Beyond Static Testbeds: An Interaction-Centric Agent Simulation Platform for Dynamic Recommender Systems

Song Jin[1*], **Juntian Zhang**[1*], **Yuhan Liu**[1†],
Xun Zhang[2], **Yufei Zhang**[2],**Guojun Yin**[2], **Fei Jiang**[2], **Wei Lin**[2], **Rui Yan**[1,3†]
[1]Gaoling School of Artificial Intelligence, Renmin University of China,
[2]Meituan, [3]Wuhan University
{jinsong8, zhangjuntian, yuhan.liu, ruiyan}@ruc.edu.cn
{zhangxun12, zhangyufei08, yinguojun02, jiangfei05, linwei31}@meituan.com

Abstract

Evaluating and iterating upon recommender systems is crucial, yet traditional A/B testing is resource-intensive, and offline methods struggle with dynamic user-platform interactions. While agent-based simulation is promising, existing platforms often lack a mechanism for user actions to dynamically reshape the environment. To bridge this gap, we introduce **RecInter**, a novel agent-based simulation platform for recommender systems featuring a robust interaction mechanism. In **RecInter** platform, simulated user actions (e.g., likes, reviews, purchases) dynamically update item attributes in real-time, and introduced *Merchant Agents* can reply, fostering a more realistic and evolving ecosystem. High-fidelity simulation is ensured through *Multidimensional User Profiling module*, *Advanced Agent Architecture*, and LLM fine-tuned on Chain-of-Thought (CoT) enriched interaction data. Our platform achieves significantly improved simulation credibility and successfully replicates emergent phenomena like *Brand Loyalty* and the *Matthew Effect*. Experiments demonstrate that this interaction mechanism is pivotal for simulating realistic system evolution, establishing our platform as a credible testbed for recommender systems research: RecInter.

1 Introduction

Recommender systems are integral to modern digital platforms, shaping user experiences and driving engagement across diverse domains. The ability to effectively evaluate and iterate upon these systems is paramount for their continued improvement and success (Ricci et al., 2010). However, traditional online A/B testing, while considered the gold standard, is often resource-intensive, time-consuming, and can carry risks associated with deploying untested algorithms to real users. Offline evaluation methods, conversely, struggle to capture the dynamic, interactive nature of user-platform engagement. Agent-based simulation has emerged as a powerful and cost-effective paradigm to bridge this gap, offering a controlled environment to test new algorithms, understand user behavior, and explore complex system dynamics such as filter bubbles, information cocoons, and user conformity behaviors (Zhang et al., 2024; Wang et al., 2025).

Early recommendation simulation, like Virtual Taobao (Shi et al., 2019) and RecSim (Ie et al., 2019), primarily rely on rule-based user models that, while useful, limited their behavioral realism and adaptability. The recent advent of Large Language Models (LLMs) has catalyzed a new wave of sophistication in agent-based modeling (Wang et al., 2023b; Xi et al., 2025). In the recommendation domain, RecAgent (Wang et al., 2025) pioneered the use of LLM agents to simulate diverse user activities on a platform. Agent4Rec (Zhang et al., 2024) and SimUSER (Bougie and Watanabe, 2025) enhance the reliability of user behavior simulation in recommender systems by incorporating enriched user profiling and specially designed memory mechanisms. However, a critical aspect often underrepresented in existing recommendation simulations is a deeply integrated interaction mechanism. Real-world platforms are not static; item attributes (e.g., popularity, average ratings, review sentiment) evolve dynamically based on continuous user feedback and even merchant interventions. Research (Lee and Hosanagar, 2021; Cui et al., 2012; Li et al., 2019) has already demonstrated the significant impact of these dynamic item attributes on user decision-making. This reciprocal relationship, where user actions reshape the environment and, in turn, influence future user decisions, is crucial for realistic simulation.

To address this gap, we introduce **RecInter**, a novel agent-based simulation platform for recommender systems incorporating a interaction mecha-

[*] Equal contribution to this work.
[†] Corresponding authors.

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 18903–18920
November 4-9, 2025

nism. Our platform allows simulated user actions such as liking, reviewing, or purchasing items to dynamically update item attributes in real time. Furthermore, we introduce *Merchant Agents* capable of replying to users, further enriching the dynamic nature of the simulated ecosystem. To achieve high-fidelity simulations, we integrate three components. Firstly, a *Multidimensional User Profiling* module to construct detailed user profiles from historical data. Secondly, our platform's User Agents are engineered with *sophisticated memory systems* (encompassing both perceptual and cognitive faculties) and advanced action selection mechanisms to more accurately emulate human decision-making processes. Thirdly, a *Behavior Simulation Training* pipeline is employed to fine-tune the LLM-based agents using high-quality, CoT enriched interaction data.

Our contributions are threefold:

• We develop a realistic recommendation simulation platform **RecInter** featuring a novel interaction mechanism where user feedback and merchant replies dynamically alter item attributes, fostering a more lifelike and evolving environment.

• Through Multidimensional User Profiling, Advanced Agent Architecture and Behavior Simulation Training, we achieve a higher authentication of simulated user behaviors, significantly surpassing previous methods.

• We explored and validated the crucial role of the interaction mechanism in modeling realistic system using **RecInter**. **RecInter** successfully reproduced the Brand Loyalty and Matthew Effect phenomena, demonstrating its credibility as a reliable testbed for recommendation systems research.

2 Related Work

2.1 LLM-based Agents

Large Language Model (LLM) agents are autonomous computational entities that perceive, decide, and act within their environment (Xi et al., 2025). With the emergence of LLMs, agent-based systems have gained renewed attention (Wang et al., 2023b). The Generative Agent framework (Park et al., 2023b) introduced agents with memory, planning, and reflection, simulating human cognition. Recent efforts divide LLM agents into task-oriented and simulation-oriented categories (Xi et al., 2025).

Task-oriented agents are designed to accomplish specific goals defined by users. For example, Voyager (Wang et al., 2023a) enables LLM-driven navigation in Minecraft, while ChatDev (Qian et al., 2023) and AutoGen (Wu et al., 2023) build collaborative multi-agent systems for software development. Simulation-oriented agents, on the other hand, focus on modeling human-like behaviors and social dynamics. The SANDBOX framework (Liu et al., 2023) explores social issues through multi-agent interaction to support LLM alignment with ethical norms, while WereWolf and AgentSims (Xu et al., 2023; Lin et al., 2023) use game-like environments to examine complex group dynamics. The FPS (Liu et al., 2024b) and FUSE (Liu et al., 2024c) respectively explore the use of LLMs-based agent for simulating the propagation and the evolution of fake news. Our framework provides a more accurate simulation of user reviews and merchant responses in recommendation scenarios.

2.2 Recommendation Simulation

Recommendation simulators have emerged as a cornerstone in recommender systems research (Afsar et al., 2023; Yang et al., 2021; Luo et al., 2022; Liu et al., 2025), offering a cost-effective alternative to online testing and addressing persistent challenges such as serendipitous discovery and filter bubbles (Huang et al., 2020; Chen et al., 2019). Early simulators primarily served as data sources for reinforcement learning applications. Notable examples include Virtual Taobao (Shi et al., 2019), which simulates e-commerce user behaviors, and RecSim (Ie et al., 2019), providing toolkits for sequential recommendation simulations. However, these conventional simulators often relied on simplistic rules, limiting their flexibility and validity. The recent advent of LLM-powered agents has shown remarkable potential in approximating human-like intelligence (Wang et al., 2023b), opening new avenues for more sophisticated recommendation simulators. A notable example is RecAgent (Wang et al., 2025), which pioneered the development of a recommendation platform integrating diverse user behaviors. Agent4Rec (Zhang et al., 2024) proposes an agent system composed of LLMs to simulate recommendation systems, and SimUSER (Bougie and Watanabe, 2025) is an agent framework that simulates human-like behavior to evaluate recommender algorithms, using self-consistent personas and memory modules. Building on these advancements, our research explores how user feedback and merchant replies dynamically influence item attributes, enabling more realistic recommendation simulations.

