Exploring Recommender System Evaluation: A Multi-Modal User Agent Framework for A/B Testing

Wenlin Zhang*
City University of Hong Kong
Hong Kong, China
wl.z@my.cityu.edu.hk

Xiangyang Li*
Huawei Technologies Ltd.
Shenzhen, China
lixiangyang34@huawei.com

Qiyuan Ge
City University of Hong Kong
Hong Kong, China
qiyuange2-c@my.cityu.edu.hk

Kuicai Dong
Huawei Technologies Ltd.
Shenzhen, China
dong.kuicai@huawei.com

Pengyue Jia
City University of Hong Kong
Hong Kong, China
jia.pengyue@my.cityu.edu.hk

Xiaopeng Li
City University of Hong Kong
Hong Kong, China
xiaopli2-c@my.cityu.edu.hk

Zijian Zhang
Jilin University
Changchun, China
zhangzijian@jlu.edu.cn

Maolin Wang
City University of Hong Kong
Hong Kong, China
morin.wang@my.cityu.edu.hk

Yichao Wang†
Huawei Technologies Ltd.
Shenzhen, China
wangyichao5@huawei.com

Huifeng Guo
Huawei Technologies Ltd.
Shenzhen, China
huifeng.guo@huawei.com

Ruiming Tang
Huawei Technologies Ltd.
Shenzhen, China
tangruiming@huawei.com

Xiangyu Zhao†
City University of Hong Kong
Hong Kong, China
xianzhao@cityu.edu.hk

Abstract

In recommender systems, online A/B testing is a crucial method for evaluating the performance of different models. However, conducting online A/B testing often presents significant challenges, including substantial economic costs, user experience degradation, and considerable time requirements. With the Large Language Models' powerful capacity, LLM-based agent shows great potential to replace traditional online A/B testing. Nonetheless, current agents fail to simulate the perception process and interaction patterns, due to the lack of real environments and visual perception capability. To address these challenges, we introduce a multi-modal user agent for A/B testing (A/B Agent). Specifically, we construct a recommendation sandbox environment for A/B testing, enabling multimodal and multi-page interactions that align with real user behavior on online platforms. The designed agent leverages multimodal information perception, fine-grained user preferences, and integrates profiles, action memory retrieval, and a fatigue system to simulate complex human decision-making. We validated the potential of the agent as an alternative to traditional A/B testing from three perspectives: model, data, and features. Furthermore, we found that the data generated by A/B Agent can effectively enhance the capabilities of recommendation models. Our code is publicly available at https://github.com/Applied-Machine-Learning-Lab/ABAgent.

CCS Concepts

• **Information systems** → **Recommender systems**.

Keywords

Recommender System, Multimodal User Agent, A/B Testing

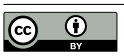
ACM Reference Format:

Wenlin Zhang, Xiangyang Li, Qiyuan Ge, Kuicai Dong, Pengyue Jia, Xiaopeng Li, Zijian Zhang, Maolin Wang, Yichao Wang, Huifeng Guo, Ruiming Tang, and Xiangyu Zhao. 2026. Exploring Recommender System Evaluation: A Multi-Modal User Agent Framework for A/B Testing. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.1 (KDD 2026), August 9–13, 2026, Jeju Island, Republic of Korea.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3770854.3785688

1 Introduction

In real-world industrial settings, recommender systems, spanning diverse paradigms from reinforcement learning [26, 51, 53, 54] and sequential modeling [10, 20, 25] to emerging large model-based approaches [9, 47], often rely on online A/B testing to optimize and evaluate model performance in real time [8, 19, 27, 43]. In A/B testing pipeline, users are randomly divided into experimental and control groups, with the experimental group receiving recommendations from a new system, while the control group uses the existing system as a baseline. However, despite its advantages, A/B testing has several challenges: 1) High resource consumption: It requires significant server resources and data analysis efforts, especially for high-traffic products [11]. 2) User experience degradation:

*Contributed equally to this work.
†Corresponding authors.



KDD 2026, Jeju Island, Republic of Korea.

ACM ISBN 979-8-4007-2258-5/2026/08
https://doi.org/10.1145/3770854.3785688

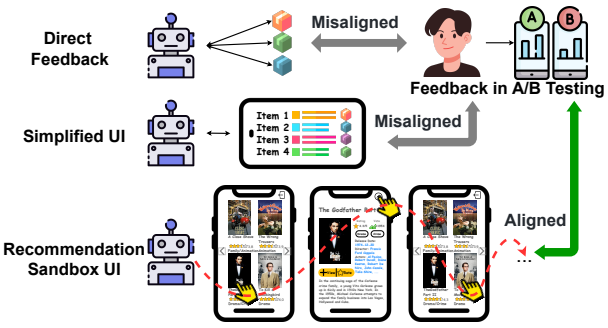


Figure 1: Comparison of alignment between simulated user feedback and real user feedback in A/B testing.

Frequent A/B testing can disrupt user interaction and lower satisfaction [21]. 3) Evaluation latency : Collecting sufficient data for statistical significance can delay timely evaluation. Therefore, it is essential to develop reliable offline evaluation methods to simulate A/B testing results.

However, replicating the complex dynamics of A/B testing poses significant challenges for user simulator design. In real-world settings, users interact with multimodal content across various interfaces, actively explore items of interest, disengage when fatigued, and generate feedback across multiple interaction stages. Capturing these behaviors accurately is crucial for building effective user simulators that reflect real-world evaluation dynamics.

With the emergence of Large Language Models (LLMs), LLM-based agents have garnered significant attention. LLM agents have broad world knowledge, require less scenario-specific training, and enable flexible, interpretable interactions through natural language [24, 37, 38, 48, 58]. For instance, iEvaLM [41] employs an LLM-based user agent to evaluate conversational recommender systems, providing flexible natural language interactions. RecAgent [39] designs a Web recommendation scenario where user agents can interact with each other, exploring the impact of human social behavior on recommendation results. Agent4Rec [46] designed a user agent for simulating page-by-page movie recommendations. These works demonstrate that LLM-based agents have the promising capability to simulate user behaviors.

Although existing work has achieved active interaction between agents and recommendation environments, there remains a significant gap between simulated paths and real human perception process. Figure 1 illustrates the simulation methods used in existing approaches. They either directly simulate user-item interaction feedback or model user behavior in simplified, text-based UI environments. Apparently, they fail to accurately reflect real user behavior on recommendation platforms. Moreover, the agents lack perception of rich multimodal information [6] and multi-layered interface simulation, which limits their ability to replicate human interactions. To address this gap, we propose A/B Agent to simulate human perception process and interaction path more effectively.

To design A/B Agent, two challenges need to be tackled. **(1) The gap between the simulated environment and the online platform UI.** Existing simulated environments directly present only textual item information to users, overlooking the fact that users in an actual online platform UI obtain multimodal information at different granularities and explore progressively. To simulate the process of how users perceive and interact with the UI on an online platform, we crawled multimodal movie data and constructed a movie recommendation sandbox environment similar to IMDB. **(2) Modeling user behavior trajectories under multimodal and multi-level information settings.** In the sandbox environment, users interact with movie information at varying levels of granularity across different interfaces, which results in complex behavior trajectories. However, existing user agent designs struggle to achieve fine-grained perception and human-like exploration. To address these challenges, we developed A/B Agent, an agent that simulates human perception and exploration more effectively. For perception, A/B Agent captures detailed user preferences across diverse levels of movie information granularity, integrating both image and text modalities. For exploration, A/B Agent incorporates a long-term and short-term memory module and a fatigue system to avoid repeated exploration or overexploitation.

