Customer-R1: Personalized Simulation of Human Behaviors via RL-based LLM Agent in Online Shopping

Ziyi Wang[1], **Yuxuan Lu**[1], **Yimeng Zhang**[2], **Jing Huang**[3], **Dakuo Wang**[1],
[1]Northeastern University, [2]Michigan State University, [3]Amazon,
Correspondence: wang.ziyi19@northeastern.edu, d.wang@northeastern.edu

Abstract

Simulating step-wise human behavior with Large Language Models (LLMs) has become an emerging research direction, enabling applications in various practical domains. While prior methods, including prompting, supervised fine-tuning (SFT), and reinforcement learning (RL), have shown promise in modeling step-wise behavior, they primarily learn a population-level policy without conditioning on a user's persona, yielding generic rather than personalized simulations. In this work, we pose a critical question: how can LLM agents better simulate personalized user behavior? We introduce CUSTOMER-R1, an RL-based method for personalized, step-wise user behavior simulation in online shopping environments. Our policy is conditioned on an explicit persona, and we optimize next-step rationale and action generation via action correctness reward signals. Experiments on the OPeRA dataset demonstrate that CUSTOMER-R1 not only significantly outperforms prompting and SFT-based baselines in next-action prediction tasks, but also better matches users' action distribution, indicating higher fidelity in personalized behavior simulation.

1 Introduction

Human behavior simulation [12, 13, 8] aims to model how humans take actions. Recent advances in the reasoning capabilities [25, 17] of Large Language Model Agents (LLM Agents) have enabled both believable [12] and accurate [13] simulations of human behavior, drawing increasing attention to this topic. These advancements have opened up new application opportunities in various practical domains, including computational social science [12], psychology [1], e-commerce [8], and UX- testing [9].

Figure 1: User Behavior Simulation in Online Shopping. The model observes a sequence of historical user actions and learns to reason over this behavioral context to predict the user's next action.

Recent efforts in human behavior simulation have shifted from simulating coarse-grained or unverified human behaviors towards accurately modeling step-wise actions [8, 22, 28]. However, existing methods typically learn an average-user policy: they predict the most common next action in seen contexts,

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: Multi-Turn Interactions in Large Language Models.

but fail to account for individual differences in goals, preferences, or browsing styles. This lack of personalization limits their usefulness since different users may take very different actions in the same context [5]. For instance, Lu et al. [8] introduced step-wise user behavior simulation via supervised fine-tuning (SFT) on private data. Zhang et al. [28] proposed Shop-R1, a Reinforcement Learning (RL)-based approach to improve action generation accuracy. While promising, neither method is conditioned on individual user traits or preferences. Although Wang et al. [22] benchmarked the value of persona information in the OPeRA dataset, they only evaluated the prompting method with off-the-shelf LLMs, which yields marginal gains in aligning actions to a specific user. These gaps motivate our question: how can LLM agents better simulate personalized user behavior?

To study this question systematically, we introduce CUSTOMER-R1, a reinforcement learning-based method for step-wise and personalized user behavior simulation in online shopping scenarios. An overview of the task setup is illustrated in Figure 1. The model takes in a sequence of historical user actions taken by user 'Alex', and learns to reason over the behavioral context to predict Alex's next action accordingly. Our method leverages explicit user persona information to guide the model toward individualized behavioral patterns and introduces a tailored reward design to encourage accurate and semantically coherent action generation. We conduct extensive experiments on the OPeRA dataset [22], which includes rich user interaction logs and annotated persona profiles. Results show that CUSTOMER-R1 significantly outperforms prompting-based and SFT-based baselines in next-action prediction task and exhibits more aligned action distribution with persona information. Ablation studies further confirm the importance of persona conditioning: using correct persona information improves performance, while shuffled personas introduce noise and degrade accuracy. The contributions of this work are as follows:

1) We introduce CUSTOMER-R1, a reinforcement learning-based method for personalized, step-wise user behavior simulation in online shopping, incorporating explicit persona information and custom reward design.

2) We provide a comprehensive evaluation on the OPeRA dataset, demonstrating substantial improvements over existing methods.

3) We conduct detailed ablations and analysis on persona, rationale, model size, and context length, together with error studies. These results offer practical guidance for building personalized behavior simulators in online shopping.

2 Related Works

2.1 Human Behavior Simulation with Large Language Models

Understanding and simulating human behavior has long been a central goal in psychology, human-computer interaction, and computational social science [10, 18]. The emergence of large language model (LLM) agents with human-like reasoning, planning, and tool-use abilities [2, 25, 17] has opened new opportunities for modeling complex behaviors across diverse environments [21, 9, 20]. For example, in computational social science, generative agents have been used to simulate daily routines and social interactions in virtual communities [12]. However, many of these efforts primarily focus on generating "believable" user behavior, without quantitative evaluation against real human data. A few studies, such as Lu et al. [8], have explored step-wise behavior simulations in online shopping and evaluated next-action prediction using real-world user traces. Yet these approaches often rely on private datasets and supervised fine-tuning (SFT), limiting reproducibility and further generalization. More recently, Zhang et al. [28] applied reinforcement learning (Shop-R1) to further improve action generation. However, these works focus on simulating an "average" user instead of an unique individual.

