannon Divergence (Interaction JSD)** (↓): population-level mismatch between simulated and expert-reference item interaction distributions,

$$\text{JSD} = \frac{1}{2}\text{KL}(P_{\text{sim}} \| P_{\text{mix}}) + \frac{1}{2}\text{KL}(P_{\text{exp}} \| P_{\text{mix}}), \quad P_{\text{mix}} = \frac{1}{2}(P_{\text{sim}} + P_{\text{exp}}).$$

Lower Interaction JSD indicates that the simulated population better matches the expert-reference population-level interaction patterns.

- **Earth Mover's Distance (EMD)** (↓): distributional distance between simulated and expert-reference downstream behavior trajectories after matched prefixes, where trajectories are represented by future interaction sequences or their embedded representations. Lower EMD indicates that simulated agents produce future behavioral trajectories closer to those observed from expert-reference logs.

Baseline Simulator Comparison. We compare PsyBer-Agent against baseline user simulators under a controlled fixed-candidate replay setting. At each step, all simulators observe the same user histories, item metadata, and candidate pages, and are evaluated on decision accuracy, CTR calibration, distributional fidelity, item-level correlation with real logs, and latency.

- **Decision Accuracy (Acc.)** (↑): step-wise action agreement between matched simulator and expert-reference trajectories,

$$\mathrm{Acc} = \frac{1}{|\mathcal{M}|} \sum_{(u,u') \in \mathcal{M}} \frac{1}{T_{u,u'}} \sum_{t=1}^{T_{u,u'}} \mathbb{I}\Big(a_{u,t}^{\mathrm{sim}} = a_{u',t}^{\mathrm{exp}}\Big) .$$

- **CTRGap** (↓):absolute CTR deviation for the evaluated dataset–recommender cell,

$$\mathrm{CTRGap} = \left|\mathrm{CTR}_{\mathrm{sim}} - \mathrm{CTR}_{\mathrm{exp}}\right| .$$

- **Pearson Correlation (R)** (↑): item-level correlation between simulated and expert-reference interaction frequencies,

$$R = \mathrm{Corr}\Big(\big(f_{\mathrm{sim}}(i)\big)_{i \in \mathcal{I}}, \big(f_{\mathrm{exp}}(i)\big)_{i \in \mathcal{I}}\Big) ,$$

  where $f_{\mathrm{sim}}(i)$ and $f_{\mathrm{exp}}(i)$ denote the simulated and expert-reference interaction frequencies of item i, respectively.

- **Latency/Decision** (↓): average wall-clock latency per simulator decision.

Recommendation Model Evaluation. For the benchmark task of evaluating target recommendation models with PsyBer-Agent, we report standard online recommendation metrics derived from simulated interaction trajectories.

- **Hit Rate@10 (HR@10)** (↑):

$$\mathrm{HR@10} = \frac{1}{|\mathcal{U}_{\mathrm{eval}}|} \sum_{u \in \mathcal{U}_{\mathrm{eval}}} \mathbb{I}(\mathcal{R}_{u,10} \cap \mathcal{G}_u \neq) .$$

- **NDCG@10** (↑):

$$\mathrm{NDCG@10} = \frac{1}{|\mathcal{U}_{\mathrm{eval}}|} \sum_{u \in \mathcal{U}_{\mathrm{eval}}} \frac{\mathrm{DCG}_u@10}{\mathrm{IDCG}_u@10}, \quad \mathrm{DCG}_u@10 = \sum_{r=1}^{10} \frac{2^{\mathrm{rel}_{u,r}} - 1}{\log_2(r+1)}.$$

  If $\mathrm{IDCG}_u@10 = 0$, we set the corresponding user-level NDCG to zero.

- **CTR** (↑): click-through rate induced by the simulator under the evaluated recommender setting,

$$\mathrm{CTR} = \frac{N_{\mathrm{click}}}{N_{\mathrm{impression}}},$$

  where N_{click} and $N_{\mathrm{impression}}$ denote the total numbers of clicks and impressions, respectively.

Scalability and Cost. We evaluate whether PsyBer-Agent remains efficient and stable under large-scale concurrent simulation. Let $N_{\mathrm{submitted}}$ denote the number of tasks placed in the queue, N_{finished} the number of tasks that terminate with either success or failure, and N_{success} the number of tasks that complete successfully. Let T_{wall} denote the elapsed time from queue initialization until the last worker terminates.

- **Throughput** (↑): aggregate terminal tasks per second across all workers,
$$\text{Throughput} = \frac{N_{\text{finished}}}{T_{\text{wall}}}.$$
- **p95 Latency/Decision** (↓): the 95th percentile of per-decision latency across completed runs.
- **Success Rate** (↑):
$$\text{SuccessRate} = \frac{N_{\text{success}}}{N_{\text{submitted}}}.$$
- **Total Token Cost** (↓): total LLM token consumption across all decisions,
$$\text{TokenCost} = \sum_{q \in \mathcal{Q}} \text{Tok}(q),$$
where $\mathcal{Q}$ denotes all LLM calls made during the evaluation and $\text{Tok}(q)$ is the total number of input and output tokens consumed by call q. For Table 5, throughput is computed globally across all workers using the same wall-clock interval reported in the table.

Group Behavior. For controlled social-cue perturbations, we use the following diagnostic metrics to quantify emergent collective behavior.

- **Herding Gain**: increase in interaction probability after positive social cues are displayed.
- **Comment Conformity** (↑): fraction of generated feedback aligned with the displayed comment-board sentiment. This metric is used as a manipulation check for whether displayed comment-board sentiment is perceived and reflected in generated feedback. It is not used as standalone evidence of action-level herding.
- **Polarization Shift** (Δr): change in exposure-similarity assortativity induced by social-cue perturbations,
$$\Delta r = r_{\text{treat}} - r_{\text{control}},$$
where r_{treat} and r_{control} are the assortativity coefficients under perturbed and control social cues, respectively.
- **Contrast–Interest Correlation** (ρ_{sp}): Spearman correlation between native item visual contrast and agent interest, used to quantify whether visual salience systematically affects collective interaction patterns.

Ablation Metrics. For component and information-channel ablations, we report simulation-fidelity metrics that capture behavior, exposure, and latent-state deviations.

Here, CTR_{sim} and CTR_{exp} denote the average click-through rates in the simulated and expert-reference environments, respectively. For Exp-JS, we compare the normalized item-exposure distributions induced by the simulator and the expert policy. For CES-MSE, $\mathcal{M}$ is the set of matched simulated–expert user pairs, $T_{u,u'}$ is the aligned interaction horizon for pair (u, u'), and $\mathbf{z}_{u,t}^{\text{sim}}$ and $\mathbf{z}_{u',t}^{\text{exp}}$ are the corresponding latent user-state vectors at step t; the factor 3 averages over the three latent dimensions.

- **CTRGap** (↓):
$$\text{CTRGap} = \left|\text{CTR}_{\text{sim}} - \text{CTR}_{\text{exp}}\right|.$$
- **Exposure JSD (Exp-JS)** (↓): Jensen–Shannon divergence between simulated and expert-reference item exposure distributions.
- **CES-MSE** (↓):
$$\text{CES-MSE} = \frac{1}{3\sum_{(u,u') \in \mathcal{M}} T_{u,u'}} \sum_{(u,u') \in \mathcal{M}} \sum_{t=1}^{T_{u,u'}} \left\| \mathbf{z}_{u,t}^{\text{sim}} - \mathbf{z}_{u',t}^{\text{exp}} \right\|_2^2.$$

CES-MSE measures internal agreement between simulator-generated latent states and reference states reconstructed by replaying observed interactions through Psy-Engine. Because both trajectories use the same model-defined C/E/S space, a lower CES-MSE indicates greater internal state consistency but does not establish correspondence to independently measured human psychological states.

Robustness Metrics. For robustness analysis, we use stress-test-specific metrics for behavior, decision, and system robustness.

- **Perturbation Exposure JSD** (↓): Jensen–Shannon divergence between the perturbed and clean exposure distributions.
- **Relative Drop** (↓): percentage degradation relative to the clean in-domain reference. For the long-horizon stress experiment, this metric is interpreted as a within-method degradation measure; claims of superior long-term stability require comparison with baselines evaluated at the same horizons.
- **Decision Agreement** (↑): fraction of perturbed rollouts that preserve the same selected action as the clean condition.
- **Policy Shift** (↓): total variation distance between clean and perturbed action distributions.
- **Latency/Decision** and **p95 Latency** (↓): average per-decision latency and 95th percentile execution latency under perturbations.
- **Stress Success Rate** (↑): fraction of simulation runs completed successfully under system-level perturbations.