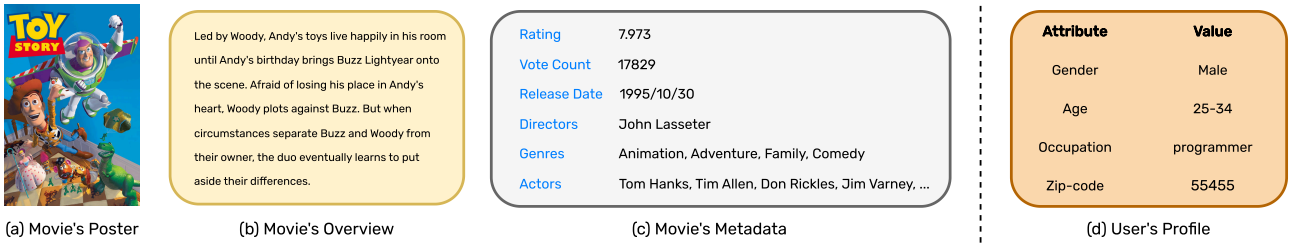
Finally, to verify the effectiveness of the agent's simulation in the sandbox recommendation environment, we conducted A/B testing using the agent for different recommendation algorithms. Furthermore, we collected feedback data from the agent during the interaction process for data augmentation experiments to validate the impact of simulated data on model improvement. Experimental results demonstrate that A/B Agent can emulate user interaction patterns in the interactive recommendation environment. Our key contributions can be summarized as follows:

- We propose A/B Agent to simulate the entire human perception process and interactive behavior trajectories for A/B testing. The agent possesses multimodal information perception and can perform human-like exploration within the sandbox recommendation environment.
- We develop an interactive sandbox recommendation environment, where the agent retrieves movie information at varying levels of granularity across different interfaces, enabling it to perform multi-interface exploration.
- We create a large-scale multimodal dataset MM-ML-1M by extending movies' meta-information and posters. This dataset can provide the necessary data for different interfaces within the interactive sandbox recommendation environment.
- We conduct extensive experiments to evaluate the agent's simulation capabilities within the sandbox environment. Both A/B testing for recommendation models and data augmentation based on agent feedback demonstrate the effectiveness of A/B Agent.

2 A/B Agent Framework

2.1 Framework Overview

As shown in Figure 4 the A/B Agent framework is composed of three primary components: 1) **MM-ML-1M Dataset**: This comprehensive dataset includes images, text, and various movie metadata, providing a realistic basis for simulating authentic movie recommendation scenarios. 2) **Recommendation Sandbox Environment**: This environment offers a wide range of popular recommendation models and a rich user interface, enabling interaction and exploration by the User Agent in a simulated real-world setting. 3) **A/B Agent**: Designed to emulate user behavior patterns in a realistic movie



(a) Movie's Poster (b) Movie's Overview (c) Movie's Metadata (d) User's Profile

Figure 2: Data example from Multimodal-Movielens-1M dataset.

Table 1: Dataset statistics in MM-ML-1M. Note: the range for text type is the length of text content.

Feature	Type	Count	Range
Title	Text	3,822	[2, 72]
Overview	Text	3,814	[13, 991]
Genres	Text	3,789	[5, 64]
Rating	Numerical	3,822	[0, 10]
Vote Count	Numerical	3,822	[0, 30,002]
Release Date	Date	3,820	[1911-05-05, 2024-06-07]
Directors	Text	3,810	[3, 172]
Actors	Text	3,785	[8, 5,101]
Poster	Image	3,814	-
User Count	-	6040	-
Movie Count	-	3952	-
Sparsity	-	0.0419	-

recommendation environment, it consists of several key components, including the agent profile module, memory module, action module, and fatigue system. These systems enable the agent to deliver feedback akin to human responses.

2.2 Dataset: MM-ML-1M

Most existing recommendation datasets fall short in effectively simulating real-world interaction scenarios. For instance, Amazon lacks user attribute information, MovieLens [13] is devoid of multimodal data, and many datasets like Criteo [56] and Avazu [56] have anonymized feature characteristics. To address these limitations, we introduce the Multimodal-MovieLens-1M (MM-ML-1M) dataset, an extension of the original MovieLens-1M [13], enhanced with additional movie posters and metadata. Figure 2 provides a detailed illustration of the data structure. Table 1 summarizes the statistics of the MM-ML-1M dataset, including the type, count, and value range for each feature. For text fields, the range denotes the character length. The dataset contains 6,040 users and 3,952 movies, with an interaction sparsity of 4.19%. MM-ML-1M enriches the movie-side information with elements such as posters, overviews, and metadata, including IMDb ratings, vote counts, directors, and actors, while maintaining the original user-side information. These enhancements offer crucial visual [7] and contextual data, improving recommendation simulations by capturing factors like movie popularity and creator preferences. This comprehensive dataset facilitates the development and evaluation of recommendation models that more accurately reflect real-world user interactions.

2.3 Recommendation Sandbox Environment

To simulate a realistic movie recommendation environment, we design a multimodal interactive user interface that emulates popular platforms such as Netflix and IMDb, as illustrated in Figure 3.

2.3.1 ***User Interface Platform.*** We design a recommendation user interface (UI) to simulate a real-world movie environment, featuring a home page and a movie detail page. Users can seamlessly navigate between these pages, each providing different levels of movie details and unique interactive options.

Home Page. As depicted in Figure 3, the home page serves as the initial interface users encounter upon visiting the site. Here, users can view concise information about each movie, including the poster, title, rating, and genre. This interface enables users to efficiently browse through multiple movies and select those of interest. Available actions on this page include navigating to the next or previous page and clicking on a movie to access more detailed information.

Movie Detail Page. Upon selecting a movie of interest, users are directed to the Movie Detail Page, as shown in Figure 3. This dedicated page offers comprehensive information about the chosen movie, significantly enriching the user's understanding through a plot overview and detailed metadata such as vote count, release date, director, and cast, in addition to the information available on the home page. A high-resolution poster is also provided. This extensive and granular information enables users to make informed decisions about whether to watch the movie or return to the home page. Users have the option to watch the movie, rate it, or navigate back to the home page.

2.3.2 ***Integration with Recommendation Algorithms.*** The environment supports the integration of various recommendation algorithms, as shown in Figure 3, offering scalability for developing and optimizing recommender systems. It includes collaborative filtering algorithms such as random recommendation, popularity-based models [36], Factorization Machines (FM) [31], DeepFM [12], AFN [4]. Model performance is evaluated using Click-Through Rate (CTR, the ratio of clicks to impressions on the home page), Conversion Rate (CVR, the ratio of movie detail page views), and Average Rating (AR) data collected from user simulation feedback.