In terms of personalization, recent studies have begun incorporating user personas into behavior simulation [19, 13, 22]. For instance, Park et al. [13] showed that agents equipped with interview-based persona profiles exhibit improved performance in survey-taking tasks. Wang et al. [22] also introduced persona into simulations, but the experiments are conducted solely on off-the-shelf LLMs without task adaptation, showing limited performance gain. As a result, verifiable and personalized user simulation at the action level remains underexplored. To fill the gap, this study investigate how user persona information can enhance personalized behavior simulation.

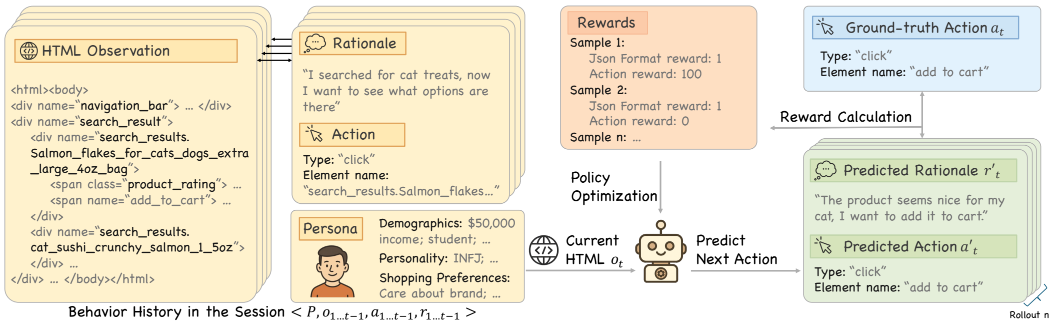

Figure 2: **CUSTOMER-R1 Framework for Simulating User Behavior in Online Shopping.** The model observes user history behaviors in a session composed of HTML observations $o_1, \ldots, o_{t-1}$, actions $a_1, \ldots, a_{t-1}$, rationales $r_1, \ldots, r_{t-1}$, along with real user persona P (demographics, personality, and shopping preferences). At time step t, given the current HTML observation o_t, the model predicts the rationale r'_t for conducting an action and the corresponding next action a'_t. During training, the model samples n rollouts per step. For each sampled prediction, a reward is calculated by comparing the predicted action a'_t with the ground-truth action a_t based on action correctness and format validity. These rewards are aggregated and used for policy optimization.

2.2 Reinforcement Learning for LLM Post-Training

Reinforcement learning (RL) has emerged as a powerful approach for training large language models (LLMs). Early methods, such as PPO [15], RLHF [11], and DPO [14] focus on aligning model outputs with human or proxy preferences. More rencently, methods with verifiable reward signals, such as GRPO [4], DAPO [26], and GSPO [29], have further improved stability and scalability. Furthermore, Chen et al. [3] systematically explores the contrast between supervised fine-tuning (SFT) and RL-based training paradigms, showing that GRPO-based methods can elicit stronger reasoning abilities compared to traditional SFT approaches. Despite these advancements, much of current RL application targets tasks with clear correctness criteria, such as mathematical problem solving [16, 27]. In these settings, reward design is straightforward because binary or graded notions of correctness are available. Extending RL to open-ended or user-centric tasks still poses challenges, particularly in defining meaningful and stable reward signals. In response, recent efforts have explored RL for more complex language interaction settings. For instance, RL has been applied to improve retrieval-augmented question answering systems [6] and recommender system outputs [7], where reward signals must account for relevance, diversity, or user engagement. Wei et al. [23] propose an end-to-end multi-turn RL framework for web agents, achieving higher task success rates by optimizing agent decisions over long-horizon interactions. However, the use of RL for simulating step-by-step personalized user behaviors remains underexplored. This presents a significant opportunity for future work.

3 Methods

3.1 Task Formulation

Following the setup in the OPeRA dataset [22], we formulate the objective as a next-action prediction task. Given a shopping session j, the model observes a history of user actions $\{a_1, a_2, ..., a_{t-1}\}$, their associated rationales $\{r_1, r_2, ..., r_{t-1}\}$, the sequence of web observations (i.e., HTML states) $\{o_1, o_2, ..., o_t\}$, and a user persona P_i. The model is tasked with generating the rationale r_t for the next action and predicting the next action a_t. Formally, the learning objective is to model the function:

$$r_t, a_t = F(a_{1...t-1}, r_{1...t-1}, o_{1...t}, P_i)$$

Each action type is associated with specific attributes that must also be predicted. Table 1 summarizes the required attributes for each action type.

