USERARENA: BENCHMARKING INTERACTIVE USER SIMULATION FOR LONG-HORIZON SHOPPING AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Agentic recommendation pursues shopper goals through sequences of context-dependent decisions. Developing and evaluating such policies requires faithful recommendation world models that capture both the interactive shopping environment and users' behavioral responses over long interactions. Existing resources rarely provide long-horizon, fine-grained real-user trajectories or support trajectory-level evaluation and offline validation against production A/B results. We introduce **USERARENA**, a real-world e-commerce interactive benchmark containing 64,729 session trajectories from 11,010 users, spanning 566,102 items and 13 action types. USERARENA pairs these trajectories with an interactive environment and supports offline validation against outcomes from a controlled production A/B test, enabling five-level evaluation from output validity and local decisions to sequential behavior, personalization, and outcomes. To improve simulator fidelity over these long-horizon interactions, we further propose **O**n-**P**olicy **T**rajectory **D**istillation (**OPTD**), which uses trajectory-level intent to provide dense teacher targets on student-generated rollouts. Experiments show that OPTD achieves the strongest overall fidelity among the evaluated simulators under USERARENA's evaluation protocol and most closely reproduces the production policy effect in offline simulation.[1]

1 INTRODUCTION

Recent advances in language and multimodal agents create new opportunities for agentic recommendation, in which an agent pursues a shopper's goal through a sequence of context-dependent decisions (Liu et al., 2026; Shang et al., 2025). Post-training and evaluation of the underlying recommendation policy require a scalable *recommendation world model* that simulates how the shopping environment and its users respond to each policy action (Ie et al., 2019). An executable environment models what the policy can observe and change, while a user simulator models how shoppers respond to those changes (Yao et al., 2022; Zhou et al., 2024; Wang et al., 2025a; Zhang et al., 2024). Together, they support offline and counterfactual evaluation as well as reinforcement-learning-based optimization, reducing reliance on costly online experiments (Shi et al., 2019; Zhao et al., 2023).

Existing resources provide rich behavioral data and interactive evaluation for such world models. Amazon Reviews (Hou et al., 2026) contains ratings and review text, Qilin (Chen et al., 2025) adds multimodal content and app-level sessions, and OmniBehavior (Chen et al., 2026) captures longitudinal traces. RecBench+ (Huang et al., 2026) and OPeRA (Wang et al., 2026) move from offline records toward interactive evaluation. Yet two crucial gaps remain: **At the data level,** existing datasets rarely provide long-horizon trajectories that align real-user profiles, visual observations, product context, and actions across repeated sessions. **At the evaluation level,** next-action agreement measures step-wise choices in isolation but does not assess whether simulated trajectories preserve a user's behavioral tendencies over long horizons. Existing benchmarks also rarely test whether simulator-based policy comparisons agree with production A/B outcomes. These gaps limit assessment of long-horizon behavioral fidelity and the real-world validity of simulation conclusions.

[1] Our code and data are available at `https://anonymous.4open.science/r/UserArena-D877`. The project page is available at `https://anonymous.4open.science/w/UserArena-Page-031B/`.

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.

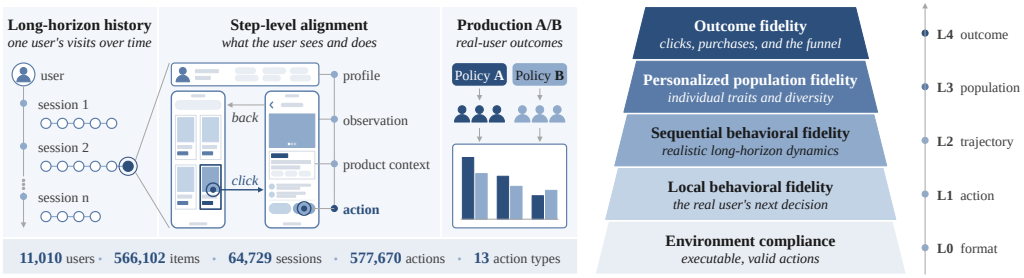


Figure 1: Overview of USERARENA. **Left:** cross-session histories, step-aligned interaction records, production A/B validation, and dataset scale. **Right:** the five-level evaluation protocol, progressing from environment compliance (L0) to outcome fidelity (L4).

To address these limitations, we introduce **USERARENA**, an interactive benchmark combining a production-grounded shopping environment with real-user trajectories recorded on one of Asia's largest e-commerce platforms (Figure 1). It contains real-user data comprising **11,010 users, 566,102 items, and 64,729 session trajectories** and covers a complete action space of 13 action types. Each trajectory aligns user profiles, visual observations, product context, and actions at every step. For long-horizon evaluation, USERARENA links sessions from the same user into cross-session trajectories. It also supports offline validation by comparing the same recommendation-policy pair in simulation with outcomes from a controlled production A/B test. We evaluate simulators at five levels: output validity (**L0**), local behavioral fidelity (**L1**), sequential behavioral fidelity (**L2**), personalization fidelity (**L3**), and outcome fidelity (**L4**). The protocol traces fidelity from output validity and local decisions to long-horizon behavior, personalization, and aggregate outcomes.

Beyond constructing the benchmark, we also address the challenge of learning faithful user simulators. Simulator design has progressed from hand-crafted and log-driven models to LLM-based models (Ie et al., 2019; Shi et al., 2019; Wang et al., 2025a; Zhang et al., 2024), yet reproducing user behavior over long rollouts remains difficult for three reasons. **(i) Behavioral miscalibration:** learned simulators can deviate from real user behavior (Zhang et al., 2024; Yoon et al., 2024). **(ii) Sparse supervision:** exact next-action targets penalize plausible alternatives and overlook trajectory-level quality (Wang et al., 2025c; Zhang et al., 2026b). **(iii) Exposure bias:** teacher forcing conditions only on ground-truth prefixes, creating a mismatch with autoregressive inference (Dai et al., 2021; Chi et al., 2022). To address these challenges, we propose **O**n-**P**olicy **T**rajectory **D**istillation (**OPTD**), which distills teacher guidance on current-policy responses at real states (Agarwal et al., 2024; Zhao et al., 2026). This on-policy supervision trains on student-generated prefixes, while trajectory-level intent guides teacher targets toward coherent long-horizon behavior.

In summary, our contributions are threefold. (1) We introduce USERARENA, a real-world e-commerce interactive benchmark pairing long-horizon user trajectories with an interactive environment and a five-level protocol spanning environment compliance, local decisions, sequential behavior, personalization, and outcomes. (2) We propose OPTD, which uses trajectory-level intent and dense teacher targets to guide student-generated rollouts toward coherent long-horizon behavior. (3) We benchmark representative text- and GUI-based simulators for effectiveness and efficiency and characterize USERARENA's behavioral composition and interaction horizons.

2 THE USERARENA BENCHMARK

2.1 DATA AND INTERACTION DESIGN

A credible e-commerce interactive benchmark requires a production-grounded environment that preserves what shoppers can observe and do. Beyond this environmental foundation, two requirements are essential. First, the benchmark should use fine-grained real-user trajectories that align user profiles, visual observations, product context, and actions while capturing behavior over long horizons. Second, its evaluation should go beyond next-action agreement to assess complete trajectories and whether simulator-based policy comparisons agree with real-user outcomes in production.

Table 1: Comparison of representative interactive recommendation benchmarks. Full Action Space covers the complete executable interaction loop. Long-Horizon denotes temporally ordered user histories spanning multiple sessions. Production A/B Validation evaluates recommendation policies offline in simulation and compares their effects with outcomes observed in production A/B tests.

Benchmark	Real Traces	Visual Context	Full Action Space	Long-Horizon	Production A/B
Qilin	✓	✗	✗	✗	✗
RL4RS	✓	✗	✗	✗	✗
OPeRA	✓	✓	✓	✗	✗
OmniBehavior	✓	✗	✗	✓	✗
USB	✓	✓	✓	✗	✗
A/B Agent	✗	✓	✓	✗	✗
USERARENA	✓	✓	✓	✓	✓

Existing resources cover these requirements partially. E-commerce datasets contain real preferences and interaction logs but rarely retain the visual context of each decision (Harper & Konstan, 2015; Hou et al., 2026; Chen et al., 2025; Wang et al., 2023). Interactive platforms enable closed-loop evaluation but depend on user dynamics whose fidelity remains insufficiently established (Ie et al., 2019; Shi et al., 2019; Zhao et al., 2023). Recent benchmarks improve behavioral grounding through visual shopping traces or longitudinal records (Wang et al., 2026; Chen et al., 2026; Tang et al., 2026; Zhang et al., 2026a), yet none combines real step-level trajectories, visual context, a full shopping action space, long-horizon cross-session behavior, and production A/B validation (Table 1).

To bridge this gap, USERARENA combines a production-grounded shopping environment, fine-grained real-user trajectories, and an evaluation protocol spanning trajectory fidelity and production validation. The environment reconstructs production recommendation-feed and item-detail pages with 13 action types, from feed navigation and product inspection to add-to-cart and purchase. It updates the visible feed by cropping the corresponding viewport from the pre-rendered full feed as the simulator scrolls. Clicking any currently exposed item, including one not clicked by the real user, opens an interactive detail page reconstructed from that item's pre-rendered views and recorded product content. Following the observation–action formulation of interactive agents (Yao et al., 2022; Zhou et al., 2024; Wang et al., 2026; Zhang et al., 2025), each recorded decision aligns the shopper's profile, visual observation, visible product context, and action. The benchmark contains **11,010 users, 64,729 sessions, 577,670 actions, and 566,102 products** and 6 user attributes. Across these trajectories, 73,717 item clicks span 41 product categories (Appendix Figure 4).

For long-horizon evaluation, USERARENA organizes 59,378 chronologically ordered sessions from 5,768 users into cross-session trajectories. They average 10.29 sessions and 86.36 actions per user, testing whether simulators preserve preferences and behavioral tendencies across repeated visits. Appendix Table 4 compares their interaction horizons with related benchmarks. For production A/B validation, USERARENA compares the same recommendation-policy pair offline in simulation with controlled production A/B outcomes.

2.2 EVALUATION PROTOCOL

As illustrated on the right side of Figure 1, USERARENA evaluates simulator fidelity at five levels of increasing scope: from whether its outputs can be parsed to whether its trajectories, personalized behavior, and session outcomes match those of real users.

We use two evaluation modes. (1) *Replay* uses the recorded state at each step of a real session to predict the logged next action. This isolates local next-action fidelity without accumulating simulator-induced state changes. (2) *Rollout* lets the simulator select actions and determine subsequent states until the session ends. It tests whether free-running simulated sessions reproduce real trajectories and aggregate outcomes. We evaluate L0 under both modes, L1 under Replay, and L2–L4 under Rollout. Appendix A.3 provides the complete definitions and computation rules.

- **L0: Environment Compliance** asks whether the environment can parse the simulator's outputs. We report valid-action rates under Replay (RVA) and Rollout (OVA), which require each output to parse as an action executable in the current state.