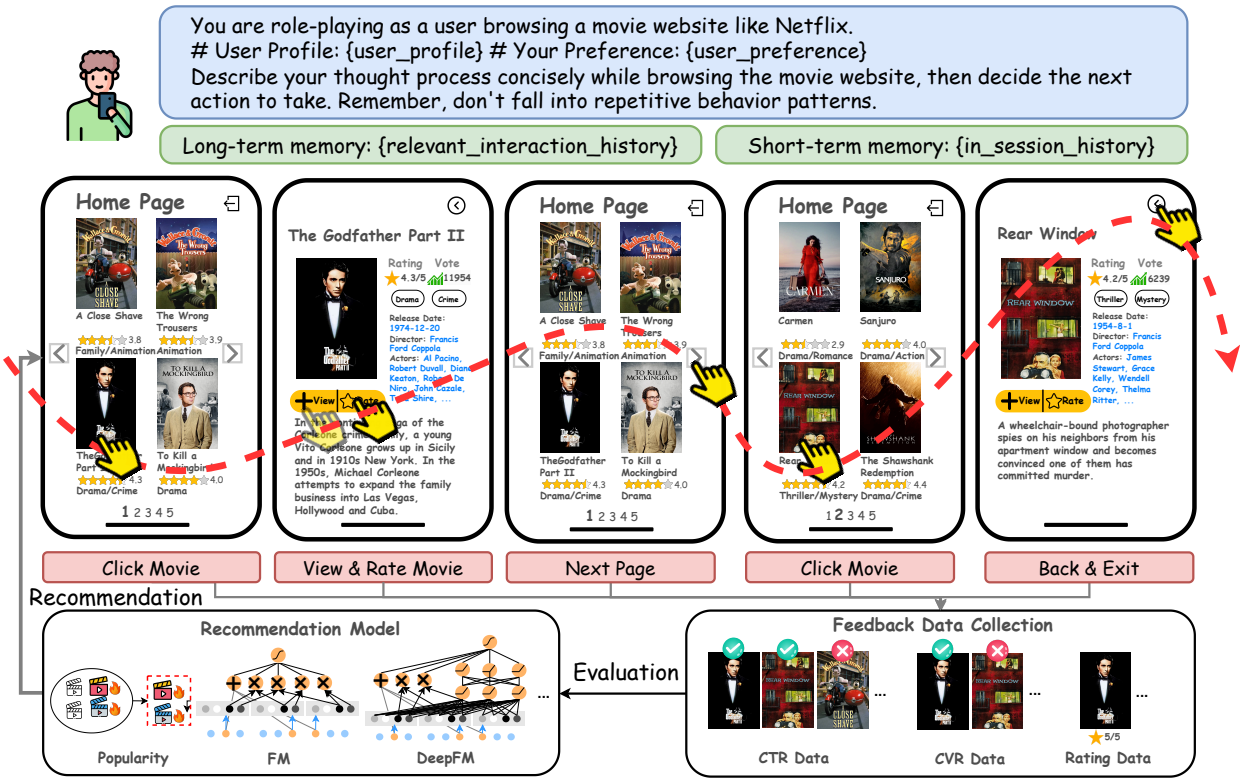


Figure 3: The recommendation sandbox environment comprises two key components: recommendation algorithms and a user interface. The recommendation algorithms generate recommendation lists displayed to users on the home page and individual movie detail pages. User interaction with this interface, including click-through rate (CTR), conversion rate (CVR), and ratings, provides data for recommendation model evaluation.

2.4 A/B Agent Architecture

A/B Agent simulates user interaction patterns in the recommendation sandbox environment. It comprises several key components, including the agent profile module, memory module, action module, and fatigue system. The overall agent design framework is shown in Figure 4.

2.4.1 ***Profile Module.*** To facilitate the agent's ability to perceive varying granularities of movie information, it is essential for the agent profile to delineate fine-grained preferences that inform the agent's core behavior patterns and decision-making logic [23]. The profile module consists of two key elements: the user profile and user preferences. The user profile includes demographic details such as gender, occupation, age, and location. The user preferences tailored to specific movie characteristics are summarized by LLMs based on the user's interaction history through prompting.

2.4.2 ***Memory Module.*** The memory module is essential for retaining interaction history and supporting decision-making in recommendation systems [15]. While existing memory designs provide a basic framework, they neglect visual modality retrieval [46]. To achieve the multimodal perception, we designed a memory mechanism that integrates both textual and visual retrieval. The memory module is composed of long-term and short-term memory, where long-term memory stores historical interaction records, and short-term memory captures interactions within the current session.

Long-Term Memory. To model persistent user preferences beyond the current session, we design a long-term memory module that stores and retrieves historical interaction data across sessions. This component captures stable interests over time and complements the short-term memory, which focuses on in-session context. (1) *Textual Retrieval:* We use the `text-embedding-3-small` model [28] to encode rich movie meta-information into textual embeddings $\boldsymbol{e}_{\text{text}}$. When exposed to new content, the agent formulates textual queries $\boldsymbol{q}_{\text{text}}$ based on the interface description and retrieves semantically aligned records from past sessions, reinforcing consistent long-term preferences. (2) *Visual Retrieval:* We further incorporate visual memory by encoding movie posters using CLIP [30] into visual embeddings $\boldsymbol{e}_{\text{image}}$. The agent generates visual queries $\boldsymbol{q}_{\text{image}}$ informed by elements such as color tones and object layouts, and retrieves relevant visual memories via cosine similarity. By integrating both textual and visual retrieval mechanisms, the long-term memory module robustly supports human-like interaction recall across modalities and sessions, thereby enabling stable and consistent user modeling over extended periods.

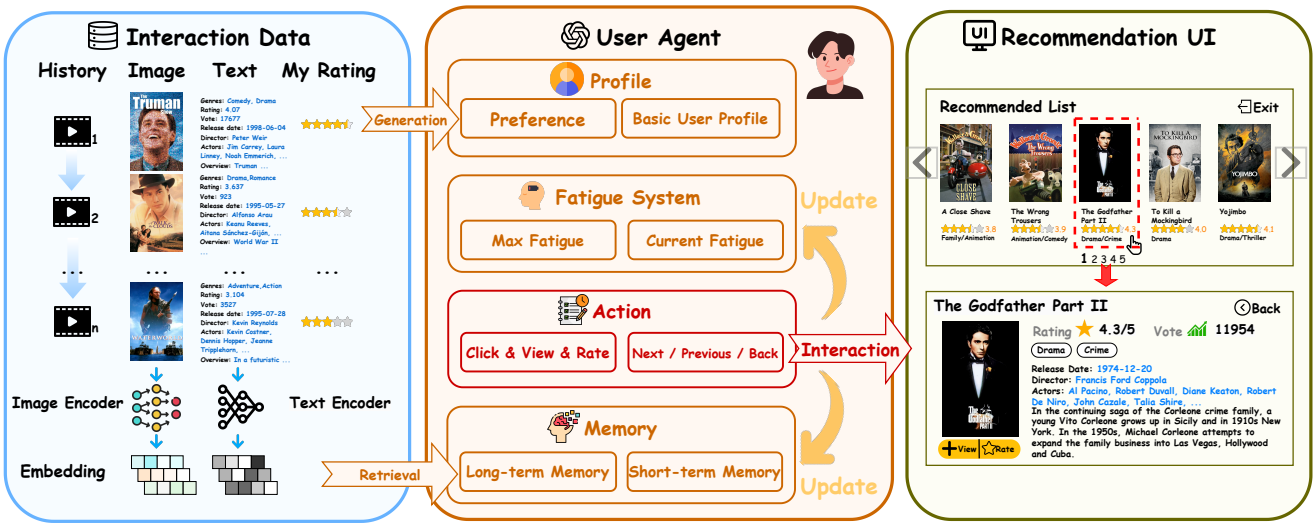


Figure 4: The Framework of A/B Agent. Our agent design involves three components: multimodal User Agent (Orange Section), Recommendation UI (Green Section), and Interaction Data (Blue Section). The Recommendation UI provides a multimodal, multi-interface sandbox for the agent. Based on Interaction Data, the agent initializes user preferences and retrieves relevant memories. The multimodal User Agent simulates multi-page, multimodal information processing and decision-making behavior based on modules including profile, action, memory, and fatigue system.