3.2 CUSTOMER-R1 Framework

Table 1: Required attributes for each action type.

Action Type	Attributes	Example
click	element_name	click on "filter_price"
input	element_name, text	search for "earbuds"
terminate	None	terminate the session

The CUSTOMER-R1 framework is illustrated in Figure 2. Each simulation step is grounded in a real person from the dataset, with a corresponding action history, web page HTML, and annotated reasoning steps. The model is instructed to generate a rationale for conducting an immediate next action as well as the corresponding action. We incorporate a rich user persona consists of surveys and interviews, which capture user demographics, personality traits, and shopping preferences. These persona profiles provide high-level behavioral tendencies (e.g., brand loyalty, price sensitivity) that help the model generate actions consistent with an individual's style rather than an"average" user. Grounding in real-person personas allows the simulation to reproduce authentic decision patterns that are often missing from generic user models. In the prompt, we explicitly inject the persona description and instruct the model to follow it when producing plausible next actions. Nevertheless, the simulation remains context-driven: if the persona conflicts with evidence from the current page or the user's goals, the latter take precedence to maintain realism and task coherence.

To optimize the model, we define a verifiable reward function based on the predicted action. Specifically, we introduce a two-part reward computation:

- **Action reward** R_{action}: This component measures the correctness of the predicted action by directly comparing it against the ground truth action. Specifically, the reward is given only when both the action type and all required action attributes match exactly.

$$R_{\text{action}} = \begin{cases} 1 & \text{if } \hat{a}_{\text{type}} = a^*_{\text{type}} \text{ and } \hat{a}_{\text{attr}} = a^*_{\text{attr}} \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

  where $\hat{a}$ is the predicted action, and a^* is the ground-truth action. A reward of 1 is assigned only if all required fields match exactly between prediction and ground truth. For example, for a click action, the model needs to predict both the action type as well as the clicked element name correct.

- **Format reward** R_{format}: This binary reward ensures that the predicted action follows a predefined JSON schema that regulates generated reasoning and action.

The overall reward is computed as:

$$R = w(\hat{a}) \cdot R_{\text{action}} + R_{\text{format}} \tag{2}$$

where $\hat{a}$ is the predicted action. Given that simulating user behavior is a challenging generation task [22], we introduce a pre-defined difficulty-aware weighting function $w(a)$ that amplifies the reward for correctly predicting complex actions. This design mitigates the model's tendency to overfit to frequent, simple actions and incentivizes accurate prediction of rarer but more informative behaviors. In addition, the output format enforces the model to first generate a rationale before the action, which implicitly guides the model toward generating a more informative reasoning process alongside correct actions.

We adopt the Group Relative Policy Optimization (GRPO) [4] method as the reinforcement learning objective. The overall optimize goal is as follows:

$$J(\theta) = \mathbb{E}\left[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{t=1}^{|o_i|}\min\Big(r_{i,t}(\theta)\,\tilde{A}_i,\ \operatorname{clip}\big(r_{i,t}(\theta),\,1-\varepsilon,\,1+\varepsilon\big)\,\tilde{A}_i\Big) - \beta\,D_{\mathrm{KL}}\big(\pi_\theta \,\|\, \pi_{\mathrm{ref}}\big)\right], \tag{3}$$

Here, $r_{i,t}$ is the ratio between the new and old policy probabilities for sample i at token t, and $\tilde{A}_i$ is the group-relative advantage:

$$r_{i,t}(\theta) = \frac{\pi_\theta(o_{i,t} \mid q, o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t} \mid q, o_{i,<t})}. \tag{4}$$

$$\tilde{A}_i = \frac{R_i - \mu_R}{\sigma_R + \delta}, \tag{5}$$

4 Experiments

4.1 Data Processing

We conduct experiments on the OPeRA-filtered dataset [22], which contains 527 real-world online shopping sessions, comprising 5,856 <action, observation> pairs and 207 annotated rationales from 49 real users. The overall action distribution is shown in Table 2. On average, each session contains 11.11 actions. Among them, `click` is the dominant action type and is further splitted into 13 fine-grained subtypes as shown in Table 3. All sessions either end with a `click on purchase related button` action or a `terminate` action.

Given the long HTML-based contexts in these sessions, we implement a dynamic content selection strategy to fit within the model's maximum context length N. For each input, we compute its token length L; if $L > N$, we truncate by discarding the earliest HTMLs while preserving the full HTML content for the most recent interactions. For older interactions, we keep only the action and rationale tokens. This preserves temporally relevant page context while retaining semantically rich behavioral cues from earlier actions.

Table 2: Action in OPeRA-filtered.