APPENDIX D ADDITIONAL DETAILS

D.1 DETAILS OF THE EVALUATED DATASETS

In this study, we evaluate PsyBer-Agent on a diverse set of multimodal recommendation datasets spanning movie, e-commerce, short-video, and cross-domain recommendation scenarios. The PsyBer Benchmark includes seven datasets in total: **MM-ML-1M** Zhang et al. (2026b), four **Amazon Review** domains Hou et al. (2024), **MicroLens-100K** Ni et al. (2025), and **NineRec-Bili500K** Zhang et al. (2025a). The detailed statistics of each dataset are summarized in Table 8.

Table 8: Statistics of datasets used in the PsyBer Benchmark.

Dataset	#Users	#Items	#Inter.	Sparsity	Avg. Inter./User
MM-ML-1M	6,040	3,952	1,000,209	95.81%	165.60
Amazon_Magazine	380	865	2,679	99.18%	7.05
Amazon_All_Beauty	22,363	12,101	198,502	99.93%	8.88
Amazon_Luxury_Beauty	1,054	791	12,397	98.51%	11.76
Amazon_Musical_Instruments	25,402	10,611	210,646	99.92%	8.29
MicroLens-100K	100,000	19,738	719,405	99.96%	7.19
NineRec-Bili500K	500,000	138,033	7,845,805	99.99%	15.69

MM-ML-1M Zhang et al. (2026b) is a multimodal extension of the classic MovieLens benchmark, enriching user–item interactions with high-quality movie posters and detailed plot summaries. It supports recommendation evaluation with rich visual and textual item signals.

Amazon Review domains Hou et al. (2024) are built from the Amazon Reviews corpus and include four representative domains: **Magazine_Subscriptions**, **All_Beauty**, **Luxury_Beauty**, and **Musical_Instruments**. Each domain is paired with multimodal product metadata, including item images and textual descriptions, providing a realistic testbed for e-commerce recommendation.

MicroLens-100K Ni et al. (2025) is a large-scale short-video recommendation dataset. In addition to behavioral feedback, each item is associated with multimodal content signals, including titles, cover images, audio, and video, enabling evaluation beyond ID-based recommendation.

NineRec-Bili500K Zhang et al. (2025a) is a large-scale dataset drawn from a cross-domain recommendation benchmark, with aligned multimodal item metadata such as descriptive text and cover images. It is used to evaluate the robustness of simulated-user behavior under transferable and cross-domain recommendation settings.

D.2 Details of the Evaluated Recommender Systems

To evaluate PsyBer-Agent across diverse recommendation environments, we select seven representative recommender backbones covering five mainstream paradigms: heuristic methods, collaborative filtering, graph neural networks, deep generative models, and sequential recommendation. The details of these models are summarized in Table 9.

Table 9: Recommender system models evaluated with PsyBer-Agent.

Model Name	Model Type	Core Mechanism
PopRec Cremonesi et al. (2010)	Heuristic Baseline	Item popularity ranking based on historical interaction frequency.
BPR-MF Rendle et al. (2012)	Collaborative Filtering	Matrix factorization optimized with pairwise ranking loss.
LightGCN He et al. (2020)	Graph Neural Network	Simplified linear message passing on the user–item bipartite graph.
Multi-VAE Liang et al. (2018)	Deep Generative Model	Variational autoencoder with multinomial likelihood for implicit-feedback recommendation.
GRU4Rec Hidasi et al. (2015)	Sequential Recommendation	Gated recurrent units for session-based sequence modeling.
SASRec Kang & McAuley (2018)	Sequential Recommendation	Unidirectional self-attention for sparse sequential recommendation.
BERT4Rec Sun et al. (2019)	Sequential Recommendation	Bidirectional self-attention trained with masked item prediction.

D.3 Details of the User Simulator Baselines

Table 10: Categorization of user simulator baselines.

Methods	Category
RecAgent Wang et al. (2025a)	Social simulation
Agent4Rec Zhang et al. (2024a)	Generative-agent simulation
UserSim Zhang et al. (2025b)	RL-oriented simulation
DyTA4Rec Wanyan et al. (2025)	Temporal-aware simulation

We compare PsyBer-Agent with representative agent-based user simulators for recommender-system evaluation, covering social simulation, generative-agent simulation, RL-oriented evaluation, and temporal-aware modeling. RecAgent Wang et al. (2025a) simulates recommendation and social behaviors with profile, memory, and action modules; Agent4Rec Zhang et al. (2024a) constructs generative users from recommendation data with persona/profile, memory, and reasoning modules; UserSim Zhang et al. (2025b) provides LLM-based engagement inference for RL recommender evaluation; and DyTA4Rec Wanyan et al. (2025) captures temporal preference dynamics. Compared with these baselines, PsyBer-Agent jointly combines explicit psychological state modeling, real-log psycho-behavioral alignment, WebSim-based interaction, and million-scale simulation.

Baseline inclusion criterion. We include a method in the quantitative comparison only when an official or author-provided implementation, required checkpoints, and a protocol-compatible action interface are available at the experimental freeze date. The method must also accept the available user histories, item metadata, and fixed candidate pages, and its outputs must be evaluable under the same metrics as PsyBer-Agent.

We considered PUB, RecInter, AlignUSER, ContextSim, RecUserSim, and CSHI as relevant recent methods. However, at the experimental freeze date, we could not obtain a complete executable artifact satisfying the above requirements for these methods. Our reproducibility records document the artifact source, version or commit when available, date checked, and the specific protocol mismatch observed for each method. These observations describe the materials we could verify at the freeze

date; they do not assert that the methods are unavailable, unreleased, or inferior. We therefore discuss these methods as related work but do not report reimplementations based only on paper descriptions, since such reimplementations would introduce additional implementation and tuning choices. At the experimental freeze date, we excluded methods only when their artifacts or interfaces could not be verified under the fixed protocol; our per-method records include the source, version, date, and observed limitation. This exclusion reflects a reproducibility and protocol-compatibility constraint, rather than a claim about the relative quality of these methods.

D.4 Details of the Prompt Design

This section presents the prompt used to drive the LLM for final action selection. The prompt template is shown below:

```
Prompt:
    "You are an online user. Please select your preferred option based on
         the screenshot and click the corresponding image. You must
        output only JSON with no additional text.\n"
    "You will receive the following decision-making information
        simultaneously: current psychological state (psy_state),
        potential choices inferred from your psychological state (
        fast_policy_candidates, top-k), recent click history (
        recent_clicks), and available items on the current page (
        candidates).\n"
    "Make your final decision through comprehensive judgment.\n"
    "Available Actions:\n"
    '1) {"action":"click","item_id":"<item_id from candidates>","reason
        ":"..."}\n'
    "Rules:\n"
    "- Only perform click actions using item_ids from the candidate list
        .\n"
    "- If next_enabled=true and Next is not included in
        fast_policy_candidates, you must select click. In addition, you
        cannot consecutively select Next by referring to recent_clicks.\n
        "
    "- If all candidates on the current page are contained in
        clicked_item_ids and next_enabled=true, you must select next.\n"
    "- If next_enabled=false, you must select click.\n\n"
    "Notes:\n"
    "- Your current psychological state and fast_policy_candidates (
        choices predicted from your psychological state) serve only as
        reference signals, not hard constraints.\n"
    "- In case of conflicts among multiple reference signals, select your
         preferred option based on the screenshot and behavioral
        coherence as the final judgment.\n\n"
    f"dataset={memory.selected_dataset_key}, model={memory.
        selected_model_key}\n"
    f"next_enabled={str(next_enabled).lower()}\n"
    f"status_text={status_text}\n"
    f"psy_state={psy_state_view}\n"
    f"psy_pareto_front={psy_pareto_front}\n"
    f"psy_candidates={psy_candidate_text}\n"
    f"fast_policy_candidates={fast_policy_text}\n"
    f"recent_clicks={recent_lines}\n"
    f"clicked_item_ids={clicked_item_text}\n"
    f"candidates={card_lines}\n"
```

Here, `psy_state` denotes the current psychological state; `psy_pareto_front` denotes the Pareto-optimal action subset under Psy-Engine; `psy_candidates` provides psycho-behavioral annotations of the current-page candidates; `fast_policy_candidates` gives the top-k suggestions from the learned fast policy; `recent_clicks` records recent click history; and `status_text` summarizes the current interaction context. Along with this Prompt, a live WebSim Platform Page screenshot is passed to drive the LLM. Below is the decomposition Figure 5 of our prompt.