Short-Term Memory. To support context-aware decision-making within a single session, we design a short-term memory module that continuously tracks all in-session interactions between the agent and the environment. Unlike long-term memory, which captures cross-session preferences and aggregates persistent behavioral patterns, this component focuses specifically on in-context observations and moment-to-moment behaviors to maintain coherence and avoid redundancy during exploration. Each interaction is stored in a structured format, including the interface type (e.g., homepage, movie detail page), local observations (e.g., whether certain items attract attention), estimated interest levels (on a 1–5 scale), and the corresponding action taken by the agent. This design enables the agent to reason over recent events, refer to its recent decisions, and dynamically adjust future actions accordingly, thereby better mimicking the sequential and adaptive nature of human short-term awareness and decision-making during a recommendation session.

2.4.3 ***Action Module.*** The action module defines the agent's workflow and permissible actions within the recommendation environment [46], adapting to diverse interactive interfaces. The agent's workflow involves retrieving relevant memories, analyzing the current page, and contextually determining its next action. After each action, the agent's memory is updated, and the environment responds with page transitions and updated content. Actions are interface-specific. For example, on the home page, the agent can *click* a movie for details or navigate using *next page* or *previous page*. On the movie detail page, the agent can *view*, *rate* or *back*.

2.4.4 ***Fatigue System.*** Although the agent can achieve personalized feedback with the recommendation environment based on its profile, memory, and action modules, it often faces the issue of excessive exploration across multiple interfaces, leading to inconsistency with real user behavior. To address this discrepancy, we propose a fatigue system.

Specifically, the agent starts each session with an initial fatigue value. Each action consumes a certain amount of fatigue, which the agent considers when selecting actions. When the agent's fatigue value reaches zero, it will actively exit the recommendation environment. We categorize actions based on the frequency with which real users perform them as follows: (1) High-frequency actions, which include browsing behaviors such as previous page, next page, exit, and back; (2) Medium-frequency actions, which involve clicking to explore movies of interest; and (3) Low-frequency actions, which consist of watching and rating movies to confirm interest. The fatigue cost is determined by two factors including the type of action and the agent's level of interest in the current page, which can be defined as

$$F = C_a \cdot \left(\phi_{\max} - \frac{(\iota - \iota_{\min})(\phi_{\max} - \phi_{\min})}{\iota_{\max} - \iota_{\min}} \right), \tag{1}$$

where: F is the computed fatigue cost, C_a represents the base fatigue coefficient for the action, $\phi_{\max}$ and $\phi_{\min}$ are the maximum and minimum fatigue modifiers, respectively, ι denotes the current interest level, $\iota_{\max}$ and $\iota_{\min}$ are the maximum and minimum interest levels, respectively.

The accumulated fatigue value influences the agent's behavior. As fatigue increases, the agent becomes more likely to engage in less demanding activities or to completely exit the session.

Table 2: Performance of recommendation model evaluation within the A/B Agent framework, across ML and Fashion datasets, including Gemini-2.5-flash as an additional backbone. The best result is bolded, and the second-best is underlined.

Dataset	Model	A/B Agent (GPT-4o)			A/B Agent (GPT-4o-mini)			A/B Agent (Gemini-2.5-flash)			Real-World	
		CTR	CVR	AR	CTR	CVR	AR	CTR	CVR	AR	Recall	NDCG
ML	Random	0.2330	0.1147	4.27	0.0872	0.0461	4.01	0.2905	0.1863	4.01	0.0066	0.0222
	Pop	0.3077	0.1835	4.30	0.1886	0.1181	4.20	0.4520	0.2920	4.12	0.0216	0.0881
	FM	0.3635	0.2940	4.70	0.2642	0.1984	4.52	0.5000	0.3855	4.43	0.0353	0.0987
	AFN	0.4301	0.3385	4.72	0.2845	0.1995	4.52	0.5313	0.4225	4.52	0.0422	**0.1238**
	DeepFM	**0.4453**	**0.3458**	**4.75**	**0.2891**	**0.2094**	**4.52**	**0.5417**	**0.4333**	**4.57**	**0.0429**	0.1130
Fashion	Random	0.0158	0.0073	4.42	0.0031	0.0008	4.12	0.0426	0.0093	4.40	0.0079	0.0028
	Pop	0.0561	0.0098	4.46	0.0212	0.0008	4.32	0.0937	0.0158	4.44	0.0157	0.0059
	FM	0.0612	0.0108	4.50	0.0311	0.0067	4.5	0.1052	0.0285	4.50	0.0591	0.0212
	AFN	0.0754	0.0158	4.50	0.0407	0.0081	4.50	0.1337	0.0341	4.52	0.0748	0.0259
	DeepFM	**0.0772**	**0.0165**	**4.53**	**0.0414**	**0.0089**	**4.51**	**0.1409**	**0.0385**	**4.60**	**0.0787**	**0.0281**

3 Experiment

3.1 Experimental Setting

Task and Evaluation Protocol. We evaluate A/B Agent on personalized recommendation tasks in two domains: MovieLens and Amazon Fashion. User-item interactions are split chronologically into training, validation, and test sets (7:2:1). The training set is used to build recommendation models and initialize user profiles. Evaluation is conducted from two perspectives. In the sandbox environment, we measure simulation performance using Click-Through Rate (CTR), Conversion Rate (CVR), and Average Rating (AR), which reflect the agent's ability to mimic human behavior. To assess simulation fidelity, we compare simulated outcomes with real user feedback metrics (Recall@20 and NDCG@20 [18]) computed on the held-out test set. The closer the simulation aligns with real-world performance, the higher the fidelity.

Simulation Environment. We build a sandbox environment that mimics real-world platforms (e.g., IMDB, Amazon), where each item includes multimodal attributes (title, genre, poster, and description). At each step, the agent views 5 items from a top-20 recommendation list and makes interaction decisions based on perceived relevance.

Implementation Details. User preferences are summarized using GPT-4o [17], and we test three backbone LLMs: GPT-4o, GPT-4o-mini, and Gemini-2.5-Flash. The agent uses *text-embedding-3-small* for memory retrieval and CLIP [30] for visual encoding. DeepFM serves as the recommendation model, with ablations on data scale and feature sets.

3.2 Recommendation Model Evaluation

To evaluate how effectively A/B Agent simulates A/B testing within the recommender system A/B testing, we conducted A/B testing experiments from three perspectives: model comparison, data scale impact, and feature importance.

3.2.1 ***Model Comparison.*** Table 2 reports simulation results across two domains (MovieLens and Amazon Fashion), using three backbone LLMs: GPT-4o, GPT-4o-mini, and Gemini-2.5-Flash. The experiments demonstrate that A/B Agent is effective in evaluating recommendation models under diverse configurations. Key observations are summarized below: (1) **Reliable performance ranking**: A/B Agent captures consistent model performance ordering aligned with offline evaluation. Specifically, Pop outperforms Random due to the popularity-based strategy, FM outperforms both, and DeepFM and AFN achieve the best results with close performance. This indicates that A/B Agent can reflect meaningful preference differences among models. (2) **Cross-domain generalization**: On Amazon Fashion, simulation results remain consistent with real-world trends, suggesting that A/B Agent generalizes well beyond the movie domain to diverse recommendation settings. (3) **Backbone robustness**: All three backbone LLMs lead to comparable model ordering, confirming that A/B Agent is not sensitive to the specific choice of language model and can be flexibly deployed with various LLM infrastructures. (4) **Multi-metric consistency**: A/B Agent maintains consistent model ranking across CTR, CVR, and average rating. This validates its ability to simulate realistic user behavior beyond simple user-item feedback, especially in recommendation sandbox environments with complex interaction flows.