3 Methodology

Our simulation platform, **RecInter** (as illustrated in Figure 1), is designed to emulate a realistic recommendation scenario. To achieve this objective, we focus on two key aspects: (1) enhancing the credibility of user simulation, and (2) constructing a interactive recommendation platform environment. To improve user simulation accuracy, we introduce modules including Multidimensional User Profiling, an Advanced User Agent Architecture, and a Behavior Simulation Training pipeline. Additionally, we build an interactive recommendation platform environment that incorporates dynamic updates, merchant reply, and recommendation algorithm.

3.1 Problem Formulation

Let $\mathcal{U}$ denote the set of users and $\mathcal{I}$ represent the item set. For each user $u \in \mathcal{U}$, we first extract user profiles from their historical interaction sequences $\mathcal{H}_u = \{(i_1, r_1, c_1), (i_2, r_2, c_2), ..., (i_{N_u}, r_{N_u}, c_{N_u})\}$, where $i_j \in \mathcal{I}$ represents an interacted item with its rich contextual information, $r_j \in \{1, 2, 3, 4, 5\}$ denotes the rating provided by the user, and c_j denotes the textual review provided by the user for item i_j. We construct a user profile pool $\mathcal{P} = \{P(u)|u \in U\}$, where each profile $P(u)$ encodes information extracted from the user's historical interactions. This pool forms the basis for instantiating simulated user agents. **RecInter** operates for T time steps. At each time step t, the platform recommends a set of items $\mathcal{R}_t \subset \mathcal{I}$ to the simulated user agent, who then provides feedback $\mathcal{F}_t$ based on their preferences. This feedback subsequently updates the attributes A of items on the platform. Our objective is to minimize the behavioral discrepancy $\mathcal{D}(B_{real}, B_{sim})$ between the simulated user agents and real users, thereby creating a realistic simulation environment for recommender systems research.

3.2 Multidimensional User Profiling

As presented in the Figure 2, Multidimensional User Profiling involves constructing the user's objective, subjective, and inferred profile.

Objective Profile Building on the statistical metrics T_{act}, T_{conf}, and T_{cons} proposed by Zhang et al. (2024), we further introduce a set of novel indicators aimed at enhancing the realism of user objective profile. We conducted a systematic analysis of product categories within user interaction history $\mathcal{H}_u$ and identified the most top-k frequently interacted item categories as $T_{\text{cate}} = \{c_1, c_2, ..., c_k\}$, where k is set to 30. We also calculated the top-k most frequently interacted items as $T_{\text{item}} = \{i_1, i_2, ..., i_k\}$, where k is set to 10. We use $T_{\text{rate}} = \frac{1}{k}\sum_{i=1}^{k} \text{r}_i$ to calculate the user's historical average rate score. In our designed **RecInter**, users are able to leave reviews for items. Therefore, it is necessary to additionally consider the characteristics of user reviews. We define the probability that a user leaves a review as $T_{repr} = \frac{1}{k}\sum_{i=1}^{k} \mathbb{I}(r_i \neq \emptyset)$. To represent the average length of a user's historical reviews, we use $T_{\text{relen}} = \frac{1}{k}\sum_{i=1}^{k} \text{len}(r_i)$, where $\text{len}(r_i)$ denotes the length of the i-th review. Additionally, we explore the user's review style by extracting the top-N most frequent keywords from their reviews using TF-IDF (Salton and Buckley, 1988). We denote the set of these keywords as $T_{\text{rekey}} = \{w_1, w_2, ..., w_N\}$, where N is set to 20.

Subjective Profile While objective profile constitute a crucial part of user profile, relying solely on statistical indicators often fails to capture the more nuanced aspects of user preferences. To address this, we leverage LLM to derive subjective user profile. Specifically, for each user, we randomly sample 60 items from their historical interactions $\mathcal{H}_u$ and apply an LLM GPT-4o to perform information augmentation, generating more detailed descriptions for these items. The items' basic information A, augmented content $\bar{A}$, and user ratings r are provided as inputs to the LLM to facilitate the construction of subjective user profiles. This approach enables the model to summarize key aspects of the subjective profile, including taste preferences, consumption budget range, scenario preferences, and consumption habits. The prompt used and one profile case are provided in Appendix D.1 and Appendix C.1 respectively.

Inferred Profile User reviews have been shown to indirectly reflect personal profile (Sachdeva and McAuley, 2020; Srifi et al., 2020). Despite this, prior user simulation in recommender systems have largely ignored the potential of review data. To address this, we leverage LLM with carefully designed prompt to infer user profile from reviews. The input includes 60 items with each item's basic attributes A, augmented content $\bar{A}$, user ratings r, and review texts c. To reduce hallucinations, the model is instructed to output "unknow" when inference is uncertain. This process yields inferred pro-

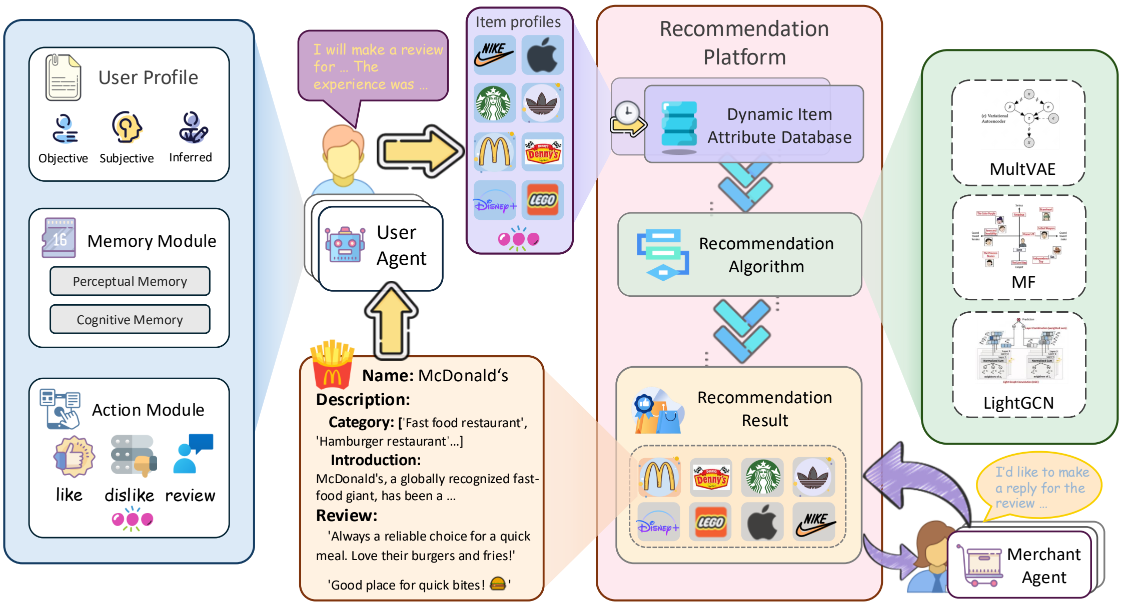

Figure 1: The overall framework of **RecInter**. The User Agent, equipped with user profile, memory module and action module, interacts with the platform by taking actions that can modify the attributes of items. In response, the platform guided by recommendation algorithm returns updated items to the user, thereby completing the interaction loop. Similarly, the Merchant Agent is also capable of participating in this dynamic process.