Action Type	Count	Percentage
Click	5,051	86.3%
Input	597	10.2%
Terminate	208	3.6%
All	5856	-

In addition, the modeling framework requires the model to generate a rationale alongside each predicted action. However, as some action entries in the dataset lack annotated rationales, directly training with such incomplete data would hinder supervised fine-tuning (SFT). To address this, we adopt the rationale augmentation approach proposed by Lu et al. [8]. Specifically, we employ `claude-3.5-sonnet` to generate synthetic rationales by conditioning on the current HTML context and the user's executed action. These model-generated rationales serve as plausible supervisory signals to facilitate fine-tuning.

Table 3: Click type distribution in OPeRA-filtered dataset.

Click Type	Count	Percentage
review	1052	20.8%
search	763	15.1%
product_option	700	13.9%
product_link	537	10.6%
other	449	8.9%
purchase	321	6.4%
nav_bar	283	5.6%
page_related	198	3.9%
quantity	191	3.8%
suggested_term	182	3.6%
cart_side_bar	145	2.9%
cart_page_select	139	2.8%
filter	91	1.8%

4.2 Evaluation

To evaluate model performance on the next action prediction task, we utilize the following evaluation metrics: **a) Next Action Generation Accuracy**: An action prediction is considered correct only when all required components exactly match the ground truth. For example, for input actions, the model need to correctly predict the action type, input area, as well as the input text; **b) Action Type F1**: Measures the correctness of action type classification. Given the highly unbalanced action type distribution, the macro F1 score is reported. **c) Fine-grained Type Accuracy**: This metric measures the accuracy of predicted action types with finer granularity. For click actions, we first calculate the click subtype from the predicted target element and then compare it to the ground-truth. For non-click actions, we assess whether the model correctly identifies them as `terminate` or `input`. This provides a more detailed view of the model's understanding of user behavior patterns. **d) Session Outcome F1**: Evaluates whether the session is correctly predicted to end in `click on purchase related button` or `terminate`, capturing the overall user intent. These metrics collectively reflect both the step-wise fidelity of behavior prediction and the ultimate decision quality of the simulated user.

4.3 Experimental Setup

We use `Qwen2.5-7B-Instruct-1M` [24] as the main model and experiment with four training configurations. In the **Zero-shot Inference** setting, the model generates actions directly without any task-specific fine-tuning. In the **SFT** setting, we apply supervised fine-tuning using behavior

Table 4: Evaluation of next action prediction task. 'Action Gen.': Next Action Generation. 'Outcome': 'Session Outcome'. All metrics are reported as percentages (%).

Method	Action Gen. (Accuracy)	Action Type (Macro-F1)	Fine-grained Type (Accuracy)	Outcome (Weighted-F1)
Zero-shot Inference	7.32	33.43	25.72	41.11
RL	24.72	31.17	39.58	40.51
SFT	35.14	72.66	56.43	66.29
SFT+RL	**39.58**	**78.50**	**61.20**	**79.45**

traces annotated with ground-truth actions. In the **RL** setting, the model is optimized via GRPO with verifiable action-level rewards. Finally, in the **SFT+RL** setting, reinforcement learning is initialized from the SFT checkpoint to improve the training stability.

In terms of reward weighting, the action reward (R_{action}) is scaled by task difficulty in the SFT+RL setting: a) correct prediction of text inputs receive 2000; b) correct prediction on most click types (harder click subtypes) receive 1000; c) correct prediction of clicks on product_option receive 10; d) correct predicting clicks on reviews or search button receive 1; e) termination receives 1; f) incorrect clicks receive -1. In the RL-only setting, we use the same weighting scheme except that incorrect clicks receive 0 instead of a negative reward, since negative rewards made training unstable.

SFT training uses a standard token-level cross-entropy objective with the AdamW optimizer, a base learning rate of 1×10^{-5}, 150 warm-up steps, and 2,000 total training steps with a batch size of 64. RL training is conducted using the VERL + Megatron framework, with tensor model parallelism, context parallelism, and activation checkpointing enabled. We train for 2 epochs with a batch size of 64. All experiments are conducted on 8×8 P4de clusters, each equipped with A100 (80GB) GPUs.

The prompt used is shown in Appendix B.

4.4 Main Results

Table 4 presents the results for the next-action prediction task across all four settings. Zero-shot performance is low, with an Next Action Generation Accuracy of only 7.32%. This highlights the difficulty of behavioral prediction and the limitations of relying on pretrained knowledge alone without model adaptation. RL training alone improves exact match accuracy to 24.72%, but is unstable across other metrics. Applying SFT leads to significant improvements, boosting Next Action Generation Accuracy to 35.14%, and substantially improving both Action-Type F1 score and Fine-Grained Type Accuracy (72.66% and 56.43%, respectively). Combining SFT with RL yields the best performance across all metrics. By first grounding the model with supervised learning and then applying RL-based optimization initialized from the SFT checkpoint, this setting benefits from both stable pretraining and reward-driven refinement. Specifically, the method achieves the highest Next Action Generation accuracy of 39.58%, the best Macro F1 score, Fine-Grained Type Accuracy (78.50% and 61.20% respectively), and the highest Session Outcome F1 score of 79.45%.