3.2.2 ***Data Scale Impact.*** Table 3 demonstrates the simulation results of DeepFM trained with three different proportions of data: 50%, 75%, and 100%. This experiment evaluates whether A/B Agent can reflect performance variations arising from training data scale. The observation can be summarized as follows: (1) **Monotonic improvement**: As the training data increases, all three metrics(i.e. CTR, CVR, and average rating) consistently improve. This aligns with real-world evaluation results. (2) **Reliable differentiation**: A/B Agent successfully distinguishes the performance differences across data scales, indicating that the agent can capture fine-grained model improvements induced by larger datasets.

3.2.3 ***Feature Importance Evaluation.*** Table 4 presents a feature ablation study on the DeepFM model, where we assess the impact of different input feature sets on recommendation performance. Specifically, we train three variants: (1) *User ID Only*, where user-side features (gender, age, occupation, zip) are removed; (2) *Movie ID Only*, where the item-side genre feature is excluded; (3) *All*

Table 3: Performance of DeepFM with various training data scale within the A/B Agent Framework.

Evaluation	A/B Agent (GPT-4o-mini)			Real-World	
Training Data	CTR	CVR	AR	Recall	NDCG
50%	0.2205	0.1803	4.51	0.0275	0.0738
75%	0.2745	0.1999	4.51	0.0330	0.0918
100%	**0.2891**	**0.2094**	**4.52**	**0.0429**	**0.1130**

Table 4: Performance of DeepFM with various features within the A/B Agent Framework.

Evaluation	A/B Agent (GPT-4o-mini)			Real-World	
Feature	CTR	CVR	AR	Recall	NDCG
User ID Only	0.2754	0.1982	4.47	0.0359	0.0981
Movie ID Only	0.2850	0.2097	4.51	0.0372	0.0966
All	**0.2891**	**0.2094**	**4.52**	**0.0429**	**0.1130**

Features, where all user and item features are retained. Key observations are summarized below: (1) **Performance degradation under ablation**: Models trained with reduced feature sets exhibit consistent performance drops across all metrics (CTR, CVR, and average rating), confirming the importance of both user-side and item-side auxiliary features in model learning. (2) **User-side features are more informative**: The *Movie ID Only* model performs better than the *User ID Only* variant in CTR and CVR, suggesting that user-related features contribute more significantly to recommendation accuracy in this setting. (3) **A/B Agent can detect fine-grained differences**: A/B Agent successfully reflects these performance gaps, demonstrating its ability to distinguish subtle variations in model input and validate feature effectiveness through simulation.

These results support the utility of A/B Agent as a reliable simulation framework for evaluating recommender system performance across various aspects, including model differentiation, data scale impact, and feature set importance.

3.3 Agent Alignment

3.3.1 ***User Taste Alignment.*** To validate the alignment between simulated preferences and real user behavior, we conduct a user taste alignment study. The hypothesis is that if the simulator captures real user interests, it should show stronger acceptance for recommendation lists containing a higher proportion of ground-truth positive items. We construct three sets of recommendation lists with positive-to-negative ratios of 1:9, 1:4, and 1:1, respectively. For each user, we include 20 items composed of real clicked movies and sampled negatives. We then evaluate how A/B Agent responds to each recommendation. As shown in Figure 5, the agent exhibits consistently higher CTR, CVR, and AR under recommendation lists with more positive items, suggesting strong alignment with real user preferences.

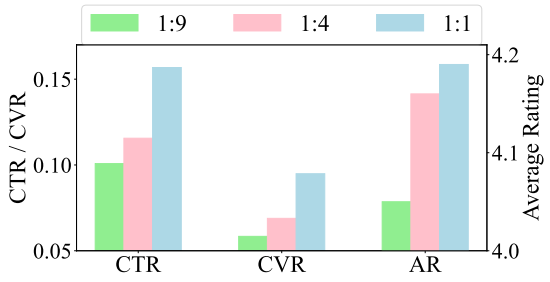


Figure 5: Impact of positive recommendation ratio on CTR, CVR, and Rating.

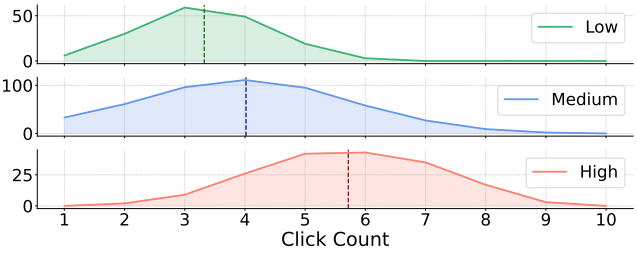


Figure 6: Click distribution among user groups with different activity traits.

3.3.2 ***Activity Trait Alignment.*** We assign each A/B Agent an activity trait (low, medium, or high) to control its overall engagement level. To verify whether the agent's behavior aligns with its assigned trait, we analyze the distribution of movie clicks under each setting. As shown in Figure 6, agents with different activity levels exhibit distinct click distributions. All groups roughly follow a Gaussian distribution, but the mean number of clicks shifts from 3.2 (low) to 4.0 (medium) and 5.8 (high). This consistent increase in click volume confirms that the agent's behavior aligns well with its assigned activity trait, demonstrating the effectiveness of our fatigue-aware design in modulating user engagement.

3.4 Data Augmentation

To assess the effectiveness of A/B Agent-generated interaction data, we simulate agent behaviors under DeepFM-based recommendations and collect two types of signals: 2,518 homepage clicks and 1,884 viewing records from movie detail pages. Although these simulated samples (4K) are significantly smaller than the original training set (700K), they are integrated with the original dataset to train downstream recommenders. Table 5 presents the offline AUC performance across eight representative models, including NFM [14], xDeepFM [12], Wide&Deep [3], DCN [40], DeepFM [12], AFN [4], AutoInt [35], and PNN [29]. We observe consistent performance gains across all models, especially when using simulated data from vision-aware A/B Agent. On click data, AUC improves by more than 0.002 across most models, with Wide&Deep achieving the highest gain of 0.0032. On view data, the improvements are even more pronounced: NFM, xDeepFM, and DeepFM achieve gains of 0.0037, 0.0039, and 0.0022, respectively. AFN, AutoInt, and PNN all

Table 5: The AUC comparison between various models for data augmentation experiment. It is worth noting that an AUC increase of 0.001 can be considered a significant improvement in CTR prediction [22] The best result is bolded, and the second-best is underlined.

Model	NFM	xDeepFM	Wide&Deep	DCN	DeepFM	AFN	AutoInt	PNN
Original	0.7438	0.7478	0.7469	0.7517	0.7520	0.7512	0.7510	0.7474
+Sim. Click Data(w/o vision)	0.7458	0.7482	0.7485	0.7533	0.7501	0.7502	0.7527	0.7503
+Sim. View Data(w/o vision)	0.7455	0.7486	0.7482	0.7534	0.7528	0.7543	0.7532	0.7518
+Sim. Click Data(w/ vision)	0.7464	0.7502	0.7501	**0.7539**	0.7541	0.7530	0.7531	0.7499
+Sim. View Data (w/ vision)	**0.7475**	**0.7517**	**0.7521**	0.7537	**0.7542**	**0.7556**	**0.7543**	**0.7520**

show their best results with vision-aware view data. These results demonstrate that even a small amount of high-quality simulation data can lead to measurable improvements in model performance, highlighting the utility of A/B Agent in enhancing recommendation systems through efficient data augmentation.