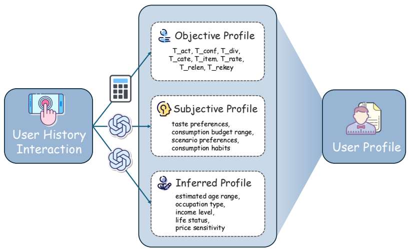

Figure 2: Multidimensional User Profiling framework.

file elements such as estimated age range, occupation type, income level, life status, price sensitivity, quality consciousness, service preferences, points of concern, and review language style. The prompt used and one inferred profile case are provided in Appendix D.2 and Appendix C.2 respectively.

3.3 Advanced User Agent Architecture

3.3.1 Memory Module

Recommender systems face unique challenges, including vast number of items, shifting user interests, and the influence of users' cognitive states. To address this, we introduce perceptual memory and cognitive memory.

Perceptual Memory Perceptual memory stores an agent's historical interactions and serves as the agent's past experiences. To enable faster and more accurate retrieval while conserving prompt space, we perform a simplification process that retains only the most essential information from each interaction. Specifically, each perceptual memory is represented as $m_t^p = (l_t, a_t, t)$, where $l_t = \{i_1, i_2, ..., i_n\}$ denotes the list of recommended items received by the agent at time step t, a_t represents the agent's action at that time.

Cognitive Memory In **RecInter**, agents engage in internal reasoning and judgment before taking actions, analogous to how individuals think before making a purchase. This internal cognitive process influences both current and future decisions. For instance, when presented with two chip flavors, a user may choose one now while planning to try the other later. To capture this process, we define cognitive memory at time step t as $m_t^c = (\bar{l}_t, s_t, a_t, t)$, where $\bar{l}_t$ is the list of recommended items with their attributes, s_t is the agent's thought process, a_t is the action taken, and t is the time step.

At time step t, the agent retrieves relevant information from its perceptual memory $M^p = [m_1^p, m_2^p, ..., m_{N_p}^p]$ and cognitive memory $M^c = [m_1^c, m_2^c, ..., m_{N_c}^c]$ before making action. Following Park et al. (2023a), each memory is assigned an importance score:

$$score_i = \alpha e^{-\gamma \cdot \nabla t} + \beta \text{sim}(m_i, \mathcal{R}_t)$$

where ∇t is the time elapsed since the memory was formed, $\text{sim}(m_i, \mathcal{R}_t)$ denotes the similarity between memory m_i and the current reasoning context $\mathcal{R}_t$, and α, β, γ are tunable parameters. In addition to retrieving the top-scoring memories, we impose adaptive thresholds θ_p and θ_c to dynamically adjust the maximum number of retrieved memory for each memory type. The specific hyper-parameter is included in Appendix A.3.

3.3.2 Action Module

We use a "think-then-act" approach, similar to chain-of-thought reasoning, enables simulated users to behave more similarly to real users, following Yang et al. (2024). Inspired by real-world recommendation platforms, we designed a rich set of user actions that more closely reflect authentic interactions. Unlike previous works (Zhang et al., 2024; Wang et al., 2025), our simulated users exhibit interactive actions that can dynamically alter the attributes A of items within the system. The action space includes: do nothing, like product, dislike product, share product, purchase prodcut, create review, like review, and dislike review.

3.4 Interactive Platform Environment

3.4.1 Interaction Implement

To achieve a more realistic simulation of recommender systems, we incorporate an interaction mechanism into **RecInter**. Specifically, we implement a set of database tables associated with items, which store their dynamic attributes A. The agents' actions are enhanced such that each action a can update these tables in real time. At each recommendation time step, the platform queries the most recent item attributes A from the database and presents them to the simulated users. This method allows the platform to update items dynamically based on user feedback and enable interaction.

3.4.2 Merchant Reply

In real-world recommendation platforms, merchants also can make changes to the attributes of their items, such as responding to user reviews or modifying product descriptions. To better simulate this interactive environment, we incorporate merchant agent into **RecInter**. The merchant agent autonomously updates item attributes based on its own strategy and interacts with users through reviews. This addition enables the study of merchant behavior within the recommender systems.

3.4.3 Recommendation Algorithm

Recommendation algorithms also constitute a critical component in the simulation of recommender systems. In **RecInter**, we have integrated a variety of algorithms, including random, most popular, LightGCN (He et al., 2020), MultVAE (Liang et al., 2018), and MF (Koren et al., 2009). This module is designed to be flexible and extensible, allowing for the incorporation of custom recommendation algorithms as well. These algorithms aim to recommend items that users are likely to be interested in, thereby enhancing user satisfaction and engagement within the simulation.

3.5 Behavior Simulation Training

We adopt a Chain-of-Thought (CoT) fine-tuning approach to enhance the reliability of agent simulation in recommender systems. To construct our training dataset, we used GPT-4o as the base model and ran **RecInter** multiple times, guiding the model to "think-then-act". This process generated a substantial number of simulated interactions enriched with CoT reasoning. To ensure the quality of the data, we implemented a multi-stage filtering pipeline consisting of four key components: (1) Format Filter: Ensures that the model outputs conform to the required structural format. (2) Preference Filter: Verifies alignment between the agent's simulated actions and the user's actual preferences by leveraging real user interaction data. Specifically, we check whether positively interacted items appeared in the user's real interactions and whether negatively interacted items did not. (3) LLM Filter: Utilizes LLM to assess the plausibility and consistency of the simulated outputs. (4) Human Filter: Involves manual verification to further ensure data quality.

Following this pipeline, we curated a high-quality dataset comprising 5,295 CoT enhanced samples. We then fine-tuned the Qwen-2.5-7B-Instruct model, resulting in the base model that achieved the best simulation performance. The fine-tuning setting is provided in the Appendix A.4.

4 Experiment

4.1 Simulation Credibility

Settings We employ the fine-tuned Qwen2.5-7B-Instruct as the default base model for **RecInter**. By default, the recommendation algorithm used is LightGCN, with 10 time steps and 1,000 user agents. The GoogleLocal serves as the default dataset. Additional experimental settings and variations will be specified in subsequent each sections.

Baselies Our baselines include RecAgent (Wang et al., 2025), Agent4Rec (Zhang et al., 2024), SimUSER (Bougie and Watanabe, 2025). Please refer to Appendix A.2 for more details.

4.1.1 Metric-Based Evaluation

This part of the experimental setup follows the evaluation used in Zhang et al. (2024). In the experiments, 1,000 agents provide feedback on 20 random items. These 20 items consist of both really interacted and non-interacted items by the users, mixed in a $1 : m$ ratio. Each agent selects the items they are interested in. Based on the correctness of the agents' selections, we compute evaluation metrics such as Accuracy, Precision, Recall, and F1 Score. Experiments are conducted on three real-world datasets: Google Local (Li et al., 2022), MovieLens (Harper and Konstan, 2015), and AmazonBook (McAuley et al., 2015). The preprocessing steps for datasets are described in the Appendix A.1. All experimental results are showed in Table 1. The experimental results demonstrate that our approach outperforms existing methods across all datasets, indicating that the proposed Multidimensional User Profiling and Behavior Simulation Training significantly enhance the accuracy of user simulation in recommender systems.

4.1.2 LLM-Based Evaluation

To address the limitations of metric-based evaluatios in capturing complex agent behaviors, we introduce an LLM-based evaluation that assesses performance across the entire simulation process. The detailed experimental procedures are provided in the Appendix B.6. The complete results are presented in Table 5. As shown, our method achieves a most higher Adjusted Win Rate of 0.6917. To further validate the reliability of our LLM-based evaluation method, we conducted a human evaluation study in Appendix B.7.