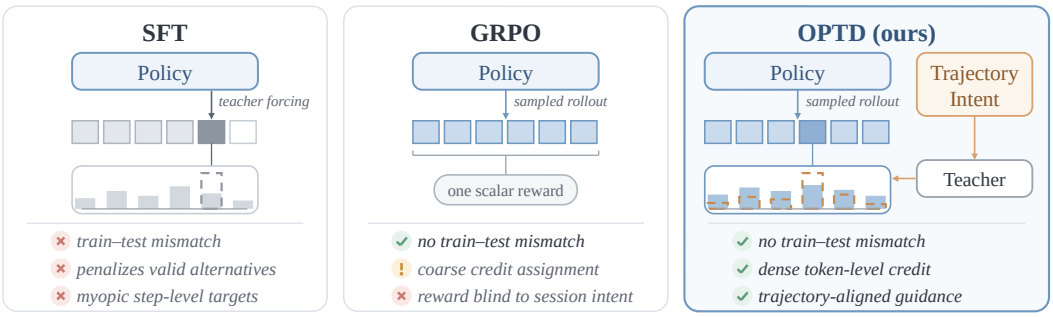


Figure 2: Supervision in SFT, GRPO, and OPTD. SFT learns from recorded responses, GRPO assigns a scalar reward to each generated response, and OPTD provides full-vocabulary teacher targets at every token of the student's own rollout.

- **L1: Local Behavioral Fidelity** asks whether the simulator and real user make the same next decision from the same state. F1 measures action-type agreement, whereas accuracy also requires the target item or comment to match when applicable.
- **L2: Sequential Behavioral Fidelity** asks whether a free-running simulator produces realistic long-horizon dynamics. Decision-mix Jensen–Shannon divergence (DM-JSD) compares next-decision distributions after item-page visits, capturing whether simulated and real users follow the same item-level decision patterns. Trajectory error (TE) compares mean action count per session.
- **L3: Personalized Population Fidelity** asks whether the simulator reproduces behavioral heterogeneity associated with users' interaction histories. Population fidelity (PF) groups users by historical activity, purchase propensity, and category breadth, then compares simulated and real behavior within each group across session dynamics, action patterns, and product preferences.
- **L4: Outcome Fidelity** asks whether the simulator reproduces aggregate behavioral and terminal outcomes. Count-based click-through rate error (ΔCTR) compares clicks per executed feed action, count-based add-to-cart rate error (ΔACR) compares add-to-cart actions per click, and terminal-state JSD (T-JSD) compares the distributions of final session outcomes.

3 On-Policy Trajectory Distillation

To model long-horizon shopping behavior in UserArena, we propose On-Policy Trajectory Distillation (OPTD). OPTD extracts a trajectory-level intent from each complete training trajectory and provides it as privileged context to a frozen teacher. The intent conditions dense token-level teacher supervision throughout each student response. A per-token trust region limits how far these targets depart from predictions based only on the student's observable state. Figure 2 contrasts the train–test mismatch and myopic targets of SFT, the coarse response-level credit assignment of GRPO, and OPTD's dense, trajectory-aligned teacher targets at every generated token.

3.1 Task Setup

Problem formulation. UserArena records each shopping session as a trajectory. At every step, the user observes a page and responds to it. Let τ_u denote the trajectory of user u, x_t the state at step t, and $y_t^\star$ the user's real response at that step. The state x_t contains the current page and interaction history before the response. The response $y_t^\star$ is a token sequence containing a short rationale followed by one structured interface action. We denote its token at position k by $y_{t,k}^\star$, its prefix before that position by $y_{t,<k}^\star$, and its token length by $|y_t^\star|$. A user simulator is a policy $\pi_\theta(y \mid x)$ that generates response y from state x, where θ denotes the trainable student parameters.

On-policy sampling. We treat π_θ as the student and optimize it with token-level guidance from a teacher on the student's own responses. In contrast, standard teacher forcing on $y_t^\star$ (Bengio et al., 2015; Ranzato et al., 2016) conditions every prediction on a ground-truth prefix. At inference, the student instead conditions on its own partial output, so errors can move it to prefixes absent from

teacher-forced training (Dai et al., 2021; Chi et al., 2022). To expose training to these self-generated prefixes, OPTD adopts on-policy distillation (Agarwal et al., 2024; Zhao et al., 2026). At each training step, we draw a real state x_t from the empirical training distribution $\mathcal{D}$ and perform a one-step rollout by sampling a complete response from the current policy.

$$y_t = (y_{t,1}, \ldots, y_{t,|y_t|}) \sim \pi_\theta(\cdot \mid x_t), \tag{1}$$

where $y_{t,k}$ is the generated token at position k, $|y_t|$ is the response length, and $y_{t,<k} = (y_{t,1}, \ldots, y_{t,k-1})$ denotes its prefix before position k. A teacher with privileged trajectory-level context scores each position of y_t, and the student learns from the resulting distributions on its own prefixes rather than only recorded prefixes. This reduces the mismatch between teacher-forced training and autoregressive inference.

3.2 Trajectory-Level Intent Supervision

To guide the student toward behavioral patterns that unfold across an entire session, we provide the teacher with a trajectory-level intent extracted from the complete training trajectory.

Intent extraction. We use a large language model as an extractor E that maps the trajectory to an abstract summary of the shopper's intent,[2]

$$z_u = E(\tau_u), \tag{2}$$

where z_u is a structured record of complementary typed anchors capturing trajectory-wide preferences, concerns, information needs, comparison criteria, and conditional decision tendencies. To keep z_u from serving as an answer key for individual steps, the extraction and validation rules exclude concrete actions, specific products and identifiers, exact prices, trajectory outcomes, future-page content, and the step indices supporting the summary. The resulting intent contextualizes the current state without specifying the recorded next action. The student never directly receives z_u or observations after x_t; the intent influences it only through the teacher's token distributions.

Intent-conditioned supervision. The teacher is a frozen supervised fine-tuned checkpoint with parameters ϕ. Let P_ϕ denote its next-token distribution and let $v \in \mathcal{V}$ be a candidate token in vocabulary $\mathcal{V}$. Given x_t, z_u, and the student prefix $y_{t,<k}$, the teacher probability of token v is

$$q^{\mathrm{I}}_{t,k}(v) = P_\phi(v \mid x_t, z_u, y_{t,<k}), \tag{3}$$

The student's distribution at the same position is $p^\theta_{t,k}(v) = \pi_\theta(v \mid x_t, y_{t,<k})$. The intent-conditioned distribution $q^{\mathrm{I}}_{t,k}$ provides one endpoint for the regularized target constructed in Section 3.3. The teacher outputs are treated as constants, and no gradient reaches ϕ.

3.3 Trust-Region Regularization

To avoid training the student to imitate predictions driven too strongly by training-only intent, we constrain how much the intent may alter the teacher's distribution at each token. The trust region keeps intent guidance within a fixed divergence budget from a reference conditioned only on information available to the student.

Unconditioned reference. We construct the reference by querying the same frozen teacher on the same state and student prefix, but without z_u,

$$q^0_{t,k}(v) = P_\phi(v \mid x_t, y_{t,<k}). \tag{4}$$

Because the teacher queries differ only in z_u, their gap isolates the effect of intent conditioning.

Per-token trust region. To bound that gap, we interpolate the two distributions in log-probability space, where $\lambda \in [0, 1]$ is the interpolation coefficient and v' ranges over the vocabulary $\mathcal{V}$:

$$q^\lambda_{t,k}(v) = \frac{\left[q^0_{t,k}(v)\right]^{1-\lambda}\left[q^{\mathrm{I}}_{t,k}(v)\right]^{\lambda}}{\sum_{v' \in \mathcal{V}}\left[q^0_{t,k}(v')\right]^{1-\lambda}\left[q^{\mathrm{I}}_{t,k}(v')\right]^{\lambda}} = \mathrm{softmax}\Big(\log q^0_{t,k} + \lambda\big(\log q^{\mathrm{I}}_{t,k} - \log q^0_{t,k}\big)\Big), \tag{5}$$

[2] We use Qwen3.5-397B as the extractor.

Setting $\lambda = 0$ recovers the unconditioned distribution and $\lambda = 1$ the fully intent-conditioned one. Let $\lambda_{\max} \in [0, 1]$ be the maximum interpolation coefficient, let $\epsilon \geq 0$ be the divergence budget, and let $D_{\mathrm{KL}}(P\|Q)$ denote the Kullback–Leibler divergence from distribution P to distribution Q. At each position, we select the largest interpolation coefficient whose target remains within the budget:

$$\lambda^{\star}_{t,k} = \max\left\{\lambda \in [0, \lambda_{\max}] \;:\; D_{\mathrm{KL}}\big(q^{\lambda}_{t,k} \,\big\|\, q^{0}_{t,k}\big) \leq \epsilon\right\}. \tag{6}$$

Along this geometric path, $D_{\mathrm{KL}}(q^{\lambda}_{t,k}\|q^{0}_{t,k})$ is nondecreasing in λ, making the feasible set an interval and allowing $\lambda^{\star}_{t,k}$ to be located by bisection. The regularized target $q^{\lambda^{\star}_{t,k}}_{t,k}$ replaces $q^{\mathrm{I}}_{t,k}$ as the distribution the student matches.

The constraint adapts intent guidance at each token. When the intent-conditioned target is feasible, the maximal permitted guidance is retained. Otherwise, $\lambda^{\star}_{t,k}$ moves the target toward $q^{0}_{t,k}$ until it satisfies the budget. Thus, ϵ bounds the token-level distributional change attributed to the intent.

3.4 Training Objective

The OPTD objective combines a distillation term that transfers trajectory-level intent through the teacher's distributions on student responses with a grounding term that provides step-level supervision from recorded responses. Let $\mathcal{L}_{\mathrm{dist}}$ denote the distillation loss, let $\mathcal{L}_{\mathrm{GT}}$ denote the grounding loss, let $\gamma \geq 0$ be their relative weight, and let $\mathcal{L}$ denote the total loss. The terms are computed on the same sampled states and combined as

$$\mathcal{L} = \mathcal{L}_{\mathrm{dist}} + \gamma\, \mathcal{L}_{\mathrm{GT}}, \tag{7}$$

Distillation term. Exact-match objectives credit one token sequence but no alternative phrasing or behaviorally equivalent action a real shopper might have produced (Wang et al., 2025c; Zhang et al., 2026b). OPTD instead matches the regularized target $q_{t,k} = q^{\lambda^{\star}_{t,k}}_{t,k}$ at every generated token using the generalized Jensen–Shannon divergence (Agarwal et al., 2024) with mixing weight $\beta \in (0, 1)$ and mixture $m_{t,k} = \beta\, q_{t,k} + (1-\beta)\, p^{\theta}_{t,k}$:

$$\mathrm{JSD}_{\beta}\big(q_{t,k} \,\big\|\, p^{\theta}_{t,k}\big) = \beta\, D_{\mathrm{KL}}\big(q_{t,k} \,\big\|\, m_{t,k}\big) + (1-\beta)\, D_{\mathrm{KL}}\big(p^{\theta}_{t,k} \,\big\|\, m_{t,k}\big), \tag{8}$$

Using $\mathbb{E}$ for expectation over training states x_t and student responses y_t, the distillation loss is

$$\mathcal{L}_{\mathrm{dist}} = \mathbb{E}_{x_t \sim \mathcal{D},\; y_t \sim \pi_{\theta}(\cdot|x_t)}\left[\frac{1}{|y_t|}\sum_{k=1}^{|y_t|} \mathrm{JSD}_{\beta}\big(q_{t,k} \,\big\|\, p^{\theta}_{t,k}\big)\right]. \tag{9}$$

The inner mean averages over generated tokens. Because JSD_{β} is bounded, large student–target disagreement cannot produce an unbounded per-token loss.