4.5 Effect of Persona and Rationale

We quantify how explicit *persona* and intermediate *rationale* affect personalized behavior. Table 5 ablates these signals under four training regimes (Zero-shot, RL, SFT, SFT+RL) by removing persona text from the prompt (*w/o persona*) and further removing rationale from both input and generation (*w/o rationale*).

Under **SFT+RL**, both signals matter and they are complementary. Removing persona reduces Next Action Generation Accuracy by 1.78 points (39.58→37.80), Macro-F1 by 11.83 (78.50→66.67), Fine-grained Type Accuracy by 1.78 (61.20→59.42), and Session Outcome F1 by 19.72 (79.45→59.73). This indicates that persona provides user-level priors that help balance action types and decide when to purchase versus terminate. Removing rationale also consistently hurts performance. This shows that step-wise reasoning supports precise behavior generation.

For **Zero-shot** and **RL-only**, removing persona can increase some surface metrics (e.g., Next Action Generation Accuracy), likely because these weaker models are not trained to use long persona text and

Table 5: Model performance without persona or rationale. 'Zero-shot': Zero-shot Inference. 'Action Gen.': Next Action Generation. 'Outcome': 'Session Outcome'. All metrics are reported as percentages (%).

Method	**Setting**	**Action Gen. (Accuracy)**	**Action Type (Macro-F1)**	**Fine-grained Type (Accuracy)**	**Outcome (Weighted-F1)**
Zero-shot	-	7.32	33.43	25.72	41.11
Zero-shot	w/o persona	10.20	33.10	26.05	35.88
Zero-shot	w/o rationale	4.10	25.33	16.91	38.78
RL	-	24.72	31.17	39.58	40.51
RL	w/o persona	26.27	31.20	41.13	32.46
RL	w/o rationale	12.64	31.20	20.84	44.25
SFT	-	35.14	75.28	56.43	75.85
SFT	w/o persona	35.37	64.22	57.43	60.95
SFT	w/o rationale	32.04	67.93	52.22	71.38
SFT+RL	-	**39.58**	**78.50**	**61.20**	**79.45**
SFT+RL	w/o persona	37.80	66.67	59.42	59.73
SFT+RL	w/o rationale	34.15	73.15	53.99	67.37

the extra input may act as noise. However, Outcome F1 often degrades (e.g., Zero-shot 41.11→35.88, RL 40.51→32.46), suggesting that ignoring persona compromises user-level intent. In **SFT**, Next Action Generation Accuracy is similar with or without persona (35.14 vs. 35.37), but Macro-F1 and Outcome already show clear gains with persona (64.22 →75.28 and 60.95→75.85), implying that persona mainly helps general type balance and end-of-session decisions. Across all regimes, removing rationale is harmful, confirming that rationales act as a scaffold tying local page context to the chosen action.

4.6 Effect of Model Size and Context Length

We further experiment with a smaller backbone model (`Qwen2.5-3B-Instruct`) and compare two context length settings (40k vs. 65k tokens) under the reinforcement learning setup. As shown in Table 6, we observe clear performance improvements when increasing the context length for the 7B model. Specifically, action generation accuracy rises from 18.85% to 24.72%, and fine-grained type accuracy improves from 28.60% to 39.58%. Although the session outcome metric slightly decreases, this is primarily due to the 40k context model overfitting on the "click on purchase button" action, leading to inflated outcome predictions. The longer context window provides more examples of how a user would react to certain context and helps the model better retain earlier user intents, which is crucial for accurate simulation of behavior. In addition, the 3B model performs significantly worse across all metrics. Its action generation accuracy drops to 18.07%, and most notably, the outcome F1 score falls sharply to just 3.97%. This indicates that the smaller model fails to capture user intent.

Table 6: Ablation results showing the effect of model size, context length. 'Action Gen.': Next Action Generation. 'Outcome': 'Session Outcome'. All metrics are reported as percentages (%).

Model Size	**Context**	**Action Gen. (Accuracy)**	**Action Type (Macro-F1)**	**Fine-grained Type (Accuracy)**	**Outcome (Weighted-F1)**
Qwen2.5-7B	65k	24.72	31.17	39.58	40.51
Qwen2.5-7B	40k	18.85	31.14	28.60	41.41
Qwen2.5-3B	65k	18.07	31.30	38.91	3.97

4.7 Analysis

We further analyze the model's error patterns under different training regimes to understand its behavior and limitations.

From Reward Hacking to Balance with SFT Init. Without supervised grounding, the RL-only model learns to exploit the reward by favoring frequent, simple moves. As Table 7 shows, the model mostly predicts `click` action type and never predicts `input` or `terminate`.

Moreover, we noticed that under RL-only setting, the model over-selects subtypes with strong surface cues (e.g., `purchase`, `review`, `search`). While these predictions result in superficially high reward, they fail to reflect the diversity of real user behavior. This highlights a core limitation of naive reward-driven optimization: the learned policy may appear effective but lacks true generalization capability.