4.1.3 Macro-Level Evaluation

In addition to assessing the simulation credibility from the perspective of the user agent, we further evaluated the overall credibility of the simulation by comparing the alignment between the real data and simulation results in terms of the distribution of actions and items.

Actions Distribution We conducted a comparative analysis of behavioral distributions between real users and simulated agents across three representative actions: like, dislike, and review. Real user data was extracted from the GoogleLocal dataset, while simulated agent data was obtained by executing a full simulation process and then collecting the corresponding actions statistics. As shown in Figure 3, the distribution patterns between the two groups are generally aligned.

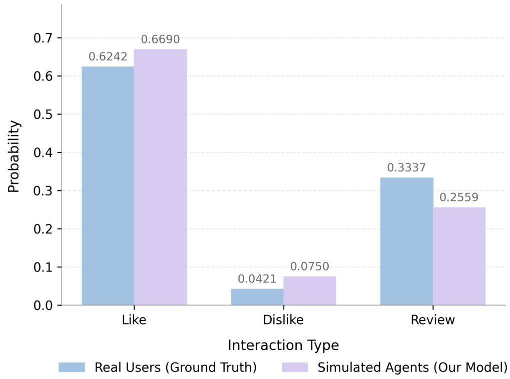

Figure 3: Comparison of actions distribution.

Items Distribution We also examined differences in items popularity between real users and simulated agents. For real users, we identified the top-10 most popular items from the GoogleLocal dataset and computed their frequency distributions. Similarly, for the simulated agents, we analyzed the top-10 most frequently interacted items based on the simulation results. As shown in Figure 11, five out of the top six most popular items overlapped between the two groups, and the overall popularity distributions showed a high degree of similarity.

4.2 Impact of Interaction Mechanism

This part of the experiment shows that introducing an interaction mechanism significantly affects the evolutionary process of recommender system simulation, highlighting its indispensable role in the simulation. In Section 4.2.1 and Section 4.2.2, we investigate the impact of the presence or absence of interaction and malicious interaction on the simulation evolutionary process. In the Appendix B.1, we demonstrate that the interactive attitude of merchants also shapes this process.

4.2.1 Impact of the Presence or Absence of Interaction

We conducted two simulations: one incorporating an interaction mechanism and the other without it, while keeping all other settings same. The goal was to observe differences in the simulated evolutionary process between the two simulations, focusing on the changes in the number of likes received by two restaurants—McDonald's and Denny's. McDonald's represents the most popular option, while Denny's is considered moderately popular. The statistical results are presented in the Figure 4.

The findings indicate a significant difference in the likes for McDonald's under the two simulation

Method	GoogleLocal				MovieLens				AmazonBook			
	Accuracy	Precision	Recall	F1 Score	Accuracy	Precision	Recall	F1 Score	Accuracy	Precision	Recall	F1 Score
RecAgent (1:1)	0.5643	0.5832	0.5342	0.5576	0.5807	0.6391	0.6035	0.6205	0.6035	0.6539	0.6636	0.6587
RecAgent (1:3)	0.5012	0.6134	0.3765	0.4666	0.5077	0.7396	0.3987	0.5181	0.6144	0.6676	0.4001	0.5003
RecAgent (1:9)	0.4625	0.6213	0.1584	0.2523	0.4800	0.7491	0.2168	0.3362	0.6222	0.6641	0.1652	0.2647
Agent4Rec (1:1)	0.6281	0.6134	0.6223	0.6178	0.6912	0.7460	0.6914	0.6982	0.7190	0.7276	0.7335	0.7002
Agent4Rec (1:3)	0.6012	0.6456	0.3905	0.4866	0.6675	0.7623	0.4210	0.5433	0.6707	0.6909	0.4423	0.5098
Agent4Rec (1:9)	0.5786	0.6631	0.2042	0.3112	0.6175	0.7753	0.2139	0.3232	0.6617	0.6939	0.2369	0.3183
SimUSER (1:1)	0.6823	0.6312	0.6754	0.6526	0.7912	0.7976	0.7576	0.7771	0.8221	0.7969	0.7841	0.7904
SimUSER (1:3)	0.6489	0.6624	0.3893	0.4904	0.7737	0.8173	0.5223	0.6373	0.6629	0.7547	0.5657	0.6467
SimUSER (1:9)	0.6042	0.6923	0.2187	0.3324	0.6791	0.8382	0.3534	0.4972	0.6497	0.7588	0.3229	0.4530
RecInter(1:1)	**0.7143**	0.6646	**0.7057**	**0.6854**	**0.7947**	0.8092	**0.7595**	**0.7812**	**0.8302**	**0.8049**	**0.7901**	**0.7975**
RecInter(1:3)	0.6753	0.7038	0.4312	0.5357	0.7852	0.8236	0.5474	0.6476	0.6804	0.7651	0.5813	0.6614
RecInter(1:9)	0.6218	**0.7580**	0.2508	0.3769	0.6869	**0.8391**	0.3638	0.5054	0.6634	0.7631	0.3214	0.4547

Table 1: Metric-based comparison of simulation credibility across different methods with the best results highlighted in bold. Our approach achieves superior performance compared to baselines across all datasets.

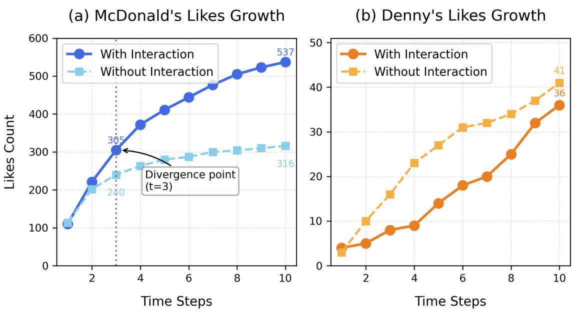

Figure 4: Impact of interaction mechanism on likes.

conditions. In the simulation with the interaction mechanism, McDonald's received a substantially higher number of likes by the end of the simulation. More specifically, from time step t = 3 onward, the increase in likes for McDonald's became notably greater in the interaction-enabled simulation. This can be attributed to a surge in likes, sales, and reviews for McDonald's at that point, which likely influenced agents to favor McDonald's more frequently. In contrast, the difference between the two simulations for Denny's was relatively minor. This suggests that Denny's had insufficient attention in the early stages of the simulation, thereby exerting limited influence on the later stages. Additionally, a case study on agent responses in the Appendix C.3 further demonstrates that the interaction influences user decision-making during the simulation.

4.2.2 Impact of Malicious Interaction

We further investigated the impact of malicious interaction on the evolution of the simulation. Specifically, at time step $t = 5$, we introduced three malicious reviews targeting McDonald's. By comparing the trends in user likes, purchases, and reviews with or without the introduction of these reviews, we aimed to assess their influence on user behavior. The results are presented in the Figure 5. Compared to the scenario without malicious reviews, the inclusion of such reviews led to a noticeable deceleration in the growth of likes, purchases, and reviews. These findings suggest that malicious reviews can significantly influence user decision-making, which aligns with real-world observations.

4.3 Ablation Study

In the ablation study, we evaluate the contributions of the Multidimensional User Profiling and Behavior Simulation Training modules. The experimental setup follows the same configuration as described in Section 4.1.1 with the parameter m set to 1. The w/o personalization variant directly uses LLM to summarize the user's profile from 60 sampled history interactions. The w/o training variant employs an untrained Qwen-2.5-7B-Instruct as the agent. As shown in the Table 2, **RecInter** achieves the best performance, indicating that both Multidimensional User Profiling and Behavior Simulation Training play critical roles in enhancing the realism of user behavior simulation. In the Appendix B.4, we conduct ablation studies by replacing different base models, demonstrating that our fine-tuned model achieves the best performance. Furthermore, Appendix B.5 evaluate the impact of different recommendation algorithms and show that a effective recommendation strategy significantly enhances the engagement of simulated users on the platform.