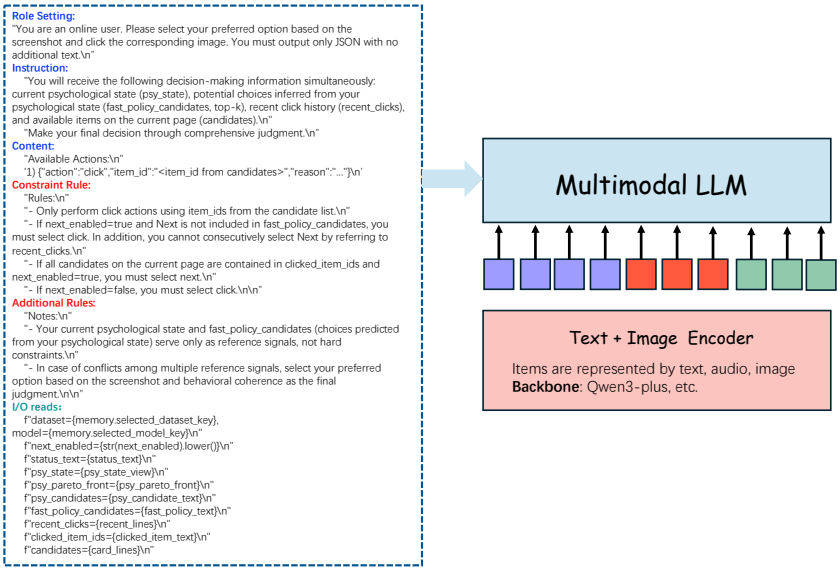


Figure 5: The schematic diagram of the specific scheme for reasoning modeling using a multimodal large model.

D.5 Details of Experimental Hardware Environment and Parallel Simulation Settings

Experimental Hardware Environment All experiments are conducted on a dedicated server running Ubuntu 20.04.6 LTS, equipped with an Intel Xeon Gold 5218R CPU @ 2.10GHz (80 cores), 251 GB of memory, and 787G of root disk space. The graphics card is 6 $\times$ NVIDIA RTX 3090 GPUs. The inference of the LLM driving the Agent is mainly implemented by calling the vendor API.

Large-scale Concurrent Simulation Setup PsyBer-Agent uses a queue-based rollout launcher for large-scale simulation. Given a target number of simulated users, the launcher first creates a task queue and starts a bounded number of asynchronous workers. Each worker takes one task from the queue and runs an independent WebSim agent process, so each simulated user maintains its own persona, Psy-Engine state, browser trace, and memory file. The launcher controls the active parallelism with a maximum-concurrency parameter and additionally applies launch-rate limiting to avoid bursty process creation. For LLM/VLM-based decisions, an API-pool mechanism assigns each task an available API slot and releases it after completion, preventing all agents from competing for the same endpoint. Failed or timed-out runs are isolated at the process level, while batch-level logs and summaries are collected for throughput, success rate, timeout rate, and decision-call statistics.

Cost Model and Unit Cost. Under a fixed configuration, the resource cost approximately satisfies Cost(N,R) $\propto NR$, where N is the total number of simulated tasks and R is the number of interaction rounds per task, so that NR denotes the total number of interaction rounds generated. In Table 5, increasing the workload from 100K to 1M and from 1M to 5M increases token consumption by approximately 10.45$\times$ and 5.09$\times$, while total execution time increases by approximately 10.19$\times$ and 5.02$\times$, respectively, indicating approximately linear rather than exponential cost growth. The unit cost is approximately \$4.45 per 1K tasks; doubling the number of interaction rounds would also approximately double the API cost. We therefore frame Table 5 as stable execution under million-scale stress rather than evidence that 5M tasks are suitable for routine evaluation. The reported token total includes offline trajectory construction, UGW alignment and policy training, browser/environment calls, and online LLM/VLM fallbacks; it therefore cannot be inferred from the small fallback percentage alone.

Fallback Usage. When the fast policy cannot make a reliable decision, the system invokes an LLM/VLM fallback; when all decision routes fail, a random fallback selects a valid action. Across large-scale runs, the LLM/VLM fallback is triggered in only $\approx$0.6% of decisions and the random

fallback in ≈0.04%, so the fast policy handles the overwhelming majority of decisions and fidelity is not dominated by fallback behavior.

D.6 Details of Fixed-Backend Human–Simulator Agreement Protocol

The fixed-backend human–simulator agreement study consists of an experimental group of 500 real human users and 500 simulated PsyBer-Agent users, each completing 10 consecutive browsing rounds. Both groups interact with the same recommendation interface and backend models in every round, with the decision-maker (human vs. agent) as the only independent variable. Participants were genuine anonymous users of the website, gave informed consent, and were informed about the purpose, procedure, and duration of the study. The specific sensitive information of users will also not be involved during the experiment. HR@10, NDCG@10, and CTR are computed using the same aggregation protocol for both groups. The relevance set for HR@10/NDCG@10 follows the held-out interaction target defined in Appendix C.6, and candidate slates are generated by the fixed backend and replayed identically for both decision-makers. The agreement is measured by the absolute difference between the corresponding metrics. Interaction logs are stored in anonymized form. The simulated users are initialized from the corresponding pre-window persona and history, and the 10 repeated decisions are treated as clustered within user. We use a prespecified user-level paired-bootstrap procedure. The tables report point estimates; when user-level outcomes are available, 95% intervals are obtained from the 2.5th and 97.5th percentiles of the paired bootstrap distribution.

Appendix E Additional Experiments & Analysis

E.1 PsyBer-Agent Group Behavior Exploration.

We probe collective dynamics in WebSim by fixing the candidate pool and base ranking while perturbing only social cues, including rating priors, hotness badges, and comment sentiment. We report herding gain, Comment Conformity (Conf.), polarization shift Δr, and contrast–interest correlation. As shown in Table 11, PsyBer-Agent reliably reflects local comment sentiment, as evidenced by high Sent. Align., but this cue uptake does not translate into strong action-level herding, polarization, or visual-salience-driven behavior.

Table 11: Group behavior under social-cue perturbations in WebSim.

Scenario	Backbone	Herding	Conf.	Δr	Contrast Corr.
MM-ML-1M	BPR-MF	0.005	0.996	0.006	-0.005
	LightGCN	0.005	0.992	0.009	-0.034
	Multi-VAE	0.016	0.995	0.007	0.018
	SASRec	-0.069	0.998	-0.002	0.029
Amazon Magazine	BPR-MF	0.002	1.000	0.012	0.014
	LightGCN	0.000	0.000	0.000	0.000
	Multi-VAE	-0.011	1.000	-0.012	-0.038
	SASRec	0.003	1.000	0.006	-0.082

Findings 4

PsyBer-Agent responds to social cues mainly through high sentiment conformity, while herding, polarization, and visual-salience effects remain limited in most tested settings.

E.2 Psy-Engine and PsyBer-Agent Parameter Sensitivity Analysis

This section examines the sensitivity of PsyBer-Agent to its update and training hyperparameters. We additionally specify an aggregation-weight sensitivity analysis to assess the dependence of behavioral fidelity on the equal-weight rules used in Psy-Engine.

Update and Training Hyperparameters. We study two hyperparameters: the Psy-Engine update rate α and the alignment learning rate `lr`. This analysis evaluates model-defined latent-state con-

sistency and runtime efficiency. CES-MSE is an internal consistency metric and does not establish correspondence to independently measured human psychological states.

Evaluation Protocol. We adopt a one-factor-at-a-time protocol around a fixed base configuration. For each hyperparameter, we sweep 4–5 values while keeping all other settings unchanged. Each setting is repeated five times with random seeds, and Table 12 reports the mean CES-MSE and Latency/Decision.

Table 12: Sensitivity results for α and `lr`. Bold entries in the Values column indicate the default settings. Results are means over five runs.

Parameter	Values	CES-MSE ↓	Latency (s) ↓
α	0.10	0.0156	4.11
	0.20	0.0070	4.08
	0.35	0.0023	4.37
	0.50	0.0077	4.03
	0.70	0.0121	4.04
`lr`	10^{-3}	0.0038	4.04
	$\mathbf{3 \times 10^{-3}}$	0.0027	4.02
	5×10^{-3}	0.0800	5.78
	10^{-2}	0.0026	7.51

Within the respective sweeps in Table 12, $\alpha = 0.35$ achieves the lowest mean CES-MSE, with higher latency than the other tested values. For the learning rate, $\texttt{lr} = 10^{-2}$ yields a slightly lower mean CES-MSE than 3×10^{-3} but substantially higher latency. We use $\alpha = 0.35$, and $\texttt{lr} = 3 \times 10^{-3}$ as the default configuration. These results characterize internal consistency and efficiency under the evaluated settings.

Aggregation-Weight Sensitivity To examine whether behavioral fidelity depends on the precise equal-weight setting, we parameterize the attractiveness, fatigue, and affective-target functions as

$$\begin{aligned} A_t &= w_A \nu_t + (1 - w_A) n_t, \\ F_t &= w_F c_t + (1 - w_F) \sigma_t, \\ E_t^\star &= w_\beta \beta_t + w_\rho \rho_t + w_p p_t, \end{aligned} \tag{51}$$

where $w_A, w_F \in [0, 1]$, $w_\beta, w_\rho, w_p \geq 0$, and $w_\beta + w_\rho + w_p = 1$. These constraints preserve bounded outputs for inputs in $[0, 1]$. The original formulation corresponds to $w_A = w_F = 0.5$ and $w_\beta = w_\rho = w_p = 1/3$.