Grounding term. The distillation loss conveys trajectory-level teacher guidance, whereas the recorded response $y^{\star}_t$ provides a step-level target at state x_t. We apply a teacher-forced loss:

$$\mathcal{L}_{\mathrm{GT}} = \mathbb{E}_{(x_t,\, y^{\star}_t)\sim\mathcal{D}}\left[-\frac{1}{|y^{\star}_t|}\sum_{k=1}^{|y^{\star}_t|} \log \pi_{\theta}\big(y^{\star}_{t,k} \mid x_t,\, y^{\star}_{t,<k}\big)\right]. \tag{10}$$

This term supervises every token of $y^{\star}_t$ in its original format, including the rationale, action, and terminating token. Together, the losses combine session-wide intent with state-level behavior.

4 Experiments

4.1 Experimental Setup

We compare OPTD against two families of user simulators, grouped by their observation modality. Text-based simulators observe textual page representations and include RecAgent (Wang et al., 2025a), Agent4Rec (Zhang et al., 2024), Shop-R1 (Zhang et al., 2026b), Customer-R1 (Wang et al., 2025c), AlignUSER (Bougie et al., 2026), and SimUSER (Bougie & Watanabe, 2025). GUI-based simulators instead observe page screenshots and include A/B Agent (Zhang et al., 2026a) and STA (Zhang et al., 2025). All methods in the main comparison use Qwen3.5-2B, enabling a controlled comparison of prompting, memory, observation modality, and post-training strategy without model-scale confounding. Additional baseline, data-processing, implementation, and hyperparameter details are provided in Appendix B.

Table 2: Single- and cross-session results on USERARENA across L0–L4. For L1–L4, bold and underlining denote the best and second-best results, respectively, within each setting.

		L0		L1		L2		L3	L4		
	Method	RVA ↑	OVA ↑	F1 ↑	Acc. ↑	DM-JSD ↓	TE ↓	PF ↑	ΔCTR ↓	ΔACR ↓	T-JSD ↓
	Single-session										
Text	RecAgent	0.9834	0.9965	0.1424	0.1507	0.2928	3.63	0.6941	0.6162	1.0642	0.1287
	Agent4Rec	0.9958	0.8192	0.1274	0.1222	0.2054	6.05	0.7393	0.4737	0.5086	0.4196
	SimUSER	0.9833	0.9651	0.1411	0.1680	0.1140	9.27	0.7230	0.2559	0.4897	<u>0.0664</u>
	Shop-R1	0.9227	0.9551	0.3419	0.1899	0.1582	3.72	0.7251	0.0283	0.3623	0.2722
	Customer-R1	0.9673	0.9866	<u>0.4108</u>	<u>0.3114</u>	0.1954	2.86	0.7382	<u>0.0227</u>	0.3997	0.2810
	AlignUSER	0.9902	0.9987	0.3276	0.2504	0.1973	6.47	0.6424	0.1049	0.3739	0.3191
GUI	A/B Agent	0.4113	0.6302	0.1017	0.0157	0.6699	9.68	0.6459	0.0291	0.2039	0.8205
	STA	0.9856	0.9768	0.4030	0.2899	<u>0.0245</u>	<u>1.77</u>	<u>0.7958</u>	0.1080	<u>0.1880</u>	0.1519
	OPTD (**ours**)	0.9773	0.9712	**0.4178**	**0.3204**	**0.0156**	**1.72**	**0.8692**	**0.0026**	**0.0430**	**0.0387**
	Cross-session										
Text	RecAgent	0.9930	0.9961	0.1032	0.1747	0.2157	7.66	0.6621	0.5521	0.7428	0.1268
	Agent4Rec	0.9943	0.8430	0.1294	0.1455	0.1541	1.57	0.7657	0.4098	1.1625	0.4424
	SimUSER	0.9860	0.9628	0.0946	0.1629	0.0387	3.51	0.7492	0.1872	0.3502	0.0275
	Shop-R1	0.9476	0.9920	<u>0.3533</u>	0.1975	0.1310	1.35	0.8739	0.0905	0.3514	0.0941
	Customer-R1	0.9531	0.9871	0.3415	0.1902	0.0574	1.91	0.8628	0.0407	0.1515	0.0244
	AlignUSER	0.9545	0.9864	0.3172	0.1933	<u>0.0370</u>	2.11	0.8732	0.0485	0.1461	<u>0.0099</u>
GUI	A/B Agent	0.5182	0.5777	0.1094	0.0265	0.5709	4.17	0.6917	0.0615	**0.0096**	0.7980
	STA	0.9563	0.9830	**0.3556**	<u>0.2919</u>	0.0465	<u>0.81</u>	<u>0.8911</u>	**0.0099**	0.1577	0.0173
	OPTD (**ours**)	0.9578	0.9792	0.3456	**0.2969**	**0.0284**	**0.72**	**0.8929**	<u>0.0357</u>	<u>0.1399</u>	**0.0077**

4.2 MAIN RESULTS

We evaluate every simulator in two settings. The *single-session* setting evaluates each user's trajectory within one session. The *cross-session* setting evaluates a chronologically ordered trajectory spanning multiple sessions from the same user, emphasizing long-horizon behavioral prediction across visits. We report both settings in Table 2.

Single-session results. The single-session block of Table 2 shows that OPTD ranks first on all eight reported L1–L4 metrics, spanning local and sequential behavioral fidelity, personalized population fidelity, and outcome fidelity. This broad improvement aligns with its training design: trajectory-level intent guides token-level targets on the student's own rollout distribution, while the trust region and grounding term keep this guidance close to behavior supported by observable states and recorded responses. Customer-R1's profile-conditioned interaction history is consistent with its strong local prediction, while STA's multimodal state context and reinforcement-learning-tuned checkpoint align with its strong sequential and population-level fidelity. A/B Agent remains competitive on ACR error, but its lower Replay and Rollout valid-action rates suggest less reliable transfer to the fixed USERARENA action protocol, limiting downstream fidelity.

Cross-session results. The cross-session block of Table 2 uses the same summary metrics. Among all methods, OPTD ranks first on five of the eight L1–L4 metrics and second on two more. It achieves the strongest L2 sequential behavioral fidelity, with the lowest decision-mix divergence and trajectory-length error, together with the highest L3 personalized population fidelity. These results demonstrate OPTD's gains in preserving long-horizon trajectory structure and personalized behavioral patterns across sessions. Its leading L1 accuracy separately indicates the strongest step-wise action prediction under Replay. STA leads macro-F1 and CTR error, suggesting that its reinforcement-learning post-training benefits selected local and outcome-level metrics, although these gains are not consistent across fidelity levels.

4.3 Production A/B Validation

To assess whether simulator-based policy comparisons reflect real-user responses to recommendation changes, we conduct an offline validation against outcomes recorded for the control and treatment arms of a production A/B test. We evaluate the same pair of ranking policies in each simulator and compare the simulated policy effect with the recorded production effect. In simulation, we assign a frozen pool of 10,000 users evenly to the two policy arms and hold the users and product catalog fixed across all simulators. Because the ranking policies alter item exposure, we measure their effect using the between-arm difference in page-view click-through rate (ΔPV-CTR). The effect ratio divides the simulated ΔPV-CTR by its online value, and the effect error is the absolute deviation of this ratio from 1. Among the evaluated user simulators, OPTD most closely reproduces the online policy effect, reaching an effect ratio of $0.713\times$. GPT-6-Sol and GPT-6-Luna instead overestimate the online response, indicating that general-purpose model capacity does not directly provide behavioral calibration in this setting. Full experimental details appear in Appendix C.1.

Table 3: Production A/B validation of the same ranking-policy pair.

Simulator	ΔPV-CTR ratio	Effect error $\downarrow$
Shop-R1	$0.162\times$	0.838
Customer-R1	$0.191\times$	0.809
AlignUSER	$0.020\times$	0.980
STA	$0.297\times$	0.703
OPTD	$\mathbf{0.713}\times$	**0.287**
Qwen3.5-2B	$0.004\times$	0.996
GPT-6-Sol	$1.576\times$	0.576
GPT-6-Luna	$2.154\times$	1.154

4.4 In-Depth Analyses

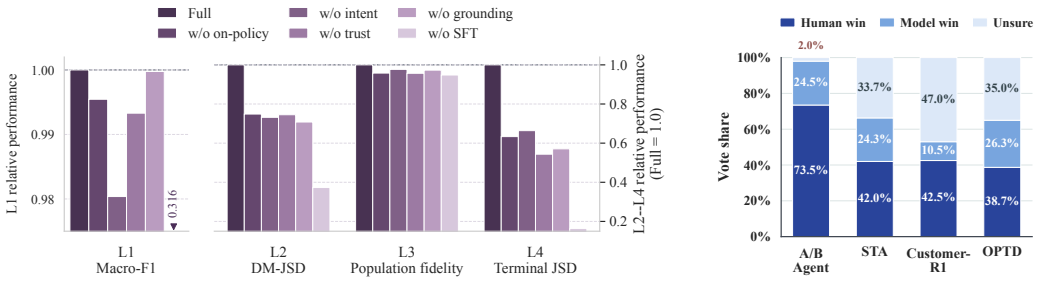
We further examine OPTD through component ablations and a Turing-style human evaluation. Appendix C provides an efficiency analysis and qualitative case studies.

Ablation study. To understand the contribution of each component within OPTD, we construct five ablations with all other settings fixed. (1) *w/o on-policy* replaces student-generated responses with logged responses, so distillation operates on recorded prefixes rather than the student's own prefixes. (2) *w/o intent* removes trajectory-level intent from the teacher context. (3) *w/o trust region* uses the fully intent-conditioned teacher distribution without trust-region interpolation. (4) *w/o grounding* sets the grounding weight to zero. (5) *w/o SFT* initializes the teacher and student from the base model. Figure 3(a) reports one representative metric from each behavioral fidelity level L1–L4. First, removing any component reduces performance on at least one selected metric, showing that each contributes to behavioral fidelity. Second, the *w/o SFT* variant degrades most markedly across the selected metrics, indicating that the warm-up establishes the instruction-following and action-generation abilities required for subsequent trajectory distillation.

Turing-style human evaluation. Automatic metrics quantify agreement with recorded behavior, but they do not directly test whether a complete simulated trajectory appears human to an observer. We therefore define a source-blinded Turing-style evaluation (Turing, 1950) that compares OPTD with the text-based Customer-R1 (Wang et al., 2025c) and two GUI-based simulators, STA (Zhang et al., 2025) and A/B Agent (Zhang et al., 2026a). For each user context, three annotators independently compare an anonymized, order-randomized real–simulated pair and identify the machine-generated trajectory or choose *Unsure*. Each simulator is evaluated on 200 trajectory pairs sampled from the held-out test set. Figure 3(b) shows that OPTD has the lowest Human win rate and the highest Model win rate among the four simulators. Appendix C.4 provides the complete protocol and experiment analysis.

5 Related Work

Interactive recommendation benchmarks. Recommendation benchmarks have evolved from static preference records toward richer behavioral data and interactive evaluation. MovieLens (Harper & Konstan, 2015) records movie ratings, Amazon Reviews (Hou et al., 2026) pairs product ratings with review text, Qilin (Chen et al., 2025) adds multimodal app-level sessions, and RL4RS (Wang et al., 2023) provides real-world sequential interactions. Agent-oriented benchmarks further broaden the task interface: RecBench+ (Huang et al., 2026) evaluates personalized recommendation through natural-language requests, AgentRecBench (Shang et al., 2025) introduces



(a) Relative performance by fidelity level. **(b)** Turing-style human evaluation.