Method	Accuracy	Precision	Recall	F1 Score
w/o personalization	0.5733	0.5865	0.5641	0.5601
w/o training	0.6715	0.6229	0.6732	0.6471
RecInter	**0.7143**	**0.6646**	**0.7057**	**0.6854**

Table 2: Ablation study results over 2 variants.

4.4 Phenomenon Observation

In Section 4.4.1 and Section 4.4.2, we reproduced the phenomena of Brand Loyalty and Matthew Ef-

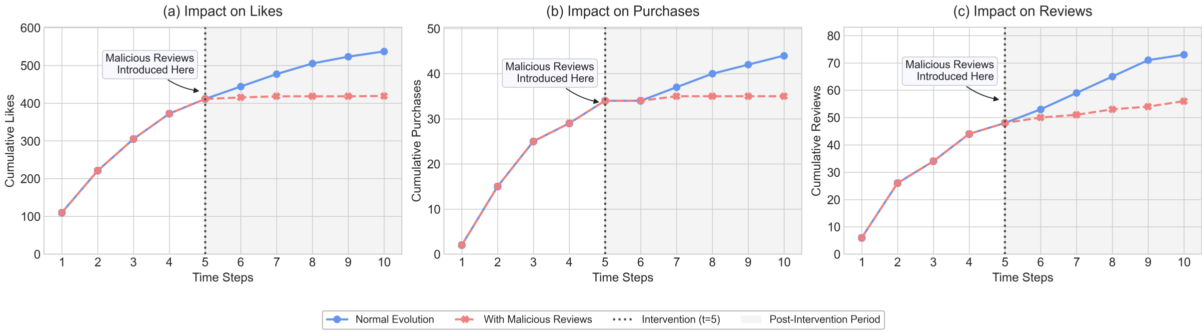

Figure 5: Impact of malicious interaction on the cumulative number of likes for McDonald's.

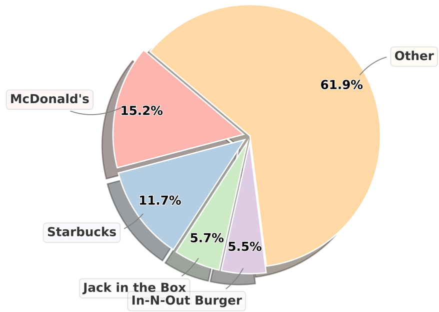

Figure 6: Distribution of user interactions across items.

fect. We also analyzed Conservative Behavior phenomenon through the reviews of simulated user in the Appendix B.2.

4.4.1 Brand Loyalty

We analyzed the proportion of interactions each item received at the final stage of the simulation relative to the total number of interactions, as shown in Figure 6. The results indicate that brand-related items were significantly more popular, with McDonald's and Starbucks accounting for 15.2% and 11.7% of all interactions respectively. This suggests the presence of Brand Loyalty among the simulated users. To further investigate this observed Brand Loyalty, we conducted an additional experiment. Specifically, we replaced the brand name "McDonald's" with a fictitious name "Stack Shack" while keeping all other attributes of the item, including its recommendation probability, unchanged. The purpose was to examine whether altering the brand name would influence user's choice. As shown in Figure 7(a), the number of likes for the item decreased substantially after the brand modification.

4.4.2 Matthew Effect

In Section 4.2.1, we showed that adding an interaction mechanism makes users more likely to choose items that received positive feedback earlier. This

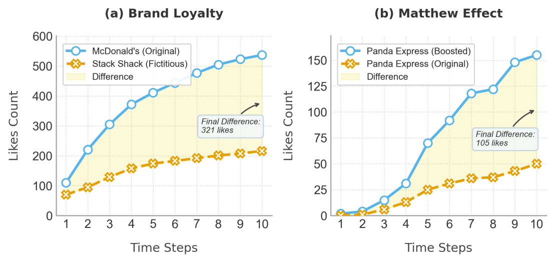

Figure 7: The phenomenon observation of Brand Loyalty and Matthew Effect.

makes those items more popular over time, which is an example of the Matthew Effect. To further test this effect in our simulation, we did another experiment. We chose Panda Express because it had a medium level of popularity. Before the simulation started, we gave it three positive reviews and set its initial sales to 100. We wanted to see if this would increase its number of likes. As shown in the Figure 7(b), this manual boost did lead to more likes for Panda Express. This shows that users prefer popular items, supporting the idea that the Matthew Effect exists in our simulation.

5 Conclusion

We introduced **RecInter**, an agent-based simulation platform featuring a novel interaction mechanism. In **RecInter**, User actions and merchant replies dynamically reshape item attributes, addressing a critical gap in prior simulations. By integrating Multidimensional User Profiling, advanced agent architecture, and Chain-of-Thought fine-tuned LLM, **RecInter** achieves significantly improved simulation credibility. Our experiments highlight that this dynamic interaction is pivotal for realistically modeling system evolution and observing emergent phenomena like Brand Loyalty and the Matthew Effect. These capabilities position **RecInter** as a valuable and credible testbed for recommender systems research.

6 Limitations & Potential Risks

Several limitations warrant consideration regarding the current platform's capabilities. Firstly, the depth of user profiling, while multidimensional, relies on LLMs for generating subjective and inferred profiles. Despite efforts to mitigate hallucinations, the inherent biases and comprehension limitations of LLMs may affect the accuracy and completeness of these profiles, potentially failing to fully capture the nuanced complexity of human preferences. Secondly, the user agents operate within a predefined action space (e.g., like, review, purchase). Although this set of actions is rich, it remains fixed, whereas real-world user behavior can be more creative, emergent, or extend beyond these predefined categories. Lastly, the platform's interaction model primarily focuses on user-item, merchant-item, and merchant-user dynamics. Consequently, other significant forms of interaction prevalent in real-world recommendation ecosystems, such as user-to-user social influence, community dynamics, or the impact of external events, are not explicitly modeled in the current implementation. Bias propagation remains a risk. Biases present in these sources (e.g., demographic underrepresentation in historical data, inherent biases in the pre-trained LLMs) could be inadvertently amplified within the simulation.

Acknowledgments

This work is supported by Meituan through Agentic system X Program. This work is also supported by the Public Computing Cloud, Renmin University of China and by fund for building worldclass universities (disciplines) of Renmin University of China.

References

Mohammad Mehdi Afsar, Trafford Crump, and Behrouz H. Far. 2023. Reinforcement learning based recommender systems: A survey. *ACM Comput. Surv.*, 55(7):145:1–145:38.

Nicolas Bougie and Narimasa Watanabe. 2025. Simuser: Simulating user behavior with large language models for recommender system evaluation. *arXiv preprint arXiv:2504.12722*.

Xinshi Chen, Shuang Li, Hui Li, Shaohua Jiang, Yuan Qi, and Le Song. 2019. Generative adversarial user model for reinforcement learning based recommendation system. In *ICML*. PMLR.

Geng Cui, Hon-Kwong Lui, and Xiaoning Guo. 2012. The effect of online consumer reviews on new product sales. *International Journal of Electronic Commerce*, 17(1):39–58.

Jinlan Fu, See Kiong Ng, Zhengbao Jiang, and Pengfei Liu. 2024. Gptscore: Evaluate as you desire. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6556–6576.