Prespecified Protocol and Scope. The analysis comprises eight prespecified configurations, listed in Table 13, with one aggregation function varied at a time. This protocol quantifies sensitivity to the tested aggregation choices; it does not claim that equal-weight aggregation is optimal or psychologically validated. The self-congruence target, state-update equations, and remaining hyperparameters are held fixed.

For each configuration, the protocol requires regenerating the training reference trajectories and retraining the regression prior and fast policy, using identical data splits, training budgets, and seed sets. This ensures that each variant is evaluated with a policy trained under its own state dynamics, rather than introducing a mismatch between training and inference.

The experiment evaluation uses MM-ML-1M dataset, with a fixed recommender backbone within each scenario and five random seeds per configuration. Behavioral fidelity will be assessed against held-out real interactions using CTRGap, Interaction JSD, and Decision Accuracy. The variant without Psy-Engine will be evaluated under the same protocol to contextualize the magnitude of weight-induced changes. The results shows in Table 14.

Table 14 shows limited sensitivity to the tested aggregation weights, with maximum absolute changes from the default of 0.010 in CTRGap, 0.015 in Interaction JSD, and 0.007 in Decision Accuracy. No configuration performs best across all metrics: E1 achieves the highest accuracy, F1 the lowest JSD, and E3 the lowest CTRGap but a higher JSD, indicating trade-offs among behavioral fidelity

Table 13: Prespecified configurations for aggregation-weight sensitivity analysis. Each variant changes one aggregation function relative to the equal-weight default.

Configuration	w_A	w_F	w_β	w_ρ	w_p
Default	0.50	0.50	$1/3$	$1/3$	$1/3$
A1	0.25	0.50	$1/3$	$1/3$	$1/3$
A2	0.75	0.50	$1/3$	$1/3$	$1/3$
F1	0.50	0.25	$1/3$	$1/3$	$1/3$
F2	0.50	0.75	$1/3$	$1/3$	$1/3$
E1	0.50	0.50	0.60	0.20	0.20
E2	0.50	0.50	0.20	0.60	0.20
E3	0.50	0.50	0.20	0.20	0.60

Table 14: Aggregation-weight sensitivity results in a single scenario.

Configuration	**Varied term**	**CTRGap ↓**	**Interaction JSD ↓**	**Decision Acc. ↑**
Default	None	0.400 ± 0.012	0.290 ± 0.010	0.280 ± 0.009
A1	$w_A = 0.25$	0.407 ± 0.011	0.294 ± 0.012	0.277 ± 0.010
A2	$w_A = 0.75$	0.396 ± 0.014	0.298 ± 0.011	0.278 ± 0.008
F1	$w_F = 0.25$	0.403 ± 0.013	0.286 ± 0.010	0.281 ± 0.010
F2	$w_F = 0.75$	0.410 ± 0.012	0.296 ± 0.013	0.275 ± 0.009
E1	$(w_\beta, w_\rho, w_p) = (0.60, 0.20, 0.20)$	0.402 ± 0.011	0.287 ± 0.012	0.285 ± 0.009
E2	$(w_\beta, w_\rho, w_p) = (0.20, 0.60, 0.20)$	0.408 ± 0.014	0.301 ± 0.011	0.276 ± 0.011
E3	$(w_\beta, w_\rho, w_p) = (0.20, 0.20, 0.60)$	0.394 ± 0.013	0.305 ± 0.014	0.273 ± 0.010
w/o Psy-Engine	Reference baseline	0.450 ± 0.016	0.340 ± 0.015	0.245 ± 0.012

measures. All weighting configurations outperform the variant without Psy-Engine on the reported mean metrics; the default reduces CTRGap and JSD by 0.050 each and improves accuracy by 0.035.

Overall, the differences associated with removing Psy-Engine exceed those induced by weight adjustments, suggesting that retaining the engine matters more than the precise weighting choice under the tested setting. These results support equal weighting as a simple and effective default, without establishing its optimality or the psychological validity of the latent states.

E.3 PSYBER-AGENT ROBUSTNESS STRESS ANALYSIS

Robustness is a prerequisite for a user agent to serve as a reliable simulation proxy. We therefore stress-test PSYBER-AGENT from three complementary perspectives: behavior robustness, decision robustness, and system robustness. For all robustness experiments, we define a clean base condition and compare each perturbation against it using the same evaluation protocol. We report metrics tailored to each stress test, including exposure-distribution deviation, CES sequence consistency, decision agreement, policy shift, success rate, and latency. See Figure 6 for the robustness results visualization.

Robustness Evaluation Protocol: For behavior robustness, we report `Exposure JSD` and `CES-MSE` as absolute fidelity metrics against expert-reference logs, and `Relative Drop` as the degradation of the primary in-domain evaluation score with respect to the clean short-horizon reference. For decision robustness, we use `Decision Agreement`, `Policy Shift`, and `Latency/Decision` to quantify action stability, distributional drift, and decision efficiency. For system robustness, we report `Success Rate` and `p95 Latency` to measure execution reliability and latency under memory perturbations.

Behavior robustness. We evaluate whether PsyBer-Agent preserves stable and interpretable behavior under contextual perturbations and environment shift. For user-profile consistency, we fix the same underlying profile and perturb the interaction context with prompt paraphrases, transient affective shifts, and noisy history observations. We quantify robustness using Exposure JSD and CES-MSE, which measure absolute exposure-distribution deviation and psychological-sequence deviation relative to expert-reference logs. For long-range and cross-domain stability, we vary the rollout horizon (track $= 20/50/100$) and evaluate transfer to an unseen target domain. We report both absolute fidelity metrics and Relative Drop (%) with respect to the in-domain short-horizon clean reference.

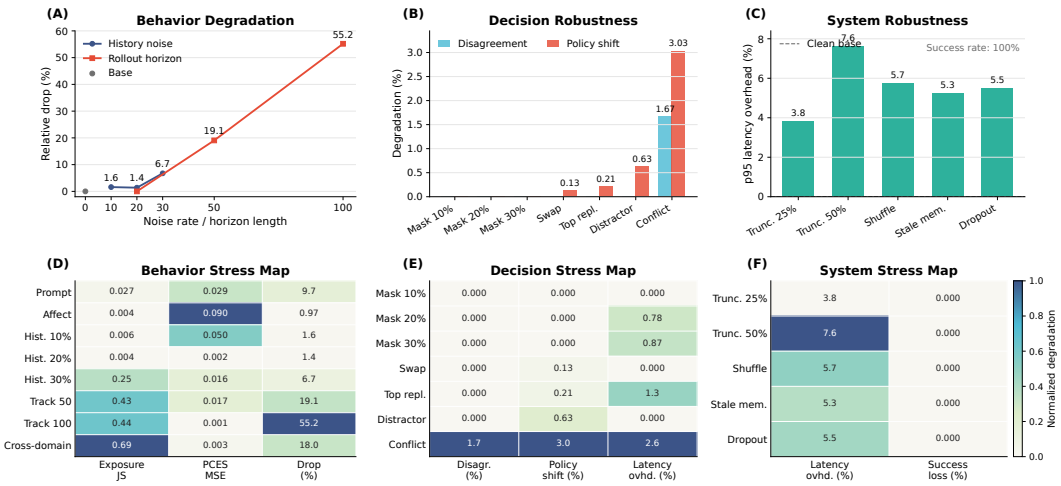


Figure 6: Robustness evaluation of PsyBer-Agent. **(A)** behavior degradation, **(B)** decision robustness, and **(C)** system robustness, together with the corresponding normalized stress maps for **(D)** behavior, **(E)** decision, and **(F)** system robustness, where darker colors indicate larger degradation.

Table 15: Behavior robustness of PsyBer-Agent under contextual perturbations and domain shifts.

Setting	Severity	Exposure JSD ↓	CES-MSE ↓	Relative Drop (%) ↓
Base	**clean**	0.5950	0.0636	–
Perturbation	mild	0.0266	0.0294	9.6608
Affective shift	mild	0.0035	0.0900	0.9705
History noise	10%	0.0058	0.0500	1.6055
	20%	0.0035	0.0016	1.3801
	30%	0.2542	0.0157	6.7454
Long horizon	track=50	0.4279	0.0170	19.0528
	track=100	0.4363	0.0014	55.1798
Cross-domain	target domain	0.6931	0.0032	18.0066

As shown in Table 15, PsyBer-Agent remains robust under local in-domain perturbations but degrades under longer horizons and domain shift. Because Exposure JSD and CES-MSE are measured against expert-reference logs rather than against the clean rollout, mild perturbations can occasionally improve one absolute fidelity metric; Relative Drop therefore serves as the primary summary of robustness degradation. Under this criterion, mild perturbation and affective shift yield limited degradation (Relative Drop $= 9.66\%$ and 0.97%), and history noise remains stable at 10%–20% before a clearer decline at 30% noise. In contrast, long-range rollout shows substantial accumulation effects, where Relative Drop increases from 19.05% (track$=50$) to 55.18% (track$=100$), and cross-domain transfer exhibits the largest exposure mismatch (Exposure JSD $= 0.6931$). These results characterize PsyBer-Agent's degradation profile under the tested horizons; because comparable long-horizon curves for all baselines were not available under the same protocol, they do not establish superiority in long-term stability.