Figure 3: Ablation and human-evaluation results. (a) Component performance relative to the full model. (b) Human win denotes correct identification of the simulated trajectory, Model win denotes misidentification of the real trajectory as simulated, and Unsure denotes abstention.

interactive recommendation scenarios, and OPeRA (Wang et al., 2026) and OmniBehavior (Chen et al., 2026) provide richer evidence for human behavior simulation. Simulation platforms such as RecSim (Ie et al., 2019), Virtual-Taobao (Shi et al., 2019), and KuaiSim (Zhao et al., 2023) further support interactive evaluation across temporal scales. Yet existing resources rarely record, throughout long-horizon sessions, what real users see and how they act, or test whether simulator-based comparisons reproduce policy effects observed in production A/B tests. UserArena provides both forms of evidence through step-aligned real-world shopping trajectories and offline validation against outcomes from a controlled production A/B test.

User simulator development. Early systems represented users through configurable or parameterized dynamics, including RecSim (Ie et al., 2019), RecoGym (Rohde et al., 2018), and the probabilistic RecSim NG framework (Mladenov et al., 2021). Log-driven approaches then learned interaction dynamics from observed behavior: Virtual-Taobao (Shi et al., 2019) combines generative customer modeling with adversarial imitation, while the Generative Adversarial User Model (Chen et al., 2019) learns user dynamics and rewards for model-based policy training. IRecGAN (Bai et al., 2019) incorporates generated trajectories into adversarial user-model learning. More recent LLM-based simulators, including RecAgent (Wang et al., 2025a), Agent4Rec (Zhang et al., 2024), and SUBER (Corecco et al., 2024), increase the expressiveness of profile-conditioned behavior, while Customer-R1 (Wang et al., 2025c), Shop-R1 (Zhang et al., 2026b), and RecVerse (Tang et al., 2026) apply reinforcement learning to shopping simulation. Despite this progress, two optimization challenges remain. **(1) Sparse rewards:** most methods use exact-match rewards, treating plausible alternatives and invalid behaviors alike. **(2) Exposure bias:** teacher forcing trains only on ground-truth prefixes, leaving models unprepared for self-generated prefixes and vulnerable to compounding errors at inference. OPTD addresses both through dense distributional supervision on student-sampled responses and trajectory-level intent for long-horizon behavioral guidance.

6 Conclusion

We introduced UserArena, an interactive benchmark combining a production-grounded shopping environment with fine-grained, long-horizon real-user trajectories. We also developed a five-level protocol that evaluates simulator quality from environment compliance and local decisions to sequential, personalized population, and outcome fidelity. For this long-horizon setting, we proposed On-Policy Trajectory Distillation (OPTD), which uses trajectory-level intent to guide dense teacher supervision on the student's own rollouts. OPTD achieves the strongest overall fidelity under this protocol and most closely reproduces the production policy effect in the controlled comparison.

As future work we will develop a leaderboard platform around UserArena to support standardized and continuously updated evaluation of user simulators. We will investigate stronger long-horizon simulators that preserve evolving preferences and behavioral consistency across sessions.

AI USE STATEMENT

We used generative AI tools to edit the manuscript for clarity and readability. We also used them to assist with translating Chinese interface text and with translating and shortening Chinese item titles for presentation in figures and qualitative examples. These presentation edits did not modify the underlying trajectories, annotations, evaluation labels, or experimental results. The authors manually reviewed all AI-assisted text and translations against the original Chinese materials and take responsibility for the final content of the paper.

REPRODUCIBILITY STATEMENT

We support reproducibility by describing the benchmark construction and evaluation protocol in Section 2, the complete OPTD formulation in Section 3, and the experimental setup in Section 4. Appendix A provides dataset statistics and metric definitions, while Appendix B documents data processing, baseline configurations, implementation details, and prompt structure. Additional analyses are reported in Appendix C.

ETHICS AND DATA STATEMENT

The dataset was derived from de-identified interaction data collected during normal platform operation. Its use for research was authorized in accordance with the platform's privacy policy and internal data-governance procedures. We selected sessions and their associated item pools from user behavior logs and applied data cleaning and de-identification before release. Product metadata contain only product titles and English top-level category labels. We removed information that could directly identify users and replaced production identifiers with non-linkable sequential identifiers.

The released interface images contain no GUI screenshots captured from real users. We reconstruct the two-column feed and item-detail pages in an offline React environment using the de-identified sessions and item pools, and then batch-render the resulting interfaces. The dataset is released for non-commercial research under a data use agreement. The agreement prohibits attempts to re-identify users, linkage with external personal data, and unauthorized redistribution.

REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *International Conference on Learning Representations*, 2024.

Xueying Bai, Jian Guan, and Hongning Wang. A model-based reinforcement learning with adversarial training for online recommendation. In *Advances in Neural Information Processing Systems*, volume 32, pp. 10734–10745, 2019.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems*, volume 28, pp. 1171–1179, 2015.

Nicolas Bougie and Narimawa Watanabe. SimUSER: Simulating user behavior with large language models for recommender system evaluation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 6: Industry Track)*, pp. 43–60, 2025. doi: 10.18653/v1/2025.acl-industry.5.

Nicolas Bougie, Gian Maria Marconi, Xiaotong Ye, and Narimawa Watanabe. AlignUSER: Human-aligned LLM agents via world models for recommender system evaluation. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 16433–16451, 2026. doi: 10.18653/v1/2026.acl-long.747.

Jia Chen, Qian Dong, Haitao Li, Xiaohui He, Yan Gao, Shaosheng Cao, Yi Wu, Ping Yang, Chen Xu, Yao Hu, Qingyao Ai, and Yiqun Liu. Qilin: A multimodal information retrieval dataset with APP-level user sessions. In *Proceedings of the 48th International ACM SIGIR Conference*

on Research and Development in Information Retrieval, pp. 3670–3680, 2025. doi: 10.1145/3726302.3730279.

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

Xinshi Chen, Shuang Li, Hui Li, Shaohua Jiang, Yuan Qi, and Le Song. Generative adversarial user model for reinforcement learning based recommendation system. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 1052–1061, 2019.

Dafeng Chi, Yuzheng Zhuang, Yao Mu, Bin Wang, Jianzhu Bao, Yasheng Wang, Yuhan Dong, Xin Jiang, Qun Liu, and Jianye Hao. Offline-to-online co-evolutional user simulator and dialogue system. In *Proceedings of the Towards Semi-Supervised and Reinforced Task-Oriented Dialog Systems*, pp. 98–113, 2022. doi: 10.18653/v1/2022.seretod-1.11.

Nathan Corecco, Giorgio Piatti, Luca A. Lanzendörfer, Flint Xiaofeng Fan, and Roger Wattenhofer. SUBER: An RL environment with simulated human behavior for recommender systems. In *ECAI 2024*, volume 392, pp. 2210–2217, 2024. doi: 10.3233/FAIA240742.

Xinyi Dai, Jianghao Lin, Weinan Zhang, Shuai Li, Weiwen Liu, Ruiming Tang, Xiuqiang He, Jianye Hao, Jun Wang, and Yong Yu. An adversarial imitation click model for information retrieval. In *Proceedings of the Web Conference 2021*, pp. 1809–1820, 2021. doi: 10.1145/3442381.3449913.

Joseph L. Fleiss. Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76(5):378–382, 1971. doi: 10.1037/h0031619.

F. Maxwell Harper and Joseph A. Konstan. The MovieLens Datasets: History and context. *ACM Transactions on Interactive Intelligent Systems*, 5(4):1–19, 2015. doi: 10.1145/2827872.

Yupeng Hou, Jiacheng Li, Xiangjun Fu, Zhankui He, An Yan, Xiusi Chen, and Julian McAuley. Bridging language and items for retrieval and recommendation: Benchmarking LLMs as semantic encoders. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics*, pp. 3251–3265, 2026. doi: 10.18653/v1/2026.acl-long.147.

Jiani Huang, Shijie Wang, Liangbo Ning, Wenqi Fan, Shuaiqiang Wang, Dawei Yin, and Qing Li. Towards next-generation recommender systems: A benchmark for personalized recommendation assistant with LLMs. In *Proceedings of the Nineteenth ACM International Conference on Web Search and Data Mining*, pp. 217–226, 2026. doi: 10.1145/3773966.3777954.

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.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, pp. 611–626, 2023. doi: 10.1145/3600006.3613165.

Fei Liu, Xinyu Lin, Hanchao Yu, Mingyuan Wu, Jianyu Wang, Qiang Zhang, Zhuokai Zhao, Yinglong Xia, Yao Zhang, Weiwei Li, Mingze Gao, Qifan Wang, Lizhu Zhang, Benyu Zhang, and Xiangjun Fan. RecoWorld: Building simulated environments for agentic recommender systems. In *Companion Proceedings of the ACM Web Conference 2026*, pp. 650–659, 2026. doi: 10.1145/3774905.3794661.

Martin Mladenov, Chih-Wei Hsu, Vihan Jain, Eugene Ie, Christopher Colby, Nicolas Mayoraz, Hubert Pham, Dustin Tran, Ivan Vendrov, and Craig Boutilier. RecSim NG: Toward principled uncertainty modeling for recommender ecosystems. *arXiv preprint arXiv:2103.08057*, 2021.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. Sequence level training with recurrent neural networks. In *International Conference on Learning Representations*, 2016.

David Rohde, Stephen Bonner, Travis Dunlop, Flavian Vasile, and Alexandros Karatzoglou. RecoGym: A reinforcement learning environment for the problem of product recommendation in online advertising. *arXiv preprint arXiv:1808.00720*, 2018.

Yu Shang, Peijie Liu, Yuwei Yan, Zijing Wu, Leheng Sheng, Yuanqing Yu, Chumeng Jiang, An Zhang, Fengli Xu, Yu Wang, Min Zhang, and Yong Li. AgentRecBench: Benchmarking LLM agent-based personalized recommender systems. In *Advances in Neural Information Processing Systems*, volume 38, pp. 171137–171161, 2025.

Jing-Cheng Shi, Yang Yu, Qing Da, Shi-Yong Chen, and An-Xiang Zeng. Virtual-Taobao: Virtualizing real-world online retail environment for reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 4902–4909, 2019. doi: 10.1609/aaai.v33i01.33014902.

Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. Megatron-LM: Training multi-billion parameter language models using model parallelism. *arXiv preprint arXiv:1909.08053*, 2019.

Jiakai Tang, Yan Mi, Jing Yu, Yang Zhang, See-Kiong Ng, Qi Cao, Fei Sun, Xu Chen, Wen Chen, Jian Wu, et al. Towards faithful simulation of human shopping behavior. *arXiv preprint arXiv:2608.20707*, 2026.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL `https://qwen.ai/blog?id=qwen3.5`.

Alan M. Turing. Computing machinery and intelligence. *Mind*, 59(236):433–460, 1950. doi: 10.1093/mind/LIX.236.433.

Kai Wang, Zhene Zou, Minghao Zhao, Qilin Deng, Yue Shang, Yile Liang, Runze Wu, Xudong Shen, Tangjie Lyu, and Changjie Fan. RL4RS: A real-world dataset for reinforcement learning based recommender system. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 2935–2944, 2023. doi: 10.1145/3539618.3591899.

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 Transactions on Information Systems*, 43(2):1–37, 2025a. doi: 10.1145/3708985.

Weixun Wang, Shaopan Xiong, Gengru Chen, Wei Gao, Sheng Guo, Yancheng He, Ju Huang, Jiaheng Liu, Zhendong Li, Xiaoyang Li, et al. Reinforcement learning optimization for large-scale learning: An efficient and user-friendly scaling library. *arXiv preprint arXiv:2506.06122*, 2025b.