F Maxwell Harper and Joseph A Konstan. 2015. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)*, 5(4):1–19.

Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 639–648.

Jin Huang, Harrie Oosterhuis, Maarten de Rijke, and Herke van Hoof. 2020. Keeping dataset biases out of the simulation: A debiased simulator for reinforcement learning based recommender systems. In *RecSys*.

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

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

Dokyun Lee and Kartik Hosanagar. 2021. How do product attributes and reviews moderate the impact of recommender systems through purchase stages? *Management Science*, 67(1):524–546.

Jiacheng Li, Jingbo Shang, and Julian McAuley. 2022. Uctopic: Unsupervised contrastive learning for phrase representations and topic mining. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6159–6169.

Xiaolin Li, Chaojiang Wu, and Feng Mai. 2019. The effect of online reviews on product sales: A joint sentiment-topic analysis. *Information & Management*, 56(2):172–184.

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

Jiaju Lin, Haoran Zhao, Aochi Zhang, Yiting Wu, Huqiuyue Ping, and Qin Chen. 2023. Agentsims: An open-source sandbox for large language model evaluation. *CoRR*, abs/2308.04026.

Ruibo Liu, Ruixin Yang, Chenyan Jia, Ge Zhang, Denny Zhou, Andrew M. Dai, Diyi Yang, and Soroush Vosoughi. 2023. Training socially aligned language models in simulated human society. *CoRR*, abs/2305.16960.

Yinhong Liu, Han Zhou, Zhijiang Guo, Ehsan Shareghi, Ivan Vulić, Anna Korhonen, and Nigel Collier. 2024a. Aligning with human judgement: The role of pairwise preference in large language model evaluators. *arXiv preprint arXiv:2403.16950*.

Yuhan Liu, Xiuying Chen, Xiaoqing Zhang, Xing Gao, Ji Zhang, and Rui Yan. 2024b. From skepticism to acceptance: Simulating the attitude dynamics toward fake news. *arXiv preprint arXiv:2403.09498*.

Yuhan Liu, Yuxuan Liu, Xiaoqing Zhang, Xiuying Chen, and Rui Yan. 2025. The truth becomes clearer through debate! multi-agent systems with large language models unmask fake news. *arXiv preprint arXiv:2505.08532*.

Yuhan Liu, Zirui Song, Xiaoqing Zhang, Xiuying Chen, and Rui Yan. 2024c. From a tiny slip to a giant leap: An llm-based simulation for fake news evolution. *arXiv preprint arXiv:2410.19064*.

Xufang Luo, Zheng Liu, Shitao Xiao, Xing Xie, and Dongsheng Li. 2022. Mindsim: User simulator for news recommenders. In *WWW*.

Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. 2015. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*, pages 43–52.

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

Joon Sung Park, Joseph C. O'Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. 2023b. Generative agents: Interactive simulacra of human behavior. *CoRR*, abs/2304.03442.

Chen Qian, Xin Cong, Cheng Yang, Weize Chen, Yusheng Su, Juyuan Xu, Zhiyuan Liu, and Maosong Sun. 2023. Communicative agents for software development. *CoRR*, abs/2307.07924.

Francesco Ricci, Lior Rokach, and Bracha Shapira. 2010. Introduction to recommender systems handbook. In *Recommender systems handbook*, pages 1–35. Springer.

Noveen Sachdeva and Julian McAuley. 2020. How useful are reviews for recommendation? a critical review and potential improvements. In *proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*, pages 1845–1848.

Gerard Salton and Christopher Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523.

Jing-Cheng Shi, Yang Yu, Qing Da, Shi-Yong Chen, and Anxiang Zeng. 2019. Virtual-taobao: Virtualizing real-world online retail environment for reinforcement learning. In *AAAI*.

Mehdi Srifi, Ahmed Oussous, Ayoub Ait Lahcen, and Salma Mouline. 2020. Recommender systems based on collaborative filtering using review texts—a survey. *Information*, 11(6):317.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. 2023a. Voyager: An open-ended embodied agent with large language models. *CoRR*, abs/2305.16291.

Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, Wayne Xin Zhao, Zhewei Wei, and Ji-Rong Wen. 2023b. A survey on large language model based autonomous agents. *CoRR*, abs/2308.11432.

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

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Shaokun Zhang, Erkang Zhu, Beibin Li, Li Jiang, Xiaoyun Zhang, and Chi Wang. 2023. Autogen: Enabling next-gen LLM applications via multi-agent conversation framework. *CoRR*, abs/2308.08155.

Zhiheng Xi, Wenxiang Chen, Xin Guo, Wei He, Yiwen Ding, Boyang Hong, Ming Zhang, Junzhe Wang, Senjie Jin, Enyu Zhou, and 1 others. 2025. The rise and potential of large language model based agents: A survey. *Science China Information Sciences*, 68(2):121101.

Yuzhuang Xu, Shuo Wang, Peng Li, Fuwen Luo, Xiaolong Wang, Weidong Liu, and Yang Liu. 2023. Exploring large language models for communication games: An empirical study on werewolf. *CoRR*, abs/2309.04658.

Mengyue Yang, Quanyu Dai, Zhenhua Dong, Xu Chen, Xiuqiang He, and Jun Wang. 2021. Top-n recommendation with counterfactual user preference simulation. In *CIKM*.

Ziyi Yang, Zaibin Zhang, Zirui Zheng, Yuxian Jiang, Ziyue Gan, Zhiyu Wang, Zijian Ling, Jinsong Chen, Martz Ma, Bowen Dong, and 1 others. 2024. Oasis: Open agents social interaction simulations on one million agents. *arXiv preprint arXiv:2411.11581*.

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

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, and 1 others. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36:46595–46623.

A Experiment Setting

A.1 Dataset Preprocessing

For the MovieLens (Harper and Konstan, 2015) and AmazonBook (McAuley et al., 2015) datasets, we followed the preprocessing procedures used by Agent4Rec (Zhang et al., 2024). In the case of the GoogleLocal (Li et al., 2022) dataset, we first filtered items related to the restaurant domain within the California region. We then selected users with an interaction history longer than 300 and items having more than 10 interactions. The resulting subset constitutes the GoogleLocal dataset used in our experiments.

A.2 Baseline Details

RecAgent (Wang et al., 2025): RecAgent uses LLM-powered agents, each with a profile, memory, and action module, to simulate diverse user behaviors in a recommendation sandbox environment.

Agent4Rec (Zhang et al., 2024): Agent4Rec simulates users for recommendation systems using LLM-powered agents that have profiles derived from real-world data and perform taste and emotion-driven actions.

SimUSER (Bougie and Watanabe, 2025): SimUSER creates user personas from historical data, then having LLM-powered agents equipped with these personas, perception, memory, and a decision-making brain module interact with the system.

A.3 Specific Hyperparameter Setting in Memory Module

In our implementation of the memory module, we set the hyperparameters α and β to 0.7 and 0.3, respectively. The value of γ is set to 0.2, while θ_p and θ_c are set to 25 and 5, respectively.

A.4 Fine-tuning Setting

The fine-tuning phase was conducted using 4 NVIDIA A100 GPUs. The batch size is set to 8 per device and the learning rate in the cosine learning rate scheduler is 1.0e-4 over three epochs. LoRA was applied to all linear modules while LoRA rank is set to 8 and LoRA alpha is set to 16. The dataset was divided into a training set and a validation set at a ratio of 9:1. We select the model that achieves the best performance on the validation set as the final trained model.