Decision robustness. We evaluate decision robustness by testing whether the decision module preserves stable choices under imperfect observations and mild perturbations to the recommendation slate. To assess noise tolerance, we randomly mask click/history signals at rates of 10%, 20%, and 30% and compare the resulting decisions with those under the clean setting. To assess sensitivity to ranking changes, we introduce small but structured slate perturbations, including adjacent swaps, top-rank replacements, distractor insertion, and conflicting signals.

Because this experiment targets decision stability rather than oracle-based A/B correctness, we report Decision Agreement, Policy Shift (measured as total variation distance between action distributions),

Table 16: Decision robustness of PsyBer-Agent under noisy observations and ranking perturbations.

Setting	Severity	Decision Agreement ↑	Policy Shift ↓	Latency/Decision (s) ↓
Base	**Clean**	1.0000	0.0000	4.3935
Click/history mask	10%	1.0000	0.0000	4.3566
	20%	1.0000	0.0000	4.4279
	30%	1.0000	0.0000	4.4319
Adjacent swap	1 pair	1.0000	0.0013	4.2522
Top-rank replacement	top-1	1.0000	0.0021	4.4487
Distractor insertion	1 item	1.0000	0.0063	4.3444
Conflicting signals	mixed	0.9833	0.0303	4.5094

and Latency/Decision. As shown in Table 16, click/history masking and mild ranking perturbations yield nearly identical decisions with negligible policy drift, indicating that light observation loss does not alter the top action. Conflicting signals produce the largest degradation in both agreement and policy shift, showing that PsyBer-Agent remains stable under modest perturbations while remaining sensitive to stronger contradictory evidence.

System robustness. Finally, we stress-test the full system under internal-state corruption. For memory perturbation, we modify the memory module through recent-window truncation, temporal shuffling, stale-memory injection, and random memory dropout, and then evaluate whether the agent can still maintain stable execution. We report Success Rate and p95 Latency to measure execution reliability and efficiency under memory noise.

Table 17: System robustness of PsyBer-Agent under memory perturbations.

Setting	Severity	Success Rate ↑	p95 Latency (s) ↓
Base Agent	**clean**	1.00	4.19
Memory truncation	25%	1.00	4.35
	50%	1.00	4.51
Memory shuffle	full	1.00	4.43
Stale memory injection	full	1.00	4.41
Memory dropout	30%	1.00	4.42

Table 17 shows that PsyBer-Agent remains stable under memory perturbations: all settings maintain a perfect success rate (1.00), and the latency overhead is modest (4.35s–4.51s vs. 4.19s in the clean setting; worst case: +7.6%). Overall, the results suggest graceful degradation rather than brittle failure, supporting PsyBer-Agent as a reliable simulation proxy under realistic memory noise.

E.4 Representative Benchmark Performance of Recommendation Models on the Test Platform

Table 18 reports the performance of the recommendation models used in the backend of the WebSim platform on a representative subset of three datasets: MovieLens-1M, Amazon All Beauty, and Amazon Magazine Subscriptions. These three datasets cover dense movie recommendation, sparse beauty recommendation, and a small-catalog shopping scenario, and are the settings most frequently used in the closed-loop experiments. The table should therefore be viewed as a backend sanity check rather than the primary benchmark result.

The near-ceiling results on some sparse shopping subsets indicate that these settings are comparatively easy for strong sequential recommenders and mainly serve to verify the backend recommendation layer used by WebSim.

Table 18: Representative benchmark performance of base recommendation models on MovieLens-1M, Amazon All Beauty, and Amazon Magazine Subscriptions.

Models	MovieLens-1M		All Beauty		Magazine Subscriptions	
	HR@10	NDCG@10	HR@10	NDCG@10	HR@10	NDCG@10
SASRec	0.3305	0.3170	0.4854	0.3219	0.2782	0.1102
LightGCN	0.0354	0.0176	0.4768	0.3782	0.2380	0.1166
Multi-VAE	0.0626	0.0299	0.5430	0.3726	0.2493	0.1182
PopRec	0.0368	0.0179	0.7736	0.5348	0.1406	0.0594
BPR-MF	0.0402	0.0188	0.2781	0.2363	0.1530	0.0946
GRU4Rec	0.1328	0.0656	0.5364	0.3831	0.1303	0.0578
BERT4Rec	0.2232	0.1159	0.5232	0.4767	0.2238	0.1217

E.5 Profile-Conditioned Group Fidelity Diagnostics

We further examine whether PsyBer-Agent produces structured behavioral variation across heterogeneous simulator-side user profiles. This analysis is intended as a diagnostic study rather than a strict real-user group benchmark: the offline logs provide aggregate user–item interactions but do not contain matched annotations for personality, consumption value, or identity-gap groups. Therefore, each simulated group is compared against the overall real interaction distribution within the same scenario. The goal is to test whether profile-conditioned agents remain anchored to real aggregate behavior while exhibiting interpretable group-level differences.

Each agent's memory contains a structured user profile with three components: personality type, consumption-value type, and latent real-self/ideal-self preference embeddings. We construct three profile groupings. The `Personality` grouping contains social-driven, novelty-seeking, risk-averse, and goal-oriented users. The `Consumption Value` grouping contains brand-loyal, value-for-money, experience-first, pragmatic, trend-sensitive, and status-oriented users. The `Identity Gap` grouping is computed as

$$g_u = 1 - \cos(\mathbf{u}_u^{\text{real}}, \mathbf{u}_u^{\text{ideal}}),$$

where $\mathbf{u}_u^{\text{real}}$ and $\mathbf{u}_u^{\text{ideal}}$ denote the agent's real-self and ideal-self preference embeddings. We split agents into low-, medium-, and high-gap groups using scenario-specific tertiles. For each group, we aggregate simulated click sequences and compare the resulting item distribution with the real aggregate interaction distribution using JSD, Pearson correlation (R), and EMD.

As shown in Table 19, PsyBer-Agent exhibits non-trivial profile-conditioned variation while remaining close to the aggregate real interaction distribution. Across nearly all profile groups, Amazon Magazine Subscriptions obtains lower JSD and EMD than MM-ML-1M, indicating that behavior in the narrower e-commerce domain is easier to align with real aggregate interaction patterns. In contrast, MM-ML-1M shows larger variation across profile groups, consistent with the higher semantic diversity and broader preference space of movie consumption.

The profile-group results provide additional diagnostic signals. Across personality groups, socially driven users achieve strong or near-strong alignment in both scenarios, whereas novelty-seeking users tend to have higher JSD, suggesting that exploratory behavior induces greater long-tail exposure. For consumption-value groups, value-for-money users show the strongest alignment on MM-ML-1M, with the lowest JSD and highest Pearson R, whereas trend-sensitive and status-oriented users are harder to match. On Amazon Magazine Subscriptions, value-for-money and brand-loyal users remain well aligned, while pragmatic users show the largest distributional gap. For identity-gap groups, the three tertiles remain relatively stable, but low-gap users achieve the best JSD on MM-ML-1M, suggesting that agents whose real-self and ideal-self preference embeddings are closer generate more predictable behavior.

Overall, this diagnostic indicates that PsyBer-Agent does not collapse different personas into a single average behavior pattern. Instead, it preserves structured profile-conditioned differences while maintaining reasonable agreement with real aggregate logs. Since the offline data lack matched real-user profile labels, these results should be interpreted as evidence of simulator-side profile consistency and aggregate-log anchoring, rather than direct proof of real-world persona-level fidelity.

Table 19: Profile-conditioned group fidelity diagnostics across user-profile groups.

Scenario	Group Type	User Group	Runs	Clicks	Unique	JSD↓	$R\uparrow$	EMD↓
MM-ML-1M	Personality	social-driven	188	3741	1946	0.3310	0.2789	0.2776
		novelty-seeking	170	3400	1682	0.3712	0.2485	0.2802
		risk-averse	178	3534	1849	0.3393	0.2738	0.2579
		goal-oriented	207	4150	2051	0.3319	0.2286	0.2807
	Consumption Value	brand-loyal	222	4422	2195	0.3127	0.3306	0.2774
		value-for-money	244	4880	2280	0.3042	0.3347	0.2607
		experience-first	200	3981	2105	0.3266	0.1419	0.2976
		pragmatic	202	4050	1834	0.3503	0.2749	0.2550
		trend-sensitive	177	3539	1650	0.3828	0.2233	0.2759
		status-oriented	163	3253	1622	0.3671	0.2670	0.2611
	Identity Gap	low	403	8016	2939	0.2737	0.2893	0.2675
		medium	405	8110	2910	0.2762	0.2771	0.2787
		high	400	7999	2853	0.2861	0.2602	0.2657
Amazon Magazine Subscriptions	Personality	social-driven	193	3828	392	0.2076	0.3429	0.1976
		novelty-seeking	156	3093	392	0.2153	0.3160	0.1954
		risk-averse	152	3020	392	0.2137	0.3028	0.1947
		goal-oriented	210	4146	392	0.2093	0.3299	0.1949
	Consumption Value	brand-loyal	257	5108	392	0.2058	0.3414	0.1928
		value-for-money	234	4612	463	0.2083	0.3558	0.1898
		experience-first	210	4178	392	0.2076	0.3463	0.1931
		pragmatic	187	3685	392	0.2180	0.2915	0.2031
		trend-sensitive	181	3610	392	0.2139	0.3132	0.1984
		status-oriented	155	3058	392	0.2090	0.3305	0.1919
	Identity Gap	low	409	8116	392	0.2038	0.3518	0.1942
		medium	408	8059	392	0.2081	0.3288	0.1994
		high	407	8076	463	0.2052	0.3658	0.1842

The following diagnostic studies complement the benchmark and use the same data processing, candidate-generation protocol, and metric definitions unless otherwise specified.