Ziyi Wang, Yuxuan Lu, Yimeng Zhang, Jing Huang, and Dakuo Wang. Customer-R1: Personalized simulation of human behaviors via RL-based LLM agent in online shopping. *arXiv preprint arXiv:2510.07230*, 2025c.

Ziyi Wang, Yuxuan Lu, Wenbo Li, Amirali Amini, Bo Sun, Yakov Bart, Weimin Lyu, Jiri Gesi, Tian Wang, Jing Huang, Yu Su, Upol Ehsan, Malihe Alikhani, Toby Jia-Jun Li, Lydia Chilton, and Dakuo Wang. OPeRA: A dataset of observation, persona, rationale, and action for evaluating LLMs on human online shopping behavior simulation. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics*, pp. 43942–43960, 2026. doi: 10.18653/v1/2026.acl-long.2033.

Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. WebShop: Towards scalable real-world web interaction with grounded language agents. In *Advances in Neural Information Processing Systems*, volume 35, pp. 20744–20757, 2022.

Se-eun Yoon, Zhankui He, Jessica Echterhoff, and Julian McAuley. Evaluating large language models as generative user simulators for conversational recommendation. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1490–1504, 2024. doi: 10.18653/v1/2024.naacl-long.83.

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, 2024. doi: 10.1145/3626772.3657844.

Wenlin Zhang, Xiangyang Li, Qiyuan Ge, Kuicai Dong, Pengyue Jia, Xiaopeng Li, Zijian Zhang, Maolin Wang, Yichao Wang, Huifeng Guo, Ruiming Tang, and Xiangyu Zhao. Exploring recommender system evaluation: A multi-modal LLM agent framework for A/B testing. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 2878–2889, 2026a. doi: 10.1145/3770854.3785688.

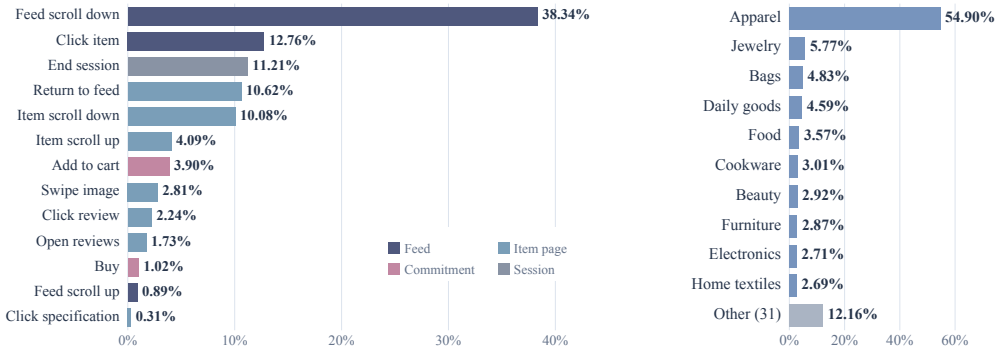
Yimeng Zhang, Jiri Gesi, Ran Xue, Tian Wang, Ziyi Wang, Yuxuan Lu, Sinong Simon Zhan, Huimin Zeng, Qingjun Cui, Yufan Guo, Jing Huang, Mubarak Shah, and Dakuo Wang. See, think, act: Online shopper behavior simulation with VLM agents. *arXiv preprint arXiv:2510.19245*, 2025.

Yimeng Zhang, Tian Wang, Jiri Gesi, Ziyi Wang, Yuxuan Lu, Jiacheng Lin, Sinong Zhan, Vianne Gao, Ruochen Jiao, Junze Liu, Kun Qian, Yuxin Tang, Ran Xue, Houyu Zhang, Qingjun Cui, Yufan Guo, and Dakuo Wang. Shop-R1: Rewarding LLMs to simulate human behavior in online shopping via reinforcement learning. In *International Conference on Learning Representations*, 2026b.

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

Siyan Zhao, Zhihui Xie, Mengchen Liu, Jing Huang, Guan Pang, Feiyu Chen, and Aditya Grover. Self-distilled reasoner: On-policy self-distillation for large language models. *arXiv preprint arXiv:2601.18734*, 2026.

Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. WebArena: A realistic web environment for building autonomous agents. In *International Conference on Learning Representations*, 2024.



(a) Decision-action distribution. **(b) Clicked item categories.**

Figure 4: Behavioral composition of USERARENA. (a) Distribution of decision actions, grouped into feed, item-page, commitment, and session actions. (b) Distribution of clicked items across product categories, with Other aggregating the remaining 31 categories.

A ADDITIONAL BENCHMARK DETAILS

A.1 COMPARISON OF INTERACTION HORIZONS

Table 1 provides a qualitative comparison of benchmark capabilities. To further substantiate its long-horizon dimension, Table 4 quantifies the interaction horizons of real-user shopping benchmarks with a full action space. OPeRA (Wang et al., 2026) and USB (Tang et al., 2026) evaluate behavior within individual shopping sessions, whereas USERARENA captures longitudinal user trajectories spanning multiple natural and consecutive shopping sessions. A/B Agent is excluded because its interaction trajectories are synthetically generated rather than collected from real users.

Table 4: Interaction-horizon statistics for real-user shopping benchmarks with a full action space. For UserArena, statistics are reported for the cross-session subset.

Benchmark	# Users	# Sessions	Trajectory Scope	Avg. Sessions	Avg. Actions
OPeRA	51	527	Single-session	1.00	11.11
USB	5,222	5,274	Single-session	1.00	13.24
USERARENA	**5,768**	**59,378**	**Cross-session**	**10.29**	**86.36**

A.2 DATASET AND BEHAVIOR STATISTICS

We provide additional analyses of USERARENA's behavioral composition, interaction horizons, and coverage.

Action and item composition. Figure 4(a) reports the proportion of every action type in USERARENA. Feed-page downward scrolling is the most frequent action at 38.34%, followed by item clicks at 12.76%, session termination at 11.21%, returns to the feed at 10.62%, and item-page downward scrolling at 10.08%. Add-to-cart and purchase actions account for 3.90% and 1.02% of decisions, respectively. This imbalance supports using macro-averaged local metrics and distributional trajectory metrics rather than allowing frequent navigation actions to dominate evaluation. Figure 4(b) shows that the 73,717 item clicks span 41 product categories. Apparel accounts for 54.90% of clicks, while the remaining 45.10% covers 40 categories, including jewelry, bags, daily goods, food, cookware, beauty, furniture, electronics, and home textiles.

Session and user horizons. Figure 5(a) shows that individual sessions contain a median of 14 actions in the single-session setting and 8 actions in the cross-session setting. Cross-session user trajectories become substantially longer by linking repeated visits: each user contributes 4–15 chronologically ordered sessions, with 10.29 sessions and 86.36 actions on average (Figure 5(b)).

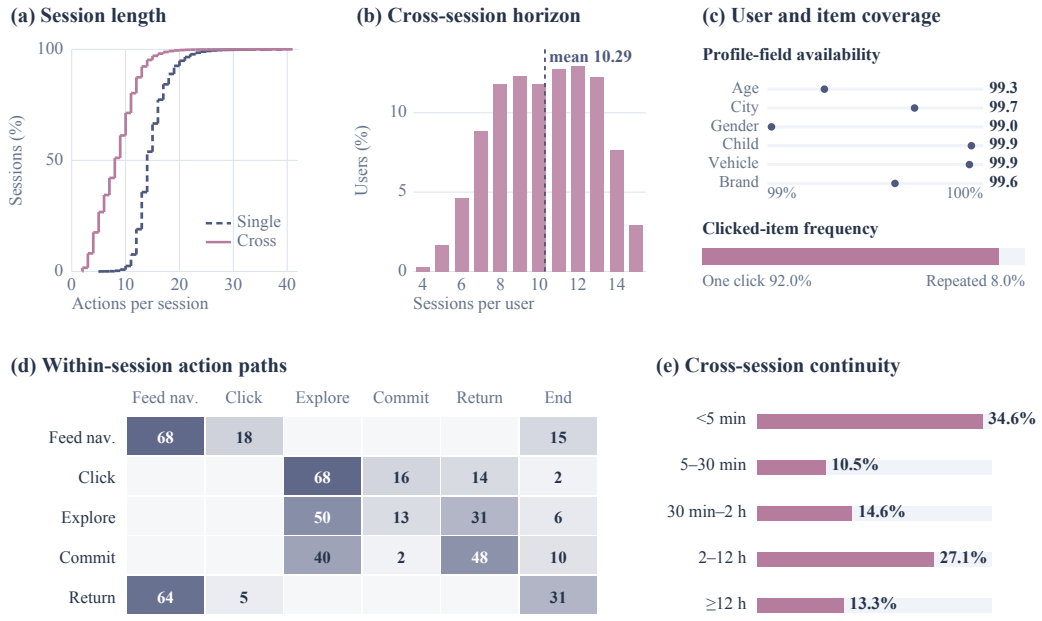


Figure 5: Additional statistics for USERARENA. (a) Empirical cumulative distributions of session length in the single- and cross-session settings. (b) Distribution of sessions per cross-session user. (c) User-profile coverage and click frequency across distinct items. (d) Row-normalized within-session transitions between semantic action groups. (e) Time gaps between adjacent cross-session pairs, excluding overlapping sessions.

User and item coverage. Figure 5(c) shows that availability across the six user-profile fields ranges from 99.0% to 99.9%, while 92.0% of the 67,267 clicked items appear only once. The trajectories cover 566,102 products overall, and the most-clicked 1% of items account for only 3.23% of clicks, indicating that interactions extend well beyond a small set of repeatedly observed products.

Behavioral paths. Figure 5(d) aggregates adjacent actions into six semantic groups and reports row-normalized transition probabilities. The resulting paths expose repeated feed navigation, item exploration after item clicks, and returns to the feed after item-page decisions. These transitions capture multi-step shopping processes that cannot be represented by isolated action frequencies.

Cross-session continuity. The cross-session subset yields 53,610 adjacent session pairs from the same users. As shown in Figure 5(e), successive sessions span intervals from minutes to more than 12 hours, and 57.8% revisit a product category present in the preceding session. The benchmark therefore captures both rapid returns and longer-term revisits while preserving each user's chronological session order.

A.3 EVALUATION METRIC DETAILS

Section 2 introduces the five-level evaluation protocol and its metrics. Here we specify how each metric reported in Table 2 is computed. We first establish notation shared across the definitions below. The symbol $\mathcal{R}$ denotes logged human sessions and $\mathcal{S}$ denotes simulator-generated sessions. Replay predicts one next action at each logged human state, while rollout generates complete sessions. For a finite set Z, $|Z|$ is its number of elements.

A.3.1 L0: Environment Compliance (Replay and Rollout)

Replay and Rollout valid-action rates (RVA and OVA). Let mode $m \in \{\mathrm{R}, \mathrm{O}\}$ denote Replay or Rollout, and let N_m denote its number of model outputs. For output j, $V_{m,j}$ denotes 1 when the output parses into an action that can execute in the observed state, including successful target resolution, and 0 otherwise. Then

$$\mathrm{VA}_m = \frac{1}{N_m} \sum_{j=1}^{N_m} V_{m,j}. \tag{11}$$

RVA is VA_R and OVA is VA_O. Every output remains in the denominator. An unparseable output or fallback substitution has $V_{m,j} = 0$.