3.5 Ablation Study

To evaluate the contribution of visual modality in agent behavior simulation, we conduct an ablation study by removing poster images from the UI interface during simulation. Under this setting, agents receive the same recommendation list (from DeepFM) but without visual exposure to movie posters. The resulting interaction data, denoted as "(w/o vision)" in Table 5, is used for training downstream recommenders. As shown in the table, while vision-agnostic simulation still leads to marginal improvements over the original training set, the gains are consistently smaller compared to vision-aware settings. For example, with simulated click data, DeepFM improves from 0.7520 to 0.7541 with visual input, while only reaching 0.7501 without it. Similar trends hold across other models and interaction types. These results confirm that visual modality plays a critical role in producing realistic user-agent behavior. The agent's interaction patterns are better aligned with user preferences when visual information is available, leading to more effective data augmentation.

3.6 Case Study

To qualitatively evaluate the effectiveness of A/B Agent in simulating user behavior under a multi-modal, multi-interface recommendation sandbox, case analysis for agent simulation are illustrated in Table 6. These cases illustrate key actions (click, next page, watch & rate, exit), showing how agent behavior aligns with preferences, historical patterns, and fatigue. Case 1 demonstrates a click decision where the agent, preferring comedy, family, and animation, is attracted to the recommended item based on its genre and vibrant poster. The agent's action aligns with its multi-modal preferences. Case 2 shows the agent skipping a page. Despite the high rating of The Shawshank Redemption, the agent dislikes drama and this is supported by low historical ratings for similar films. The decision is driven by genre disinterest. Case 3 presents the agent watching and rating a movie 5.0. The agent's preference for drama and romance is aligned with the recommendation, and past experiences reinforce the high rating. Case 4 demonstrates an exit behavior due to a mismatch in preferences and fatigue. Although the recommended movie matches the agent's adventure preference, its release year (1962) diverges from the agent's preference for 80s-90s films. High fatigue (25.4/30) also leads the agent to exit. These cases confirm that A/B Agent effectively simulates nuanced user behaviors, accounting for preferences, rejection cues, and fatigue-induced disengagement.

3.7 Comparison with Existing Simulators

Table 7 summarizes representative user simulators across simulation interface, modality, and evaluation. Early works like RecoGym [32] and RecSim [16] adopt simple direct-response setups with limited interaction fidelity. Later simulators such as VirtualTaobao [34], Adversarial [2], KuaiSim [50], and Agent4Rec [46] introduce richer interaction flows, but remain text-only and overlook visual signals prevalent in real-world platforms. LLM-SimuRec [49] leverages LLMs for text-based simulation but lacks UI realism and multi-modal context. In contrast, A/B Agent integrates a sandbox-style UI with full multi-modal support, including posters and metadata. It enables more realistic and explainable user modeling, and is evaluated using both offline metrics and case-based analysis. By bridging the gap between interaction fidelity and modality grounding, A/B Agent sets a new standard for user simulation in recommendation research.

4 Related Work

4.1 Traditional User Simulator

Traditional user simulators set the behavior mode based on rules, or use GAN and reinforcement learning to model user behavior [1, 2, 16, 32, 34]. RecSim [16] configures the simulation environment based on rules, including user preferences, user status, item similarity, and recommendation models, *etc.* to simulate the sequential interaction between users and items. UserSim [52] uses GAN to train user simulators, uses generators to capture the distribution of user historical log behavior, and uses discriminators to distinguish between real and fake user logs. Traditional user simulators have two main drawbacks: 1) They rely on predefined rules or require large amounts of data to train user simulators, which can be resource-intensive and less adaptable; 2) They often lack interpretability, making it difficult to understand the reasoning behind simulated user behaviors.

Table 6: Case analysis of A/B Agent simulation: green highlights preference/acceptance-related info; red highlights rejection/disinterest cues

Target Items	 Tom and Jerry: Willy Wonka Animation, Comedy, Family Date: 2017-06-28 3.4	 The Shawshank Redemption Drama, Crime Date: 1994-09-23 4.4	 Central Station Drama, Family Date: 1998-11-20 4.1	 Lawrence of Arabia Adventure, War, History Date: 1962-12-11 4.0
Action	Click	Next Page	Watch & Rate 5.0	Exit
Profile	Prefers Comedy, Family, Animation. Colorful posters	Limited interest in drama	Prefers Drama and Romance (5 stars);	Prefers Adventure films from the 80s and 90s
Memory	Rated *The Addams Family* 4.0 Rated *Last Action Hero* 4.0	Rated *Solo* 2.0 Rated *King Kong* 3.0	Rated *Breaking the Waves* 5.0 Rated *Three Colors: Red* 5.0	Click *The Third Man* Page 3 don't fit my tastes ..
Fatigue	0.0/30 → 2.0/30	0.0/30 → 2.0/30	9.5/30 → 19.5/30	25.4/30 → 30.0/30
Explanation	Like poster and genres	Drama not my taste	Matches past high ratings	Old and fatigue, not interested

Table 7: Comparison of user simulators

Simulator	Simulation	Multi-modal	Evaluation
RecoGym [32]	Direct-Response	✗	Case Study
RecSim [16]	Direct-Response	✗	Case Study
VirtualTaobao [34]	Simplified UI	✗	Online
Adversarial [2]	Simplified UI	✗	Offline
KuaiSim [50]	Simplified UI	✗	Offline
SUBER [5]	Simplified UI	✗	Offline & Case Study
Agent4Rec [46]	Simplified UI	✗	Offline & Case Study
LLM-SimuRec [49]	Direct-Response	✗	Offline & Case Study
A/B Agent	Sandbox UI	✔	Offline & Case Study

4.2 LLM User Simulator

Equipped with prior open-world knowledge, LLM-based user simulators can provide flexible interaction feedback and explainable thought processes [42]. Several works apply LLM-based user simulators in recommender systems. ToolRec [55] designs a user simulator to evaluate user preference and provide recommendations by tool learning. RecAgent [39] and S^3 [**?**] introduce a recommendation agent with social networks environment. Agent4Rec [46] develops a user simulator to interact with movie websites in a page-by-page manner. In addition, some works also utilize the LLM-based user simulator to evaluate the conversational recommender system [41, 44, 45, 57]. However, these works lack the ability to use image-modal information and interact effectively with a realistic recommendation environment, limiting the agent's capability to simulate real-world user behavior.

5 Conclusion

In this paper, we propose a novel multimodal LLM-based user agent framework A/B Agent for A/B testing. Recognizing the limitations of current agents in replicating human perception and interaction within realistic environments, we first constructed a comprehensive multimodal and multi-page recommendation sandbox that mirrors online platform UIs. Within this environment, A/B Agent leverages multimodal information perception, fine-grained user preferences, and integrates profiles, memory, and a fatigue system to simulate complex human decision-making. Through extensive recommendation system evaluation and data augmentation experiments, we demonstrate the strong potential of A/B Agent to simulate user behavior in A/B testing sandbox environment.