E.6 Offline Backbone Sanity Check

Using the publicly released WebSim checkpoints and MovieLens-1M logs, we sampled 500 users with at least 20 pre-window interactions (seed 42), held out each user's last 10 chronological interactions, and evaluated the three available backbones. The intervals below are user-level percentile bootstrap 95% intervals with 2,000 resamples. This is an offline backend sanity check; it does not provide human outcomes or human–simulator ranking validation.

Table 20: Offline MovieLens-1M backbone sanity check on 500 users. Values are means with user-level 95% bootstrap intervals.

Backbone	HR@10	95% CI	NDCG@10	95% CI	Recall@10	95% CI
BPR-MF	0.2900	[0.2350, 0.3501]	0.2101	[0.1531, 0.2731]	0.0465	[0.0350, 0.0600]
LightGCN	0.2600	[0.2050, 0.3200]	0.1874	[0.1393, 0.2431]	0.0400	[0.0295, 0.0515]
SASRec	0.2550	[0.1950, 0.3150]	0.1767	[0.1306, 0.2264]	0.0340	[0.0255, 0.0430]

On this offline split, BPR-MF has the highest point estimates, followed by LightGCN and SASRec. We use this ordering only to verify the backend and held-out evaluation pipeline; it is not evidence of online recommender ranking validity.

E.7 Aggregate Baseline Comparison

The main-text baseline comparison in Table 4 is the primary quantitative result. It reports results for 500 users over 10 interaction steps per user, averaged across five random seeds. This appendix does not provide separate estimates for individual backbones; the main-text values should not be interpreted as backbone-specific measurements.

E.8 Fallback Mechanisms and Usage Statistics

PsyBer-Agent uses a hierarchical decision pipeline in which a distilled fast policy handles most decision steps. When the fast policy cannot produce a valid decision, the system invokes an LLM/VLM fallback; if all decision routes fail, a valid action is selected by random fallback. These mechanisms prevent the closed-loop simulation from stalling.

We monitor their activation rates as an operational diagnostic. Table 21 reports the rates over all decision steps in the evaluation workload.

Table 21: Fallback activation frequency over all decision steps.

Method	**LLM/VLM Fallback** $\downarrow$	**Random Fallback** $\downarrow$
PsyBer-Agent	$\approx 0.6\%$	$\approx 0.04\%$

The low activation rates indicate that most observed decisions are handled by the fast policy, while random fallback contributes only a small fraction of actions in the reported workload.

E.9 Large-Scale Cost and Execution Analysis

The large-scale evaluation uses a queue-based launcher with bounded asynchronous workers. Under a fixed configuration, the resource cost approximately follows

$$\mathrm{Cost}(N, R) \propto NR, \tag{52}$$

where N is the total number of queued simulation tasks and R is the number of interaction rounds per task.

Increasing the workload from 100K to 1M tasks increases token consumption and execution time by approximately $10.45\times$ and $10.19\times$, respectively. Increasing the workload from 1M to 5M tasks increases token consumption and execution time by approximately $5.09\times$ and $5.02\times$, respectively. Under the reported configuration, the estimated API cost is approximately \$4.45 per 1K tasks.

The experiments are conducted on a server equipped with an Intel Xeon Gold 5218R CPU with 80 cores, 251 GB RAM, 787 GB disk space, and six NVIDIA RTX 3090 GPUs. The 5M-task setting is therefore a stress test rather than a recommended routine evaluation size; practical deployment should select the workload and rollout length according to the required fidelity–cost trade-off. Each workload scale is evaluated over three runs. The 5M setting denotes the total number of queued tasks rather than 5M simultaneously active tasks. Each task corresponds to a complete WebSim trajectory with R interaction rounds.

These measurements characterize execution scaling under the reported configuration and are intended as a stress-test analysis rather than a claim of universal cost-effectiveness.

Alignment Ablation Protocol. To evaluate the contribution of trajectory alignment and prior-guided initialization, we compare four variants under a matched experimental protocol: direct behavioral cloning without trajectory alignment, deterministic relative-position matching, UGW with uniform initialization, and the proposed prior-guided UGW. For the direct behavioral cloning and relative-position variants, the expert position assigned to simulator step j is determined by

$$r(j) = 1 + \mathrm{round}\left(\frac{(j-1)(n-1)}{m-1}\right). \tag{53}$$

Table 22: Cost and execution statistics supplemented in the rebuttal.

Quantity	**Value**
Cost model	$\mathrm{Cost}(N, R) \propto NR$
Token increase, 100K $\rightarrow$ 1M	$\approx 10.45\times$
Token increase, 1M $\rightarrow$ 5M	$\approx 5.09\times$
Time increase, 100K $\rightarrow$ 1M	$\approx 10.19\times$
Time increase, 1M $\rightarrow$ 5M	$\approx 5.02\times$
Estimated API cost	$\approx \$4.45$ per 1K tasks
Runs per workload scale	3
CPU	Intel Xeon Gold 5218R, 80 cores
Memory	251 GB
Disk	787 GB
GPUs	6 $\times$ NVIDIA RTX 3090

Table 23: Alignment ablation results.

Variant	Alignment	Acc. $\uparrow$	CTRGap $\downarrow$	Lat./Dec. $\downarrow$
Direct BC	None	0.242	0.458	2.48
Relative-pos.	Deterministic matching	0.253	0.441	2.51
UGW-Uniform	Uniform initialization	0.267	0.418	2.51
Prior-guided UGW	Regression prior	0.278	0.406	2.53

If the corresponding expert item is absent from the current candidate slate, its target action is mapped to `skip`. For the UGW variants, all methods use the same state representation, expert-reference trajectories, candidate replay, policy architecture, cross-entropy objective, optimization budget, data split, and random-seed set. The only differences are the trajectory-matching procedure and the initialization of the transport matrix. We evaluate all variants on identical test rollouts using Decision Accuracy, CTRGap, Interaction JSD, and Latency/Decision. CES-MSE is reported as an auxiliary measure of consistency in the model-defined C/E/S state space. The comparison between UGW with uniform initialization and prior-guided UGW isolates the contribution of the regression-based initialization prior.

According to the experimental results, prior guided UGW has the best effect: accuracy 0.278 and CTRGap 0.406. Compared with relative position matching, uniform initialization improves the accuracy by 1.4pp and decreases the CTRGap by 0.023, which reflects the advantages of structure-aware trajectory alignment. With the introduction of prior guidance, the accuracy is further +1.1pp and CTRGap -0.012. The latency of each variant is similar (2.48 – 2.53s/ decision), which is consistent with offline alignment.

APPENDIX F PSYBER-AGENT CASE STUDY

F.1 MOVIELENS-BASED WEB BROWSING SCENARIO

We instantiate WebSim on the Multi-Modal MovieLens dataset as a movie-browsing interface for agent evaluation. As shown in Figure 7, the default page presents recommendation items in a one-row, four-column grid, while two drop-down menus allow switching the dataset and recommender under test. In addition to item clicking, the interface supports random reset and next-page navigation. In this case study, we use LightGCN as the driving recommendation system and MM-ML-1M as the dataset.