A.3.2 L1: Local Behavioral Fidelity (Replay)

Macro-F1 (F1). At replay step i, y_i is the logged action type and $\hat{y}_i$ is the predicted type. An unparseable or invalid prediction is an incorrect empty prediction. Let $\mathcal{C}$ be the union of logged and valid predicted types. For type $c \in \mathcal{C}$, TP_c counts $y_i = c, \hat{y}_i = c$, FP_c counts $y_i \neq c, \hat{y}_i = c$, and FN_c counts $y_i = c, \hat{y}_i \neq c$. We compute

$$\mathrm{Precision}_c = \frac{\mathrm{TP}_c}{\mathrm{TP}_c + \mathrm{FP}_c}, \tag{12}$$

$$\mathrm{Recall}_c = \frac{\mathrm{TP}_c}{\mathrm{TP}_c + \mathrm{FN}_c}, \tag{13}$$

$$F_{1,c} = \frac{2\,\mathrm{Precision}_c\,\mathrm{Recall}_c}{\mathrm{Precision}_c + \mathrm{Recall}_c}, \tag{14}$$

$$\mathrm{F1} = \frac{1}{|\mathcal{C}|} \sum_{c \in \mathcal{C}} F_{1,c}. \tag{15}$$

A zero denominator gives zero, and the final mean weights action types equally.

Accuracy (Acc.). An exact match requires the action type to match. Item and comment clicks must also match their targets, using an exact item-title match only when comparable item identifiers are unavailable. Missing targets and invalid predictions are misses. Let $\mathcal{C}_{\mathrm{NS}}$ denote the logged action types after excluding feed-page and item-page scroll-up and scroll-down actions. For type $c \in \mathcal{C}_{\mathrm{NS}}$, n_c denotes the number of logged steps and h_c denotes the exact matches. Then

$$\mathrm{Acc.} = \frac{1}{|\mathcal{C}_{\mathrm{NS}}|} \sum_{c \in \mathcal{C}_{\mathrm{NS}}} \frac{h_c}{n_c}. \tag{16}$$

This weights every retained action type equally. If $\mathcal{C}_{\mathrm{NS}}$ is empty, Acc. is not measured. We exclude the four scroll actions because they are target-free navigation primitives whose exact-match criterion reduces to action-type agreement. Treating them as four separate exact-match classes would over-represent low-level navigation and duplicate the action-type signal already measured by F1, rather than testing decisions with behavioral or target-specific consequences.

A.3.3 L2: Sequential Behavioral Fidelity (Rollout)

Decision-mix JSD (DM-JSD). For every item-page visit, we record the first decision in $\mathcal{D} = \{\mathrm{return}, \mathrm{cart}, \mathrm{buy}, \mathrm{end}, \mathrm{open}\}$. Let $q_X(d)$ denote the fraction of visits in session set $X \in \{\mathcal{R}, \mathcal{S}\}$ whose decision is $d \in \mathcal{D}$. For distributions P and Q over $\mathcal{D}$, let $M(d) = (P(d) + Q(d))/2$. We use the base-2 Jensen–Shannon divergence

$$\mathrm{JSD}(P, Q) = \frac{1}{2} \sum_{d \in \mathcal{D}} P(d) \log_2 \frac{P(d)}{M(d)} + \frac{1}{2} \sum_{d \in \mathcal{D}} Q(d) \log_2 \frac{Q(d)}{M(d)}, \tag{17}$$

where a zero-probability term contributes zero. JSD ranges from 0 for identical distributions to 1 for disjoint distributions. If both session sets contain no item-page visits, DM-JSD is 0. If exactly one set contains no visits, DM-JSD is 1. We compute

$$\text{DM-JSD} = \mathrm{JSD}(q_\mathcal{R}, q_\mathcal{S}). \tag{18}$$

Trajectory error (TE). For session s, ℓ_s denotes the number of executed policy actions, including the terminal action when present. For session set $X \in \{\mathcal{R}, \mathcal{S}\}$, $\bar{\ell}_X = |X|^{-1} \sum_{s \in X} \ell_s$ denotes mean trajectory length. Truncated sessions contribute their observed executed length. We compute

$$\mathrm{TE} = |\bar{\ell}_{\mathcal{S}} - \bar{\ell}_{\mathcal{R}}|. \tag{19}$$

A.3.4 L3: Personalized Population Fidelity (Rollout)

Population fidelity (PF). PF tests whether a simulator preserves behavioral differences among shopper groups rather than reproducing only population-average behavior. We compute it in four steps.

Step 1: Construct user cohorts. Using only history recorded before the evaluated session, we group users by activity, purchase propensity, and category breadth. We fit cohort boundaries on $\mathcal{R}$ and apply them unchanged to $\mathcal{S}$.

Step 2: Summarize cohort behavior. For every cohort in each session set, we construct a behavioral signature with eight blocks: session length, action distribution, information-depth distribution, item-page entry, item-page return, conversion, item category, and item price. Information depth counts information-gathering actions before the first return, cart, purchase, or session end within an item-page visit. We bin depth into four real-data quantiles and price into five, and represent categories using the ten most frequent real categories plus an other category. Item-page entry is the fraction of sessions that visit an item page, while item-page return is the mean fraction of item-page visits that return to the feed among sessions containing such a visit.

Step 3: Compare real and simulated cohorts. For cohort g, let $d_{g,b} \in [0, 1]$ denote the real–simulated discrepancy for block b. We use the base-2 JSD defined above for distribution blocks and absolute differences for rate blocks. The conversion discrepancy averages the cart-rate and purchase-rate gaps, while the session-length gap is scaled by the population standard deviation in $\mathcal{R}$ and capped at 1. If that standard deviation is zero, we scale by the real mean session length, or by 1 when both quantities are zero. The fidelity of cohort g is

$$F_g = 1 - \frac{1}{8} \sum_{b=1}^{8} d_{g,b}. \tag{20}$$

Step 4: Aggregate across cohorts and schemes. For scheme k, $\mathcal{G}_k$ contains cohorts with at least 30 real and 30 simulated sessions, and $F_k = |\mathcal{G}_k|^{-1} \sum_{g \in \mathcal{G}_k} F_g$. Let $\mathcal{K}$ contain schemes with at least two such cohorts. Then

$$\mathrm{PF} = \frac{1}{|\mathcal{K}|} \sum_{k \in \mathcal{K}} F_k. \tag{21}$$

Both averaging stages use equal weights. If $\mathcal{K}$ is empty, PF is not measured.

A.3.5 L4: Outcome Fidelity (Rollout)

Count-based CTR error (ΔCTR). For session set $X \in \{\mathcal{R}, \mathcal{S}\}$, c_X denotes the total number of click actions and f_X denotes the total number of executed feed-page actions. The count CTR is $v^X_{\mathrm{click}} = c_X / f_X$. We compute

$$\Delta\,\mathrm{CTR} = |v^{\mathcal{S}}_{\mathrm{click}} - v^{\mathcal{R}}_{\mathrm{click}}|. \tag{22}$$

If $f_X = 0$, we set $v^X_{\mathrm{click}} = 0$.

Count-based ACR error (ΔACR). For session set $X \in \{\mathcal{R}, \mathcal{S}\}$, a_X denotes the total number of add-to-cart actions and c_X denotes the total number of click actions. The count add-to-cart rate is $v^X_{\mathrm{cart}} = a_X / c_X$. We compute

$$\Delta\,\mathrm{ACR} = |v^{\mathcal{S}}_{\mathrm{cart}} - v^{\mathcal{R}}_{\mathrm{cart}}|. \tag{23}$$

If $c_X = 0$, we set $v^X_{\mathrm{cart}} = 0$.

Terminal-state JSD (T-JSD). Each session receives one outcome: purchase, cart only, browse only, or truncated. Purchase has first priority, followed by a truncation without normal termination, followed by cart, and then browse only. For session set $X \in \{\mathcal{R}, \mathcal{S}\}$ and outcome o, let $n_X(o)$ be the number of sessions assigned to o. We define the empirical terminal-outcome distribution as $T_X(o) = n_X(o)/|X|$ over the four outcomes. Then

$$\text{T-JSD} = \mathrm{JSD}(T_{\mathcal{R}}, T_{\mathcal{S}}). \tag{24}$$

B EXPERIMENT SETUP DETAILS

B.1 BASELINE DETAILS

We compare against eight representative user simulators while preserving each method's observation modality, context construction, memory mechanism, and training objective. All methods use the common USERARENA environment, action space, source test split, and output schema.

1. **RecAgent** (Wang et al., 2025a) conditions an LLM agent on a user profile and hierarchical sensory, short-term, and long-term memories. The memories are updated only with interactions that have already occurred.
2. **Agent4Rec** (Zhang et al., 2024) represents a user through profile, memory, and reflection modules. Retrieved memories and a compact reflection on prior interactions inform each action.
3. **SimUSER** (Bougie & Watanabe, 2025) constructs a profile from pre-session history and updates category preferences from executed interactions. Our adapted version retains one profile update per session and one action prediction per decision.
4. **Shop-R1** (Zhang et al., 2026b) is a text-based simulator trained with supervised fine-tuning followed by reinforcement learning. It receives the textual page state and interaction history but not the user profile.
5. **Customer-R1** (Wang et al., 2025c) extends the Shop-R1 training recipe with the shopper profile, recent clicks and purchases, and the executed interaction history.
6. **AlignUSER** (Bougie et al., 2026) augments supervised training with counterfactual actions and teacher-generated comparison rationales. Its evaluation prompt contains only the profile, past interactions, and current state.
7. **A/B Agent** (Zhang et al., 2026a) is a GUI-based simulator that combines the current screenshot with a user profile, retrieved long-term memory, working memory, and a fatigue mechanism.
8. **STA** (Zhang et al., 2025) is a GUI-based simulator that predicts a rationale and structured action from the current screenshot, textual page state, and interaction history. We use its reinforcement-learning-tuned checkpoint.

B.2 DATA PROCESSING AND IMPLEMENTATION

Data processing. We normalize the raw interaction logs into the 13-action USERARENA space and reconstruct the state immediately preceding each recorded decision. The visual observation and textual page state are drawn from the same pre-action interface state. We stratify users into training, validation, and test sets at an 8:1:1 ratio using seed 42, assigning all sessions from each user to the same split for both single- and cross-session experiments. Training and validation examples may be resampled to balance action coverage, whereas every test trajectory is retained once and is never resampled. Replay reconstructs each recorded state independently, while Rollout starts from the same initial context and thereafter advances only with actions accepted by the environment.

Implementation details. To control for backbone capacity, all methods in the main comparison, including the prompting-only baselines, use Qwen3.5-2B (Team, 2026) with a maximum context length of 32k tokens. This shared-backbone setting isolates differences in prompting, memory, observation modality, and post-training strategy from differences in model scale. We train every trainable baseline and OPTD on NVIDIA A100 GPUs and standardize their interaction interface. Their base prompts use the same task instruction, action definitions, and response schema, while retaining any method-specific context described in Appendix B.1. We use Qwen3.5-397B (Team, 2026) to generate a short first-person rationale for each recorded action from the corresponding user context, interaction history, and current state. Each SFT target pairs this rationale with the recorded structured action. The rationale encourages the model to summarize relevant evidence from the user context, interaction history, and current state before selecting an action, and using this format for every trainable method keeps the supervision interface consistent. We cap each GUI image at 200,704 input pixels. For OPTD, we perform an SFT warm-up on the same interaction format so that the resulting checkpoint can follow task instructions and generate a short rationale followed by a valid action from the current state. We then freeze this checkpoint as the teacher and initialize the student from it. We use ROLL (Wang et al., 2025b) for distributed training, vLLM (Kwon et al., 2023)

for on-policy generation, and Megatron-Core (Shoeybi et al., 2019) for student optimization and teacher inference. OPTD training uses BF16 precision, tensor and sequence parallelism, distributed optimization, and activation recomputation to reduce GPU memory use.