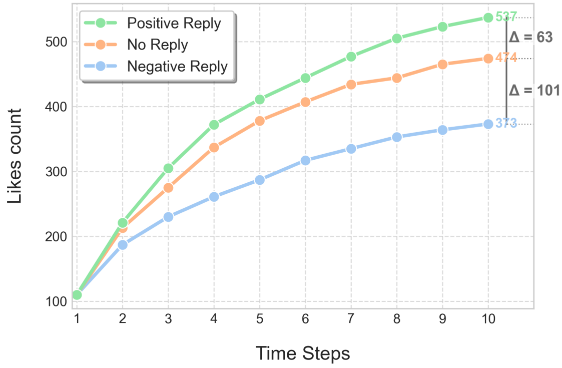

Figure 8: Impact of merchant reply on likes count.

A.5 Base Model Setting

For closed-source large language models, we access them via the official API, configuring the temperature to 0, top-p to 1, and setting both the frequency penalty and presence penalty to 0. For open-source models, we deploy them locally using vllm, and the API parameter configurations are consistent with closed-source models.

B Supplementary Experiments

B.1 Impact of Merchant Reply

Thanks to the design of our interaction mechanism, it becomes feasible to investigate the impact of merchant decision-making on recommendation system simulation. In this experiment, we selected McDonald's as the focal case due to its high level of popularity. We designed three types of merchant response attitudes: (1) merchant who do not respond to user reviews; (2) merchant who actively engage with users in order to defend their brand and interests; and (3) merchant who respond negatively, potentially engaging in verbal conflicts with users. Our objective is to examine how these three distinct response strategies adopted by McDonald's influence the likes of its products within the recommendation system simulation.

As shown in the Figure 8, merchants who actively respond to user reviews can significantly increase the number of likes their store receive. At the end of the simulation, the store from merchant with positive responses received 164 and 63 more likes compared to those with negative or no responses, respectively. This significant difference demonstrates that the responsiveness of merchants has a notable impact on the popularity of their store.

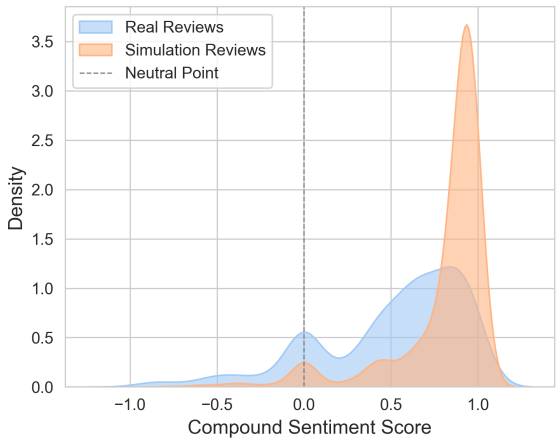

Figure 9: Comparison of compound sentiment score between real reviews and simulation reviews.

B.2 Conservative Behavior

We conducted an analysis comparing reviews generated by simulated users and those from real users. By randomly sampling 1,000 reviews from both the real-world dataset and the simulated dataset, we applied the VADER sentiment analysis tool to compute the Compound Sentiment Scores. The results, as shown in the Figure 9, indicate that both groups of reviews exhibit a similar overall sentiment pattern: predominantly positive, with relatively few negative reviews, and a higher proportion of positive over neutral reviews. However, simulated users demonstrated a significantly higher tendency to leave positive reviews. We attribute this to the ethical alignment mechanisms in large language models, which encourage more cautious and friendly responses, reflecting the conservative behavior of simulated user. Additionally, we generated word clouds for both reviews shown in Figure 10. The results reveal that real user reviews tend to be more colloquial and feature concentrated vocabulary, whereas simulated users prefer more formal and structured expressions. Future work could focus on refining simulated user behavior to better mimic authentic user reviews.

B.3 Items Distribution Comparison

The detailed items distribution comparison between reality and simulation for GoogleLocal dataset is shown in Figure 11.

B.4 Impact of Base Model

We maintained the same configuration as 4.1.1, setting $m = 1$ and experimented with different base models. The results are presented in Table 3.

Figure 10: The word clouds of real reviews and simulation reviews.

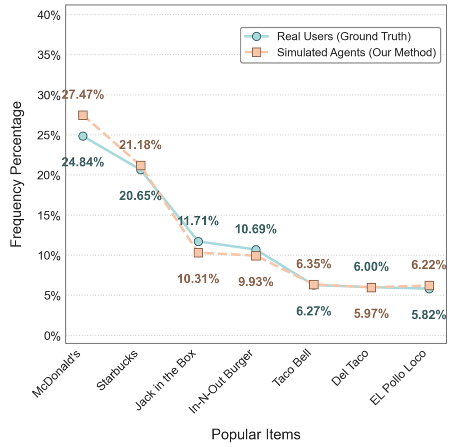

Figure 11: Items distribution comparison between reality and simulation for GoogleLocal dataset.

It can be observed that the choice of base model has a certain impact on the final performance; closed-source models such as GPT-4o generally achieve higher accuracy compared to open-source models. However, our fine-tuned model achieved the best results, indicating that our fine-tuning approach can significantly enhance the simulation accuracy of open-source models.

Model	Accuracy	Precision	Recall	F1 Score
Llama3-8B-Instruct	0.6605	0.6138	0.6697	0.6406
Llama3.1-8B-Instruct	0.6696	0.6214	0.6721	0.6458
Qwen2.5-7B-Instruct	0.6715	0.6229	0.6732	0.6471
Qwen3-8B	0.6824	0.6214	0.6833	0.6509
Gemma2-9B-It	0.6632	0.6172	0.6702	0.6426
Qwen2.5-14B-Instruct	0.6831	0.6342	0.6673	0.6503
GPT-4o-mini	0.6852	0.6307	0.6833	0.6559
GPT-4o	0.6924	0.6319	0.6889	0.6592
GPT-4.1	0.6921	0.6284	0.6899	0.6577
Llama-3-8B-Instruct(finetuned)	0.7128	0.6634	0.7025	0.6824
Qwen2.5-7B-Instruct(finetuned)	**0.7143**	**0.6646**	**0.7057**	**0.6845**

Table 3: Simulation result of **RecInter** with different base models.

B.5 Imapct of Recommendation Algorithm

We investigated the impact of different recommendation algorithms within the **RecInter** in this experiment. Specifically, we experimented with five recommendation algorithms: Random, Most Popular, MF (Koren et al., 2009), LightGCN (He et al., 2020), and MultVAE (Liang et al., 2018). After the simulation ended, we collected statistics on user actions, including the number of Like, Purchase, Review, and Dislike. A higher number of Likes, Purchases, and Reviews, along with fewer Dislikes, is indicative of greater user satisfaction with the platform. The experimental results are presented in the Table 4. As observed, LightGCN achieved the best performance in terms of user satisfaction, while the Random algorithm resulted in the poorest performance. Overall, the effectiveness of the recommendation algorithms exhibited a positive correlation with user satisfaction, aligning well with real-world expectations.

Algorithm	Like	Purchase	Review	Dislike
Random	2297	127	843	455
Pop	2937	195	908	425
MF	3041	273	1038	415
MultVAE	3134	264	1018	373
LightGCN	**3246**	**282**	**1242**	**364**

Table 4: Overall performance of **RecInter** with different recommendation algorithms.