Acknowledgements

This research was partially supported by National Natural Science Foundation of China (No.62502404), Hong Kong Research Grants Council (Research Impact Fund No.R1015-23, Collaborative Research Fund No.C1043-24GF, General Research Fund No.11218325), Institute of Digital Medicine of City University of Hong Kong (No.9229503), and Huawei (Huawei Innovation Research Program).

Statement on Copyrights

The movie poster images displayed in this paper are utilized strictly for academic research and illustrative purposes to demonstrate the recommendation scenarios. We acknowledge that the copyright of these visual assets belongs to their respective production studios or distributors, and their inclusion here falls under the fair use principles for academic publication.

Discussion

Despite the potential of A/B Agent, we acknowledge limitations regarding simulation scope and model constraints. First, our framework operates within a closed environment, overlooking external real-world influences like social media and peer interactions that shape user decisions. Second, intrinsic LLM issues, such as hallucinations and repetitive behaviors, may compromise simulation reliability. Future work will focus on integrating broader information channels and mitigating these generative anomalies.

References

[1] Xueying Bai, Jian Guan, and Hongning Wang. 2019. A model-based reinforcement learning with adversarial training for online recommendation. *Advances in Neural Information Processing Systems* 32 (2019).

[2] 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 *International Conference on Machine Learning*. PMLR, 1052–1061.

[3] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. 2016. Wide & deep learning for recommender systems. In *Proceedings of the 1st workshop on deep learning for recommender systems*. 7–10.

[4] Weiyu Cheng, Yanyan Shen, and Linpeng Huang. 2020. Adaptive factorization network: Learning adaptive-order feature interactions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 3609–3616.

[5] Nathan Corecco, Giorgio Piatti, Luca A Lanzendörfer, Flint Xiaofeng Fan, and Roger Wattenhofer. 2024. Suber: An rl environment with simulated human behavior for recommender systems. *arXiv preprint arXiv:2406.01631* (2024).

[6] Kuicai Dong, Yujing Chang, Derrick Goh Xin Deik, Dexun Li, Ruiming Tang, and Yong Liu. 2025. MMDocIR: Benchmarking Multimodal Retrieval for Long Documents. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (Eds.). Association for Computational Linguistics, Suzhou, China, 30959–30993. doi:10.18653/v1/2025.emnlp-main.1576

[7] Kuicai Dong, Yujing Chang, Shijie Huang, Yasheng Wang, Ruiming Tang, and Yong Liu. 2025. Benchmarking Retrieval-Augmented Multimodal Generation for Document Question Answering. arXiv:2505.16470 [cs.IR] https://arxiv.org/abs/2505.16470

[8] Aleksander Fabijan, Pavel Dmitriev, Helena Holmstrom Olsson, and Jan Bosch. 2018. Online controlled experimentation at scale: an empirical survey on the current state of A/B testing. In *2018 44th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*. IEEE, 68–72.

[9] Zichuan Fu, Xiangyang Li, Chuhan Wu, Yichao Wang, Kuicai Dong, Xiangyu Zhao, Mengchen Zhao, Huifeng Guo, and Ruiming Tang. 2025. A unified framework for multi-domain ctr prediction via large language models. *ACM Transactions on Information Systems* 43, 5 (2025), 1–33.

[10] Jingtong Gao, Xiangyu Zhao, Muyang Li, Minghao Zhao, Runze Wu, Ruocheng Guo, Yiding Liu, and Dawei Yin. 2024. Smlp4rec: An efficient all-mlp architecture for sequential recommendations. *ACM Transactions on Information Systems* 42, 3 (2024), 1–23.

[11] Alexandre Gilotte, Clément Calauzènes, Thomas Nedelec, Alexandre Abraham, and Simon Dollé. 2018. Offline a/b testing for recommender systems. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*. 198–206.

[12] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: a factorization-machine based neural network for CTR prediction. *arXiv preprint arXiv:1703.04247* (2017).

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

[14] Xiangnan He and Tat-Seng Chua. 2017. Neural factorization machines for sparse predictive analytics. In *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*. 355–364.

[15] Xu Huang, Weiwen Liu, Xiaolong Chen, Xingmei Wang, Hao Wang, Defu Lian, Yasheng Wang, Ruiming Tang, and Enhong Chen. 2024. Understanding the planning of LLM agents: A survey. *arXiv preprint arXiv:2402.02716* (2024).

[16] 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. *arXiv preprint arXiv:1909.04847* (2019).

[17] Raisa Islam and Owana Marzia Moushi. 2024. GPT-4o: The Cutting-Edge Advancement in Multimodal LLM. *Authorea Preprints* (2024).

[18] Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems (TOIS)* 20, 4 (2002), 422–446.

[19] Ron Kohavi and Roger Longbotham. 2015. Online controlled experiments and A/B tests. *Encyclopedia of machine learning and data mining* (2015), 1–11.

[20] Chengxi Li, Yejing Wang, Qidong Liu, Xiangyu Zhao, Wanyu Wang, Yiqi Wang, Lixin Zou, Wenqi Fan, and Qing Li. 2023. STRec: Sparse transformer for sequential recommendations. In *Proceedings of the 17th ACM conference on recommender systems*. 101–111.

[21] Lihong Li, Wei Chu, John Langford, Taesup Moon, and Xuanhui Wang. 2012. An unbiased offline evaluation of contextual bandit algorithms with generalized linear models. In *Proceedings of the Workshop on On-line Trading of Exploration and Exploitation 2*. JMLR Workshop and Conference Proceedings, 19–36.

[22] Xiangyang Li, Bo Chen, HuiFeng Guo, Jingjie Li, Chenxu Zhu, Xiang Long, Sujian Li, Yichao Wang, Wei Guo, Longxia Mao, et al. 2022. Inttower: the next generation of two-tower model for pre-ranking system. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 3292–3301.

[23] Yuanchun Li, Hao Wen, Weijun Wang, Xiangyu Li, Yizhen Yuan, Guohong Liu, Jiacheng Liu, Wenxing Xu, Xiang Wang, Yi Sun, et al. 2024. Personal llm agents: Insights and survey about the capability, efficiency and security. *arXiv preprint arXiv:2401.05459* (2024).

[24] Jiaju Lin, Haoran Zhao, Aochi Zhang, Yiting Wu, Huqiuyue Ping, and Qin Chen. 2023. Agentsims: An open-source sandbox for large language model evaluation. *arXiv preprint arXiv:2308.04026* (2023).

[25] Langming Liu, Liu Cai, Chi Zhang, Xiangyu Zhao, Jingtong Gao, Wanyu Wang, Yifu Lv, Wenqi Fan, Yiqi Wang, Ming He, et al. 2023. Linrec: Linear attention mechanism for long-term sequential recommender systems. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 289–299.

[26] Shuchang Liu, Qingpeng Cai, Bowen Sun, Yuhao Wang, Ji Jiang, Dong Zheng, Peng Jiang, Kun Gai, Xiangyu Zhao, and Yongfeng Zhang. 2023. Exploration and regularization of the latent action space in recommendation. In *Proceedings of the ACM Web Conference 2023*. 833–844.

[27] Preetam Nandy, Divya Venugopalan, Chun Lo, and Shaunak Chatterjee. 2021. A/b testing for recommender systems in a two-sided marketplace. *Advances in Neural Information Processing Systems* 34 (2021), 6466–6477.

[28] OpenAI. 2024. text-embedding-3-small. https://platform.openai.com/docs/guides/embeddings. Accessed: 2025-07-29.