Replay uses greedy decoding and Rollout samples with temperature 1.0, top-p 0.95. Generation is capped at 512 tokens per decision. We cap Rollout at 31 executed actions for single-session evaluation and 42 for cross-session evaluation, one more than the maximum recorded action count in each setting. A trajectory is marked as truncated when it reaches the corresponding limit without terminating.

B.3 Training Hyperparameters

All trainable baselines and OPTD use a peak learning rate of 2×10^{-8}. For GRPO-trained baselines, each rollout batch contains 64 samples organized into groups of eight responses per prompt. We use an initial KL-controller coefficient of 0.001 and a KL-loss coefficient of 0.01, with temperature 0.99 and top-p 0.95 for rollout generation. For OPTD, we optimize the student with Adam and zero weight decay, using linear learning-rate warmup followed by linear decay. We set the Jensen–Shannon mixing weight to $\beta = 0.5$, which yields the standard Jensen–Shannon divergence, and the grounding weight to $\gamma = 0.1$. The trust region uses a maximum interpolation coefficient of $\lambda_{\max} = 1$ and a divergence budget of $\epsilon = 0.02$. At each token, we approximate the maximal feasible interpolation coefficient $\lambda^\star$ in Eq. 6 by running eight iterations of bisection over $[0, 1]$. The resulting interval has width at most $1/2^8 = 1/256$.

B.4 Prompt Structure

Table 5 summarizes the prompt structure used by OPTD. The student receives only information available at the current decision, including the user context, executed history, current interface state, and executable actions. Trajectory-level intent is appended only to the frozen teacher during training and is absent from student inference.

Table 5: Prompt structure used by OPTD. Italicized fields in braces are filled for each decision.

Prompt block	Template content
Shared student–teacher prompt	
System instruction	Role-play a real shopper, infer the next action from the supplied context, and return a concise first-person rationale followed by one structured action.
User context	{*user_profile*}, {*recent_purchases*}, and {*recent_clicks*}. The two history lists contain up to 20 titles available before the interaction, with consecutive duplicates collapsed.
Interaction history	{*interaction_history*} contains prior observations, rationales, and executed actions in chronological order. We retain the 10 most recent actions for single-session evaluation and 20 across visits for cross-session evaluation.
Current environment	{*current_screenshot*}, {*page_description*}, {*visible_elements*}, and {*page_state*}.
Executable actions	{*available_actions*} is a JSON list of actions accepted in the current state, including valid targets and the permitted scrolling range.
Teacher-only training context	
Trajectory-level intent	{*trajectory_intent*} contains high-level behavioral intent extracted from the complete training trajectory. It excludes evidence indices and is never exposed to the student.
Shared response contract	
Response format	{"rationale": "…", "action": {…}}. The action must match one executable candidate.

C Additional Experiment Details

C.1 Production A/B Validation Details

Section 4.3 validates simulator-based policy comparisons offline against outcomes recorded in a production A/B test. The production reference is computed from outcomes recorded over a fixed observation window in two production traffic buckets, one serving the control policy and the other the treatment policy. We evaluate the same pair of policies in simulation using a fixed pool of 10,000 users and a frozen product catalog. Each policy is assigned exactly 5,000 users, and every simulator receives the same user assignment and product catalog. We compare five task-specific checkpoints, including OPTD, STA, Customer-R1, Shop-R1, and AlignUSER, with the Qwen3.5-2B base model and two general-purpose foundation models, GPT-6-Sol and GPT-6-Luna.[3]

We compute PV-CTR within each policy arm as the total number of click actions divided by the total number of exposure page views. This definition matches the production metric and counts repeated exposures to the same item. For both the recorded production results and the offline simulation, we define the policy effect as ΔPV-CTR $=$ PV-CTR$_{\text{treatment}}$ $-$ PV-CTR$_{\text{control}}$. The effect ratio divides the simulated policy effect by the production policy effect, and the effect error is the absolute deviation of this ratio from 1.

Table 3 reveals three findings about simulator calibration to the production policy effect. First, OPTD provides the closest match, achieving an effect ratio of $0.713\times$ and an effect error of 0.287, which is 50.2% lower than that of the next-closest simulator. In contrast, the other task-specific simulators substantially attenuate the policy response, with effect ratios ranging from $0.020\times$ to $0.297\times$. The untuned Qwen3.5-2B model reaches only $0.004\times$. Second, the general-purpose GPT-6-Sol and GPT-6-Luna models overestimate the production response, reaching $1.576\times$ and $2.154\times$, respectively. This result indicates that general-purpose model capacity, without task-specific behavioral calibration, does not directly translate into realistic sensitivity to ranking-policy changes. Third, all evaluated simulators retain some gap from the production reference. Even OPTD, the closest method in this comparison, underestimates the production effect by 28.7%.

C.2 Efficiency Analysis

In this section, we compare how the inference cost of each method evolves over an interaction. For every Rollout step $k \in \{1, \ldots, 31\}$ of single-session evaluation, we record input, output, and total token counts together with model-call latency for each trajectory that reaches step k. If a method makes multiple calls at the same Rollout step, including planning calls or invalid-action retries, we first sum their token counts and latency within that trajectory-step. At each step, we then report the mean and sample standard deviation across the contributing trajectories. This step-wise protocol separates cost growth caused by an expanding interaction history from differences in when simulators terminate. All methods use the same inference hardware and evaluation configuration so that the latency comparison reflects method-level computation rather than different execution settings.

[3] Official model documentation: `https://developers.openai.com/api/docs/models/all`.

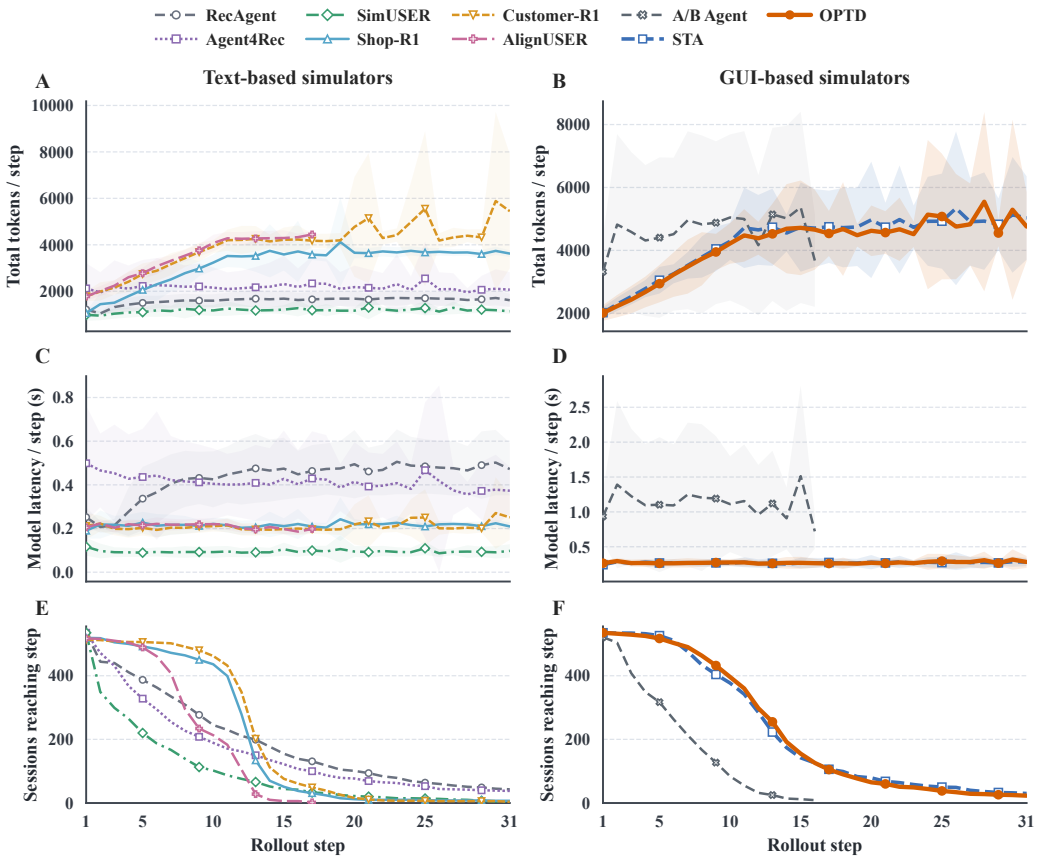


Figure 6: Step-wise inference cost on single-session Rollout. Lines show the mean across simulated trajectories that reach each step, and shaded bands span the mean ± 1 sample standard deviation. Before aggregation, we sum all model calls assigned to the same trajectory-step, including method-internal calls and invalid-action retries. Panels A–B report total tokens, Panels C–D report model-call latency, and Panels E–F report the number of contributing trajectories. No absent step is imputed, so curves end when no trajectory reaches a later step.

Figure 6 shows that the context-carrying GUI simulators exhibit the expected increase in token cost as interaction histories grow. Among text-based simulators, AlignUSER has the highest mean token cost at earlier steps, while Customer-R1 becomes the most expensive among the methods that reach later steps. Among GUI-based simulators, A/B Agent generally has the highest token cost and requires substantially more model calls per step than STA or OPTD. OPTD generally uses fewer total tokens than STA, while their latency curves remain closely aligned. Thus, the fidelity gains of OPTD do not require a systematic increase in per-step model latency. Additionally, the token-cost shaded bands widen at later steps because the remaining trajectories have accumulated more heterogeneous interaction histories, resulting in greater variation in token cost.

C.3 Qualitative Case Studies

We present representative examples to make the interaction structure of UserArena and the trajectory-level intent used by OPTD concrete.

C.3.1 Representative User Trajectories

Figure 7 shows one anonymized ground-truth trajectory from three consecutive visits by the same user. Panel (a) preserves the order and count of all 26 actions, grouping only consecutive feed-scroll

actions for readability. Panel (b) presents the full feed observation immediately before each of four product clicks, and the numbered markers connect these observations to their positions in the action sequence.