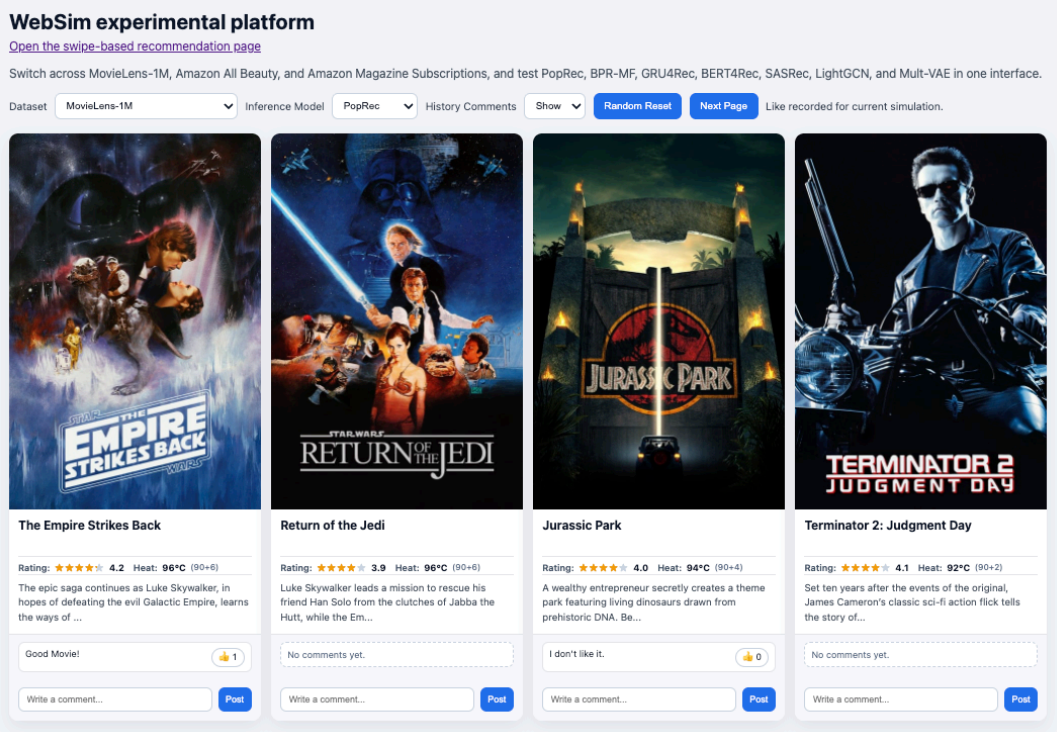
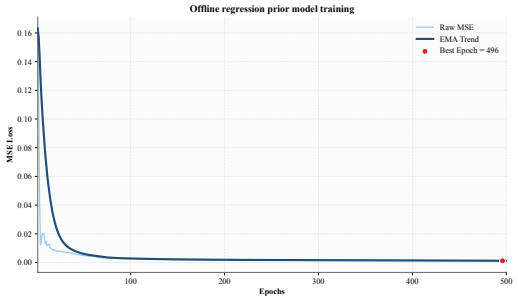


Figure 7: WebSim platform for the multimodal MovieLens browsing interface. Each movie item consists of a cover, title, rating, and popularity score. The task can switch the dataset and the recommender under test, and the agent may navigate to the next page through psychology-informed decision-making.

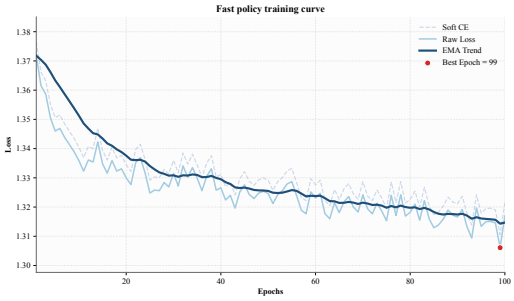
We first train an offline regression prior model to obtain an initialization prior for psycho-behavioral alignment. Figure 8a shows the training curve of the prior model. Based on this initialization, we further optimize the fast policy using UGW-based alignment to better link latent psychological changes to specific browsing behaviors, as shown in Figure 8b.

Figure 9 shows the average psychological factor trajectories across browsing steps from a simulation of 10,000 users. The three variables C, E, and S generally decrease as agents browse for longer horizons. Among them, self-congruence declines the fastest, which is consistent with relatively weaker profile–item alignment in this scenario.

Figure 10 compares the item popularity distribution in the offline dataset with the popularity of items selected by PsyBer-Agent. We report the Top-30 items for comparison. Except for a higher-than-reference selection probability for *Toy Story*, the remaining head-item probabilities broadly follow the offline popularity distribution.



(a) Offline regression prior model training curve



(b) Fast policy training curve after UGW-based alignment

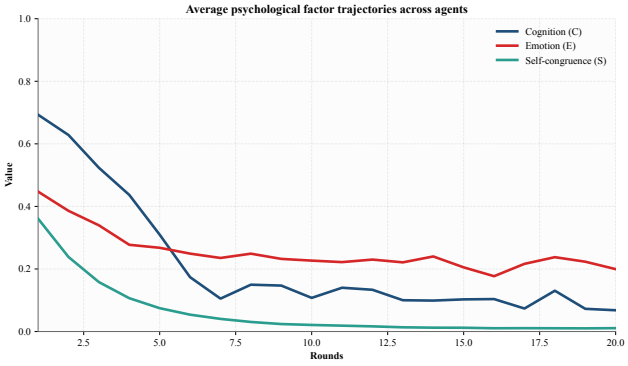


Figure 9: Average psychological factor trajectories across agents in the MovieLens scenario

F.2 Amazon Magazine Subscriptions Shopping Scenario

We instantiate WebSim on the Amazon Magazine Subscriptions subset of the Amazon Review dataset as a shopping interface for agent evaluation. As shown in Figure 11, the default page displays recommended items in a one-row, four-column grid, while two drop-down menus allow switching between datasets and the recommender under test. In addition to item clicking, the interface supports random reset and next-page navigation. In this case study, we use GRU4Rec as the driving recommendation model.

We first train an offline regression prior model on Amazon interaction data to obtain an initialization prior for psycho-behavioral alignment. Figure 12a shows the training curve of the prior model, where the loss decreases rapidly in early epochs and then converges stably. Based on this initialization, we further optimize the fast policy using UGW-based alignment to better link latent psychological changes to shopping behaviors such as clicks, skips, and pagination, as shown in Figure 12b.

Figure 13 reports the average psychological factor trajectories in a 10,000-user simulation. We observe consistent temporal dynamics across browsing steps, with gradual attenuation under prolonged exposure and noticeable sensitivity when recommendation relevance drops. This trend suggests that the Amazon-based simulator preserves interpretable latent-state evolution rather than producing purely random shifts in behavior.

Finally, Figure 14 compares item popularity in the offline dataset with the popularity of items selected by PsyBer-Agent. We report the Top-30 items for comparison. The two distributions are broadly aligned in the head, indicating that the simulator preserves the main demand pattern of real interactions while still showing selective deviations on several items due to psychology-driven preference dynamics.

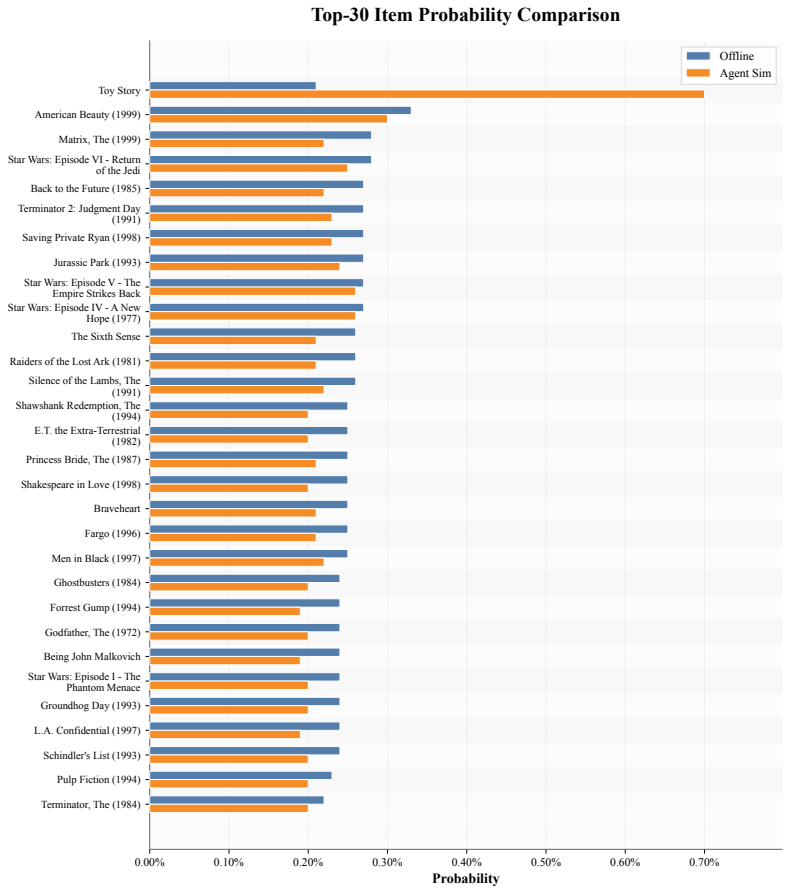


Figure 10: Top-30 item popularity comparison between the offline dataset and PsyBer-Agent selections in the MovieLens scenario

APPENDIX G DECLARATIONS

G.1 LIMITATIONS

PsyBer-Agent provides a controllable proxy for pre-deployment recommender evaluation, but it still has several limitations. First, its psychological state dynamics are designed around a compact three-dimensional representation of cognition, emotion, and self-congruence, which improves interpretability but may not capture all factors that shape real user behavior, such as social context, long-term habit formation, or domain-specific motivations. Second, the psycho-behavioral alignment module constructs teacher-forced pseudo-expert latent trajectories using the same Psy-Engine, because offline interaction logs do not contain ground-truth psychological annotations. As a result, these latent trajectories should be interpreted as log-conditioned reference states rather than human-validated psychological ground truth. If the inductive bias of Psy-Engine is misspecified, the resulting alignment signal may partially reinforce that misspecification, since the current framework does not directly anchor latent states to independently validated human psychological measurements. Third, although the simulator is aligned with offline interaction logs and evaluated across movie and e-commerce scenarios, its fidelity may degrade under substantially longer horizons, in unseen domains, or in recommendation interfaces whose interaction patterns differ from those in WebSim. Fourth, simulated users cannot fully replace real-user studies or online A/B testing, because LLM-based decisions and psycho-behavioral alignment may introduce systematic biases or overestimate deployment readiness. Finally, large-scale simulation depends partly on external LLM APIs and public interaction logs, so cost, latency, reproducibility across API versions, privacy protection, and fairness across user groups remain important practical considerations for future work.