In contrast, initializing RL from a SFT checkpoint breaks this shortcut. The SFT policy already assigns non-trivial probability to rare actions, so RL can refine a balanced policy instead of relearning from scratch. The model shows a more balanced action prediction distribution and recovery of underrepresented actions (e.g., `terminate`) (Detail action distribution can be found in Appendix A).

Table 7: RL-only action type distribution and accuracy. The model ignores `input`/`terminate` and produces spurious `other` actions.

Action Type	Ground Truth	Predicted	Correct	Accuracy
Click	786	831	739	94.0%
Terminate	40	0	0	0.0%
Input	76	4	1	1.3%
Other	0	67	0	0.0%

Persona Guides How to Act and When to Stop. Within SFT+RL, real-person personas provide user-level priors (e.g., price sensitivity, brand loyalty) that resolve ties when page evidence alone is ambiguous. Removing persona destabilizes action type prediction and weakens end-of-session decisions. Specifically, the model shows a drift with fewer correct `purchase` and `terminate` predictions (Appendix A). In addition, Table 8 shows the results after shuffling the persona in the input information. Breaking the alignment between the user and their profile causes large drops across all metrics: Next Action Generation Accuracy 39.58→28.94, Action Type Macro-F1 78.50→38.88, Fine-grained Type Accuracy 61.20→40.35, and Session Outcome Weighted-F1 79.45→48.41. Moreover, persona information increases recall of rare but consequential actions (especially `terminate`) and improves calibration across types, yielding higher overall performance.

All these results demonstrates that persona information shifts the model policy from an *average user* heuristic to *this specific user's* behavior, deciding which element *this user* would act on and whether *this user* would stop.

Table 8: Model performance after shuffle the persona.'Zero-shot': Zero-shot Inference. 'Action Gen.': Next Action Generation. 'Outcome': 'Session Outcome'. All metrics are reported as percentages (%).

Method	Setting	Action Gen. (Accuracy)	Action Type (Macro-F1)	Fine-grained Type (Accuracy)	Outcome (Weighted-F1)
SFT+RL	persona	**39.58**	**78.50**	**61.20**	**79.45**
SFT+RL	shuffle	28.94	38.88	40.35	48.41

5 Conclusion

We introduced CUSTOMER-R1, a reinforcement learning method for step-wise, personalized user behavior simulation in online shopping. By conditioning the policy on explicit persona information and optimizing a tailored reward that favors action correctness, the model achieves superior next-action prediction accuracy on the OPeRA dataset and stronger personalization than prompting and SFT baselines. Comprehensive ablations and analysis show that persona and rationale make complementary contributions: persona supplies user-level priors that guide action selection under ambiguous page states, while rationale supports more stable credit assignment during RL. This combination improves calibration across action types and increases recall of rare but consequential actions such as `terminate`, leading to better end-of-session decisions and overall performance.

References

[1] Gati Aher, Rosa I. Arriaga, and Adam Tauman Kalai. Using large language models to simulate multiple humans and replicate human subject studies. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

[2] Anthropic. Claude 3.7 sonnet and claude code, February 2025. URL `https://www.anthropic.com/news/claude-3-7-sonnet`. Accessed: 2025-04-06.

[3] Hardy Chen, Haoqin Tu, Fali Wang, Hui Liu, Xianfeng Tang, Xinya Du, Yuyin Zhou, and Cihang Xie. Sft or rl? an early investigation into training r1-like reasoning large vision-language models. *arXiv preprint arXiv:2504.11468*, 2025.

[4] DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL `https://arxiv.org/abs/2501.12948`.

[5] Arief Helmi, Rita Komaladewi, Vita Sarasi, and Ledy Yolanda. Characterizing young consumer online shopping style: Indonesian evidence. *Sustainability*, 15(5):3988, 2023.

[6] Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *arXiv preprint arXiv:2503.09516*, 2025.

[7] Jiacheng Lin, Tian Wang, and Kun Qian. Rec-r1: Bridging generative large language models and user-centric recommendation systems via reinforcement learning. *arXiv preprint arXiv:2503.24289*, 2025.

[8] Yuxuan Lu, Jing Huang, Yan Han, Bingsheng Yao, Sisong Bei, Jiri Gesi, Yaochen Xie, Qi He, Dakuo Wang, et al. Prompting is not all you need! evaluating llm agent simulation methodologies with real-world online customer behavior data. *arXiv preprint arXiv:2503.20749*, 2025.

[9] Yuxuan Lu, Bingsheng Yao, Hansu Gu, Jing Huang, Jessie Wang, Laurence Li, Jiri Gesi, Qi He, Toby Jia-Jun Li, and Dakuo Wang. Uxagent: An llm agent-based usability testing framework for web design, 2025. URL `https://arxiv.org/abs/2502.12561`.