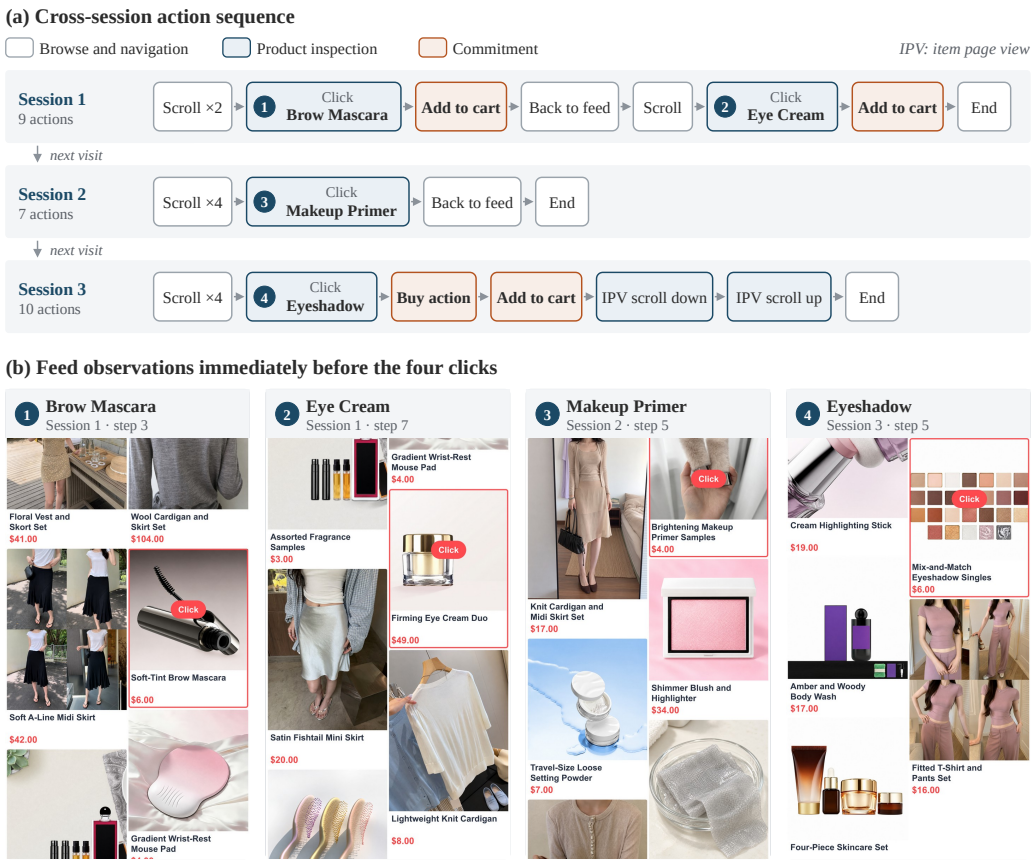


Figure 7: Ground-truth cross-session behavior over three consecutive visits by the same user. (a) Complete action sequence, with consecutive feed-scroll actions grouped as *Scroll* $\times N$ for readability. Colors distinguish browsing and navigation, product inspection, and commitment behavior. (b) Full feed observations immediately before four product clicks. Red outlines identify the selected products, and numbered markers link each observation to the corresponding click in panel (a).[5]

In Session 1, the user adds a brow mascara and an eye cream to the cart after alternating between feed navigation and product inspection. Session 2 continues the broader interest in beauty products but ends after viewing a makeup primer. Session 3 returns to eye makeup and records both *Buy Action* and *Add to Cart* for an eyeshadow product, followed by additional product-detail browsing. Across the three visits, the user's broader interest in beauty products persists, while the product focus shifts from brow and eye care to face makeup and then back to eye makeup. The example therefore illustrates how USERARENA preserves both continuity and modest shifts in user interests, together with changing levels of engagement, across chronologically ordered sessions.

[5]Product titles are translated for presentation. Item order, viewport position, clicked items, and action order are preserved.

C.3.2 TRAJECTORY-LEVEL INTENT EXAMPLE

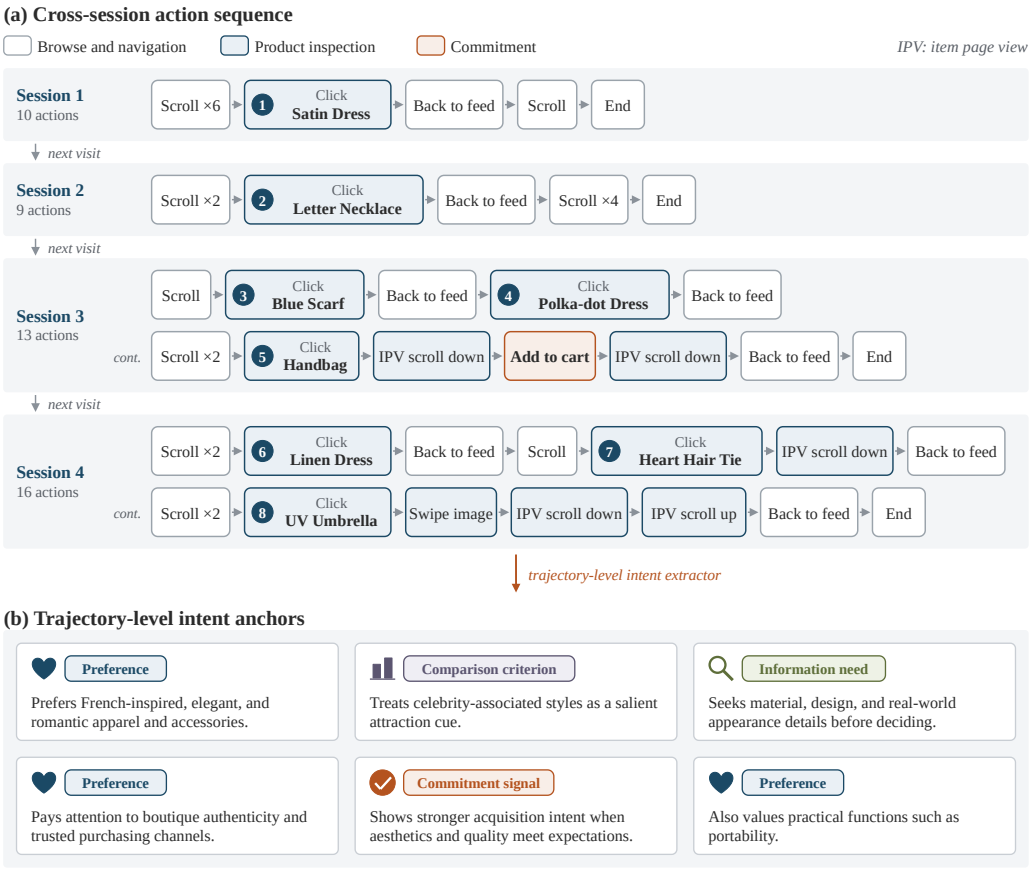


Figure 8: One cross-session trajectory and its structured trajectory-level intent. (a) Complete behavior across four consecutive visits by the same user. Colors distinguish browsing and navigation, product inspection, and commitment behavior. Consecutive feed-scroll actions are grouped as *Scroll* $\times N$, and the two longer visits wrap onto a continuation line for readability. (b) The single structured intent record extracted from the complete trajectory. Each card is a typed anchor within the same intent.

Figure 8 pairs a real-user training trajectory with the privileged trajectory-level intent extracted from it and supplied to the frozen teacher. Panel (a) preserves all 48 recorded actions, grouping only consecutive feed-scroll actions and wrapping the two longer sessions onto continuation lines. Panel (b) shows the resulting intent as it is represented during training: one structured record intent containing six complementary typed anchors.

Across the four visits, the user repeatedly inspects premium apparel and accessories, while varying the product category from dresses and jewelry to a scarf, handbag, hair accessory, and umbrella. The extracted preferences capture the stable interest in elegant design and branded goods. The comparison criterion abstracts the recurring attention to celebrity-associated styles, while the commitment signal reflects stronger acquisition intent when aesthetic and quality expectations align. None of these anchors identifies a specific product, page, price, or next action. The same structured record conditions the teacher at every state of this trajectory, while the student receives neither z_u nor future observations. The teacher must therefore guide the student's own response prefix at the current state, preserving the on-policy supervision used to reduce the mismatch between teacher-forced training and autoregressive inference.

C.4 TURING-STYLE HUMAN EVALUATION DETAILS

The source-blinded evaluation in Section 4.4 compares OPTD with Customer-R1, STA, and A/B Agent. For each user context, we pair a real trajectory with one generated by a simulator, anonymize the sources as A and B, and randomize their display order. Three annotators independently inspect the action sequences and corresponding interface observations, then choose *A is machine-generated*, *B is machine-generated*, or *Unsure*. Each simulator is evaluated on 200 trajectory pairs sampled from the held-out test set. **Annotator characteristics.** All annotators are adults (18+) familiar with e-commerce shopping interfaces and have experience with recommender systems or user-behavior annotation. All annotators provided informed consent and were compensated at prevailing local market rates. Recruitment considered non-sensitive demographic characteristics, educational background, and relevant domain expertise. No sensitive personal attributes were collected or used for annotator selection or analysis. Figure 9 shows the shared user context, both action timelines and their step-level interface observations, the three response options, and the required rationale field.

Figure 3(b) shows that OPTD has the lowest Human win rate (38.7%) and the highest Model win rate (26.3%) among the four simulators, while Customer-R1 has the highest Unsure rate (47.0%). Across trajectory pairs, Fleiss' κ is 0.528 (Fleiss, 1971). This statistic measures consistency among annotators rather than attribution correctness or human-likeness. Together, the agreement score and source-attribution rates indicate that annotators often reach similar judgments while still finding OPTD trajectories difficult to distinguish from recorded behavior. This pattern suggests that USERARENA's densely observed interactions provide useful supervision for learning human-like trajectories.

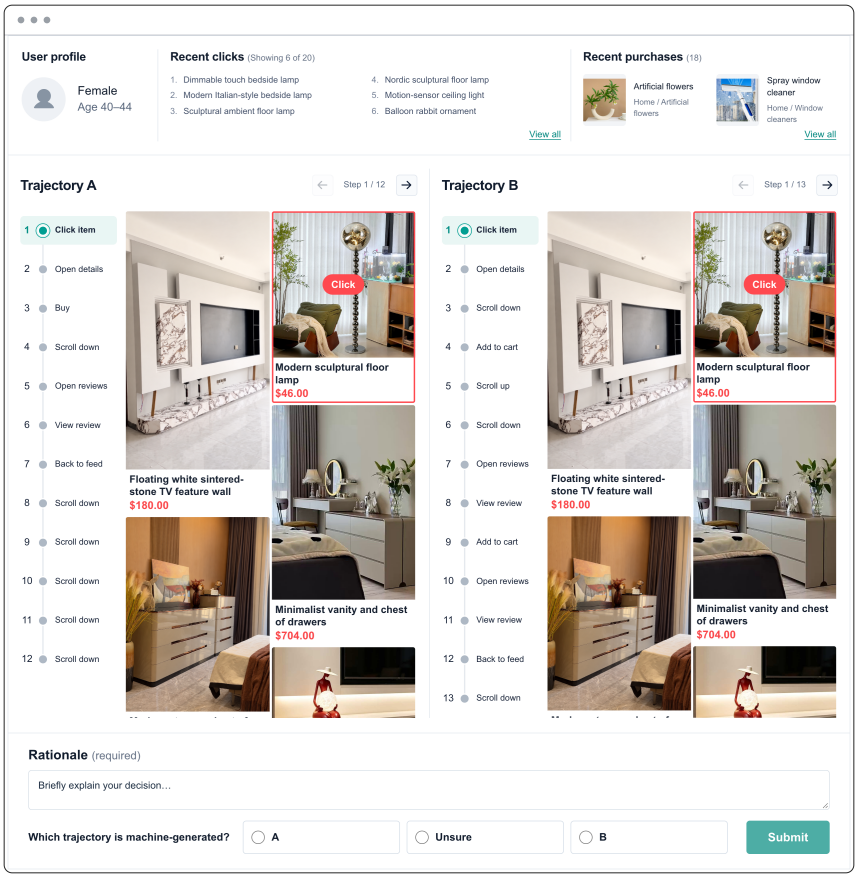


Figure 9: Human evaluation interface. Annotators compare two anonymized e-commerce interaction trajectories conditioned on the same user context. The interface presents recent interaction history, action timelines, and step-level interface observations. Annotators select the trajectory they believe is machine-generated or choose *Unsure*, and provide a supporting rationale.