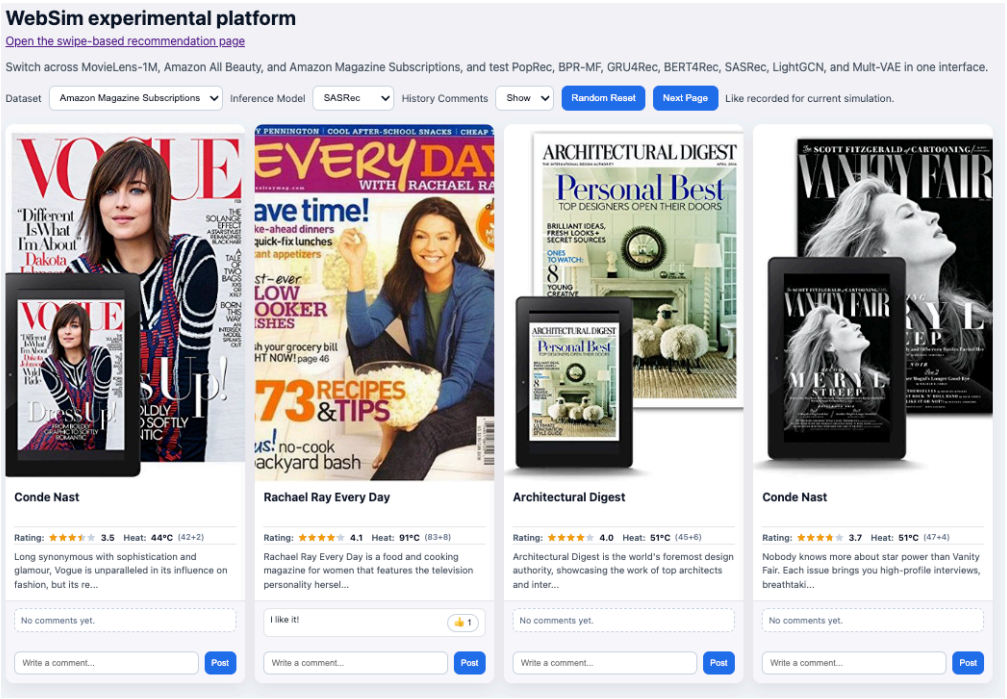
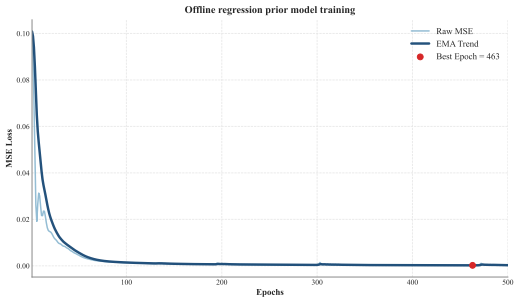
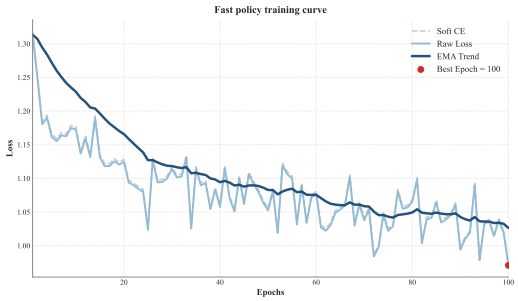


Figure 11: WebSim platform for the Amazon Magazine Subscriptions shopping interface. Each item consists of a cover, title, rating, and popularity score. The task can switch the dataset and the recommender under test, and the agent may navigate to the next page through psychology-informed decision making. Most Amazon Review subsets are supported in this mode.



(a) Offline regression prior model training curve



(b) Fast policy training curve after UGW-based alignment

G.2 Broader Impacts

This work supports safer and more cost-effective pre-deployment evaluation of recommender systems by using PsyBer-Agent and WebSim to simulate closed-loop user interactions before online release, helping identify behavioral mismatch, exposure concentration, robustness failures, and unintended group-level effects without immediately exposing real users to untested policies. At the same time, psychology-driven user simulation may be misused to optimize manipulative recommendation strategies, amplify engagement-driven feedback loops, or probe group responses to social cues in ways that increase polarization or unfair exposure; careless use of interaction logs may also raise privacy concerns. Therefore, PsyBer-Agent should be used as a diagnostic tool rather than the sole basis for deployment decisions, with responsible data handling, documented dataset provenance, monitoring for distributional bias and excessive herding, fairness checks across user groups, human oversight, and staged online validation in high-impact recommendation domains.

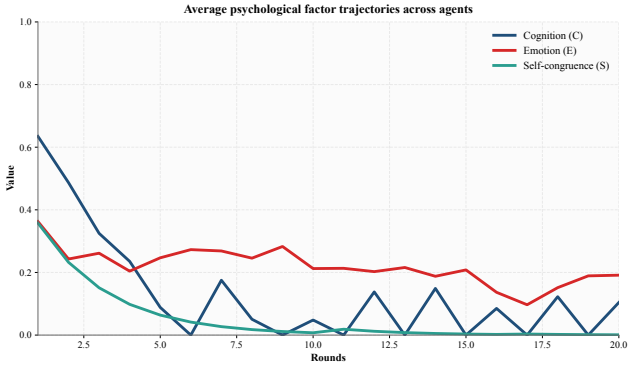


Figure 13: Average psychological factor trajectories across agents in the Amazon Magazine Subscriptions scenario

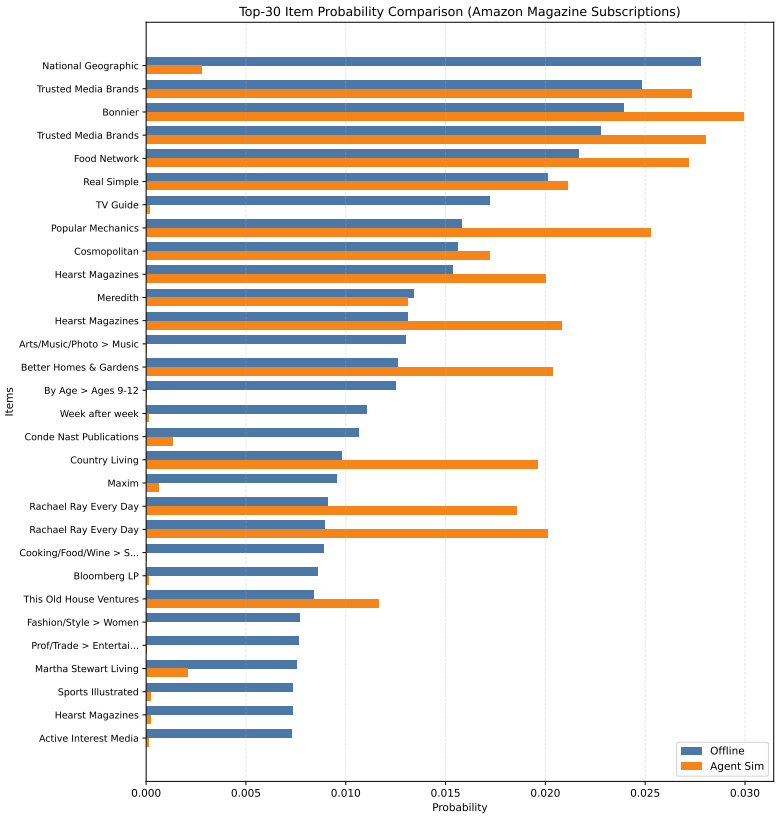


Figure 14: Top-30 item popularity comparison between the offline dataset and PsyBer-Agent selections in the Amazon Magazine Subscriptions scenario