[10] James V McConnell. *Understanding human behavior: An introduction to psychology.* Holt, Rinehart & Winston, 1974.

[11] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback, 2022. URL `https://arxiv.org/abs/2203.02155`.

[12] Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th annual acm symposium on user interface software and technology*, pages 1–22, 2023.

[13] Joon Sung Park, Carolyn Q Zou, Aaron Shaw, Benjamin Mako Hill, Carrie Cai, Meredith Ringel Morris, Robb Willer, Percy Liang, and Michael S Bernstein. Generative agent simulations of 1,000 people. *arXiv preprint arXiv:2411.10109*, 2024.

[14] Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model, 2024. URL `https://arxiv.org/abs/2305.18290`.

[15] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017. URL `https://arxiv.org/abs/1707.06347`.

[16] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

[17] Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36:8634–8652, 2023.

[18] Karthik Sreedhar, Alice Cai, Jenny Ma, Jeffrey V Nickerson, and Lydia B Chilton. Simulating cooperative prosocial behavior with multi-agent llms: Evidence and mechanisms for ai agents to inform policy decisions. In *Proceedings of the 30th International Conference on Intelligent User Interfaces*, IUI '25, page 1272–1286, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400713064. doi: 10.1145/3708359.3712149. URL `https://doi.org/10.1145/3708359.3712149`.

[19] Chenkai Sun, Ke Yang, Revanth Gangi Reddy, Yi R Fung, Hou Pong Chan, ChengXiang Zhai, and Heng Ji. Persona-db: Efficient large language model personalization for response prediction with collaborative data refinement. *arXiv preprint arXiv:2402.11060*, 2024.

[20] Dakuo Wang, Ting-Yao Hsu, Yuxuan Lu, Hansu Gu, Limeng Cui, Yaochen Xie, William Headean, Bingsheng Yao, Akash Veeragouni, Jiapeng Liu, Sreyashi Nag, and Jessie Wang. Agenta/b: Automated and scalable web a/btesting with interactive llm agents, 2025. URL `https://arxiv.org/abs/2504.09723`.

[21] Lei Wang, Jingsen Zhang, Hao Yang, Zhiyuan Chen, Jiakai Tang, Zeyu Zhang, Xu Chen, Yankai Lin, Ruihua Song, Wayne Xin Zhao, Jun Xu, Zhicheng Dou, Jun Wang, and Ji-Rong Wen. User behavior simulation with large language model based agents, 2024. URL `https://arxiv.org/abs/2306.02552`.

[22] 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, 2025. URL `https://arxiv.org/abs/2506.05606`.

[23] Zhepei Wei, Wenlin Yao, Yao Liu, Weizhi Zhang, Qin Lu, Liang Qiu, Changlong Yu, Puyang Xu, Chao Zhang, Bing Yin, Hyokun Yun, and Lihong Li. Webagent-r1: Training web agents via end-to-end multi-turn reinforcement learning, 2025. URL `https://arxiv.org/abs/2505.16421`.

[24] An Yang, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoyan Huang, Jiandong Jiang, Jianhong Tu, Jianwei Zhang, Jingren Zhou, Junyang Lin, Kai Dang, Kexin Yang, Le Yu, Mei Li, Minmin Sun, Qin Zhu, Rui Men, Tao He, Weijia Xu, Wenbiao Yin, Wenyuan Yu, Xiafei Qiu, Xingzhang Ren, Xinlong Yang, Yong Li, Zhiying Xu, and Zipeng Zhang. Qwen2.5-1m technical report, 2025. URL `https://arxiv.org/abs/2501.15383`.

[25] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*, 2023.

[26] Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Mu Qiao, Yonghui Wu, and Mingxuan Wang. Dapo: An open-source llm reinforcement learning system at scale, 2025. URL `https://arxiv.org/abs/2503.14476`.

[27] Jingyi Zhang, Jiaxing Huang, Huanjin Yao, Shunyu Liu, Xikun Zhang, Shijian Lu, and Dacheng Tao. R1-vl: Learning to reason with multimodal large language models via step-wise group relative policy optimization. *arXiv preprint arXiv:2503.12937*, 2025.

[28] 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, 2025. URL `https://arxiv.org/abs/2507.17842`.

[29] Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, Jingren Zhou, and Junyang Lin. Group sequence policy optimization, 2025. URL `https://arxiv.org/abs/2507.18071`.

A Action Distribution

Figure 3 shows the fine-grained action type distribution of ground truth actions, predicted actions, and correctly predicted actions (i.e. Exact Match) among three training regimes: a) RL-only, b) SFT+RL, c) SFT+RL without persona information. We observe that, with RL-only method, the model tends to predict mostly purchase and review and search action. In contrast, under SFT+RL setting, the policy shows a balanced predicted action distribution, while removing the persona would make the performance on predicting termination to drop.