B.6 LLM-Based Evaluation Details

We first run a full simulation, randomly sample 100 agents, and retrieve their associated memories to construct representative agent simulation samples. An LLM then evaluates these samples based on behavioral logic and alignment with the user's profile to assess simulation reliability. Following prior work (Zheng et al., 2023; Liu et al., 2024a; Fu et al., 2024), we adopt a pairwise evaluation strategy using a Judge Agent (GPT-4o). Each pair of samples from different method is evaluated twice with swapped order to reduce bias. A win is counted only if one method is consistently preferred in both orders; otherwise, the result is a tie. Finally, we report the number of Wins, Ties, Losses, and compute the Adjusted Win Rate for comparison:

$$\text{Adjusted Win Rate} = \frac{\text{Win Counts} + 0.5 \cdot \text{Tie Counts}}{\text{Win Counts} + \text{Loss Counts} + \text{Tie Counts}}$$

B.7 Human Evaluation Study for LLM Evaluation

To further substantiate the reliability of our LLM-based evaluation, we conducted a human evaluation study. From the pairs evaluated by the GPT-4o Judge Agent, we randomly selected a subset of 30 distinct pairs of agent simulation samples. These pairs were presented to three human evaluators with expertise in agent-based simulation. The three evaluators were all graduate students conducting research in the field of artificial intelligence. The evaluators were provided with the same instructions and criteria as the Judge Agent: to assess which sample in the pair demonstrated more realistic behavioral logic and better alignment with the associated user profile. Evaluators made independent judgments, and we used a majority vote to determine the human-preferred sample for each pair. We then compared the human consensus judgments with the GPT-4o Judge Agent's decisions for these 30 pairs. As shown in Table 8,We observed a high degree of agreement, with the LLM's judgments aligning with the human majority vote in 26 out of 30 cases (86.7% agreement). This strong correlation between human and LLM evaluation lends significant credibility to our LLM-based evaluation.

Method	Win	Loss	Tie	Win Rate	Loss Rate	Adjusted Win Rate
RecAgent	12	216	72	0.0400	0.7200	0.1600
Agent4Rec	114	77	109	0.3800	0.2567	0.5617
SimUSER	106	54	140	0.3533	0.1800	0.5867
RecInter	143	28	129	0.4767	0.0933	**0.6917**

Table 5: LLM-Based simulation credibility comparison of different methods.

C Case Study

C.1 Subjective Profile Case

A case of Subjective Profile is shown in the Figure 12.

C.2 Inferred Profile Case

A case of Inferred Profile is shown in the Figure 13.

C.3 Agent Response Case

We conducted an analysis of the agent responses after completing the full simulation. The Figure 14 illustrates the response case of simulated 426-th agent at time step 3. Although McDonald's does not align with the agent's dining preferences, the agent ultimately decided to make a purchase due to its high number of positive reviews and strong sales performance. This response case highlights how the interaction mechanism can influence user decision-making again.

- **Taste preferences:**
  - Classic American flavors
  - Fast foodstaples
  - Comfort foods like barbecue
  - Bold and rich seasoning styles
- **Consumption budget_range:** $10-$30
- **Scenario preferences:**
  - Family-friendly restaurants
  - Casual dining
  - Quick service options
- **Consumption habits:**
  - Values convenience
  - Appears to frequent family-friendly or casual chains
  - Occasional focus on traditional or nostalgic favorites

Figure 12: A case of Subjective Profile.

- **Estimated age_range:** 35-50
- **Occupation type:** Middle-management professional
- **Income level:** Middle income
- **Life status:** Possibly married with older children
- **Price sensitivity:** Moderate; values quality but comments on price and value occasionally
- **Quality consciousness:** Moderate; often comments on cleanliness and service standards
- **Service preferences:**
  - Friendly and efficient service
  - Attention to details like cleanliness
  - Proactive assistance
- **Points of concern:**
  - Cleanliness
  - Value for money
  - Service quality
- **Review language style:**
  - Casually critical
  - Detail-oriented
  - Occasionally humorous
  - Objective but firm
  - Expresses nostalgia or disappointment when standards aren't met

Figure 13: A case of Inferred Profile.

Reason: I have a strong preference for high-quality dining experiences, especially in seafood and Italian cuisine. McDonald's and In-N-Out Burger are both fast food options that I find reliable for quick bites, but they don't align with my usual dining preferences. However, the reviews about McDonald's are overwhelmingly positive, highlighting the quality of the food and friendly service. And the sales are also very good. So, I will make a purchase try for it",

Action:

```
[{
   "name": "purchase_product",
    "arguments": {
       "product_id": 4, #McDonald's
       "purchase_num": 1
    }
}]
```

Figure 14: A case of agent response.

D Prompts

D.1 Subjective Profile Prompt

The prompt used in Subjective Profile is shown in Table 6.

D.2 Inferred Profile Prompt

The prompt used in Inferred Profile is shown in Table 7.

Please analyze and summarize the user's profile based on the following interaction history between the user and items. Output in JSON format, including the following aspects :

Taste preferences
Consumption budget range
Price sensitivity
Scenario preferences
Consumption habits

The output must be in the following JSON format:

```
{
  "reason": "Please explain in detail the analysis process and the basis for
your conclusions",
 "profile": {
 "consumption_budget_range": "",
 "scenario_preferences": [],
 "consumption_habits": [],
 "taste_preferences": [],
 }
}
```

Interaction History: {input data}

Table 6: The prompt used in Subjective Profile.

Please thoroughly analyze the interaction history between the user and each item, with particular attention to the specific expressions in the user's reviews. From the reviews and items information, mine and summarize the following:

1. Basic user characteristics:
   Estimated age range
   Possible occupation type
   Estimated income level
   Life status (single/married/with children)

2. Consumption patterns:
   Price sensitivity
   Quality consciousness
   Service preferences
   Points of concern

3. Review language style:
   Summarize the user's language style in reviews (e.g., formal, casual, humorous, critical, concise, detailed, emotional, objective, etc.)

Please form a comprehensive user profile based on this information. When there is insufficient information to make a judgment, please output "unknown" for that item.

The output must be in the following JSON format:

```
{
  "reason": "Please explain in detail the analysis process and the basis for your conclusions",
 "profile": {
 "estimated_age_range": "",
 "possible_occupation_type": "",
 "estimated_income_level": "",
 "life_status": "",
 "price_sensitivity": "",
 "quality_consciousness": "",
 "service_preferences": [],
 "points_of_concern": [],
 "review_language_style": []
 }
}
```

Interaction History with Reviews: {input data}

Table 7: The prompt used in Inferred Profile.

Pair ID	Evaluator 1	Evaluator 2	Evaluator 3	Human Consensus (Majority)	GPT-4o Judge Agent	Agreement (Human vs. GPT-4o)
1	A	A	A	A	A	Yes
2	B	A	B	B	B	Yes
3	A	B	A	A	A	Yes
4	B	B	B	B	B	Yes
5	A	A	B	A	A	Yes
6	B	B	A	B	B	Yes
7	A	A	A	A	A	Yes
8		B	A	B	A	No
9	A	B	A	A	A	Yes
10	B	A	B	B	B	Yes
11	A	A	B	A	A	Yes
12	B	B	A	B	B	Yes
13	A	A	A	A	A	Yes
14	B	B	B	B	B	Yes
15	B	A	B	B	A	No
16	B	A	B	B	B	Yes
17	A	A	B	A	A	Yes
18	B	B	A	B	B	Yes
19	A	A	A	A	A	Yes
20	B	B	B	B	B	Yes
21	A	B	A	A	A	Yes
22	B	A	B	B	B	Yes
23	A	A	B	A	A	Yes
24	B	B	A	B	B	Yes
25	A	A	A	A	A	Yes
26	B	A	B	B	B	Yes
27	A	A	B	A	B	No
28	B	B	B	B	B	Yes
29	A	B	A	A	B	No
30	A	B	A	A	A	Yes
Total					**Agreements:**	**26 / 30 (86.7%)**

Table 8: Detailed human evaluation study results.