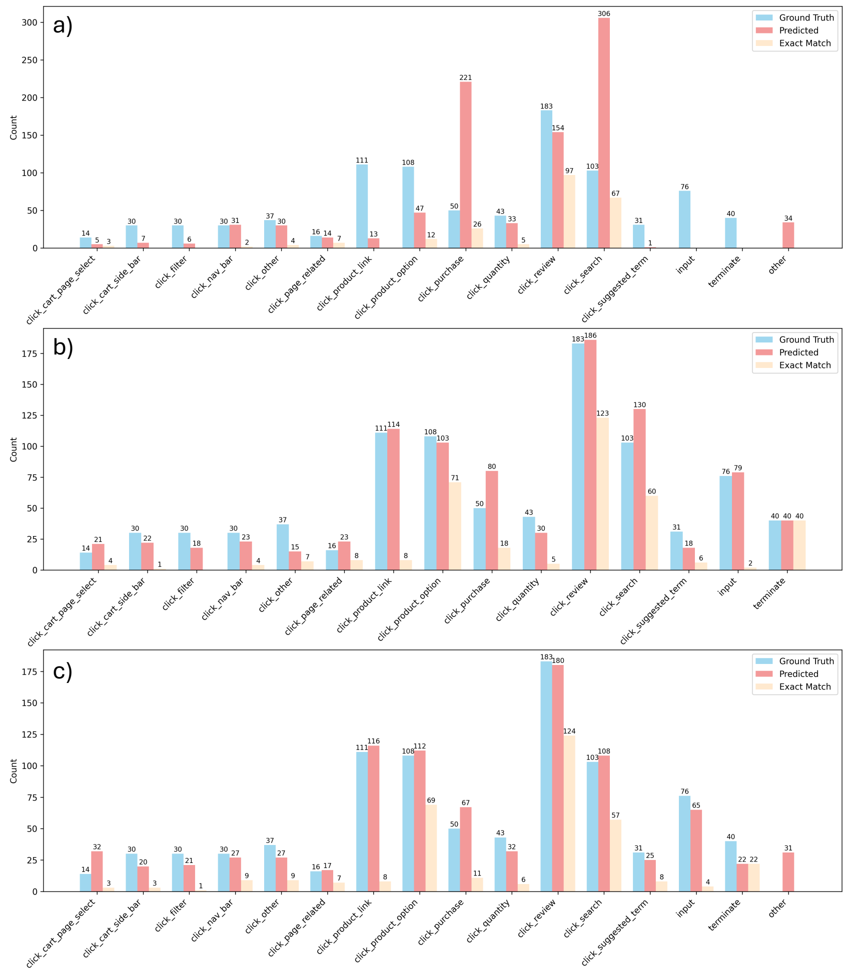

Figure 3: Fine-grained action distribution. a) Model trained using RL only. b) Model trained using SFT+RL. c) Model trained using SFT+RL without persona

B Experiment Prompt Design

Below are the two prompts for action prediction task and joint rationale and action generation task:

```
<IMPORTANT>
Your task is to predict the immediate next action of a shopper.
You need to pretend that you are a real user shopping on amazon.com.
The history action, rationale, context and the user persona will be provided to you.
Ensure your prediction follows natural behavior sequences (e.g., users may click a
    ↪ search box before typing, type a query before clicking search)
</IMPORTANT>
```

```
# Action Space

An action is represented in JSON format, and there are four primary types of
    ↪ actions:

#### 1. `input`:
Type text into an input field. The input field is identified by `name`.
{
    "type": "input",
    "name": "input_name",
    "text": "input_text"
}

#### 2. `click`:
Click on a button or clickable element identified by `name`.
{
    "type": "click",
    "name": "clickable_name",
}

#### 3. `terminate`:
When you are unsatisfied with the current search result and you don't want to buy
    ↪ anything, use `terminate` to indicate that you want to close the browser
    ↪ window and terminate the task.
{
    "type": "terminate"
}

# Rationale
Rationale is the reason why the user takes the action. Some of the rationale is
    ↪ provided to you.

# Context
Your context will be the HTML of the amazon page you are looking at. Some
    ↪ interactable elements will be added a unique "name" attribute, which you
    ↪ can use to identify the element to interact with (click or input).

# Persona
The user persona reflects the user's demographics, personality, and shopping
    ↪ preference. First identify which aspects of the persona might be relevant
    ↪ to the current shopping context, then consider them only if they naturally
    ↪ align with the ongoing shopping journey. DO NOT RELY ON IT.

# Output Format
You need to predict the rationale AND the corresponding next action. Your output
    ↪ should follow a strict JSON format:

{
    "rationale": "<rationale>", // rationale goes here, a string
    "action": {
        "type": "<type>",
        ...
    }// action goes here, a dictionary
}

<IMPORTANT>
OUTPUT A SINGLE JSON OBJECT, NOTHING ELSE.
</IMPORTANT>
```