[29] Yanru Qu, Han Cai, Kan Ren, Weinan Zhang, Yong Yu, Ying Wen, and Jun Wang. 2016. Product-based neural networks for user response prediction. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*. IEEE, 1149–1154.

[30] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*. PMLR, 8748–8763.

[31] Steffen Rendle. 2010. Factorization machines. In *2010 IEEE International conference on data mining*. IEEE, 995–1000.

[32] David Rohde, Stephen Bonner, Travis Dunlop, Flavian Vasile, and Alexandros Karatzoglou. 2018. RecoGym: A Reinforcement Learning Environment for the problem of Product Recommendation in Online Advertising. *arXiv preprint arXiv:1808.00720* (2018).

[33] Weichen Shen. 2017. DeepCTR: Easy-to-use,Modular and Extendible package of deep-learning based CTR models. https://github.com/shenweichen/deepctr.

[34] Jing-Cheng Shi, Yang Yu, Qing Da, Shi-Yong Chen, and An-Xiang Zeng. 2019. Virtual-taobao: Virtualizing real-world online retail environment for reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 4902–4909.

[35] Weiping Song, Chence Shi, Zhiping Xiao, Zhijian Duan, Yewen Xu, Ming Zhang, and Jian Tang. 2019. Autoint: Automatic feature interaction learning via self-attentive neural networks. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 1161–1170.

[36] Harald Steck. 2011. Item popularity and recommendation accuracy. In *Proceedings of the fifth ACM conference on Recommender systems*. 125–132.

[37] Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. 2023. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291* (2023).

[38] Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, et al. 2024. A survey on large language model based autonomous agents. *Frontiers of Computer Science* 18, 6 (2024), 186345.

[39] Lei Wang, Jingsen Zhang, Hao Yang, Zhiyuan Chen, Jiakai Tang, Zeyu Zhang, Xu Chen, Yankai Lin, Ruihua Song, Wayne Xin Zhao, et al. 2023. User behavior simulation with large language model based agents. *arXiv preprint arXiv:2306.02552* (2023).

[40] Ruoxi Wang, Bin Fu, Gang Fu, and Mingliang Wang. 2017. Deep & cross network for ad click predictions. In *Proceedings of the ADKDD'17*. 1–7.

[41] Xiaolei Wang, Xinyu Tang, Wayne Xin Zhao, Jingyuan Wang, and Ji-Rong Wen. 2023. Rethinking the evaluation for conversational recommendation in the era of large language models. *arXiv preprint arXiv:2305.13112* (2023).

[42] Zhiheng Xi, Wenxiang Chen, Xin Guo, Wei He, Yiwen Ding, Boyang Hong, Ming Zhang, Junzhe Wang, Senjie Jin, Enyu Zhou, et al. 2023. The rise and potential of large language model based agents: A survey. *arXiv preprint arXiv:2309.07864* (2023).

[43] Ya Xu, Nanyu Chen, Addrian Fernandez, Omar Sinno, and Anmol Bhasin. 2015. From infrastructure to culture: A/B testing challenges in large scale social networks. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*. 2227–2236.

[44] Dayu Yang, Fumian Chen, and Hui Fang. 2024. Behavior Alignment: A New Perspective of Evaluating LLM-based Conversational Recommendation Systems. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2024)*. ACM. doi:10.1145/3626772.3657924

[45] Se-eun Yoon, Zhankui He, Jessica Maria Echterhoff, and Julian McAuley. 2024. Evaluating Large Language Models as Generative User Simulators for Conversational Recommendation. *arXiv preprint arXiv:2403.09738* (2024).

[46] 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*. 1807–1817.

[47] Chao Zhang, Haoxin Zhang, Shiwei Wu, Di Wu, Tong Xu, Xiangyu Zhao, Yan Gao, Yao Hu, and Enhong Chen. 2025. Notellm-2: Multimodal large representation models for recommendation. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*. 2815–2826.

[48] Yingyi Zhang, Pengyue Jia, Derong Xu, Yi Wen, Xianneng Li, Yichao Wang, Wenlin Zhang, Xiaopeng Li, Weinan Gan, Huifeng Guo, et al. 2025. Personalize Before Retrieve: LLM-based Personalized Query Expansion for User-Centric Retrieval. *arXiv preprint arXiv:2510.08935* (2025).

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

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

[51] Xiangyu Zhao, Long Xia, Liang Zhang, Zhuoye Ding, Dawei Yin, and Jiliang Tang. 2018. Deep Reinforcement Learning for Page-wise Recommendations. In *Proceedings of the 12th ACM Recommender Systems Conference*. ACM, 95–103.

[52] Xiangyu Zhao, Long Xia, Lixin Zou, Hui Liu, Dawei Yin, and Jiliang Tang. 2021. Usersim: User simulation via supervised generativeadversarial network. In *Proceedings of the Web Conference 2021*. 3582–3589.

[53] Xiangyu Zhao, Liang Zhang, Zhuoye Ding, Long Xia, Jiliang Tang, and Dawei Yin. 2018. Recommendations with Negative Feedback via Pairwise Deep Reinforcement Learning. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 1040–1048.

[54] Xiangyu Zhao, Xudong Zheng, Xiwang Yang, Xiaobing Liu, and Jiliang Tang. 2020. Jointly learning to recommend and advertise. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 3319–3327.

[55] Yuyue Zhao, Jiancan Wu, Xiang Wang, Wei Tang, Dingxian Wang, and Maarten de Rijke. 2024. Let Me Do It For You: Towards LLM Empowered Recommendation via Tool Learning. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1796–1806.

[56] Jieming Zhu, Jinyang Liu, Shuai Yang, Qi Zhang, and Xiuqiang He. 2021. Open benchmarking for click-through rate prediction. In *Proceedings of the 30th ACM international conference on information & knowledge management*. 2759–2769.

[57] Lixi Zhu, Xiaowen Huang, and Jitao Sang. 2024. How Reliable is Your Simulator? Analysis on the Limitations of Current LLM-based User Simulators for Conversational Recommendation. In *Companion Proceedings of the ACM on Web Conference 2024*. 1726–1732.

[58] Xizhou Zhu, Yuntao Chen, Hao Tian, Chenxin Tao, Weijie Su, Chenyu Yang, Gao Huang, Bin Li, Lewei Lu, Xiaogang Wang, et al. 2023. Ghost in the minecraft: Generally capable agents for open-world environments via large language models with text-based knowledge and memory. *arXiv preprint arXiv:2305.17144* (2023).

A More Implementation Details of A/B testing

In this section, we firstly detailed the **fatigue setting** on A/B Agent in Table 8.

Table 8: Fatigue values for different actions in GPT-4o and GPT-4o-mini

Action	Fatigue Value	
	GPT-4o	GPT-4o-mini
click_movie	15	2
watch_and_rate_movie	40	10
previous_page	2	2
next_page	2	2
back_action	2	5
exit_action	0	0

Then, we display our prompt design in our agent framework and give a typical example of user preference construction.

A.1 User Preference Generation

We provide below the prompt used for user preference generation (Figure 7).

User Preference Generation Prompt

System Instruction: You are given a user's movie interaction history. Assume the role of user. Your task is to write a clean, concise, and well-structured user preferences (taste) summary in the first person.
Guidelines:
(1) **Genres:** Identify and list the genres you prefer.
(2) **Directors:** Mention directors whose works you consistently enjoy.
(3) **Actors:** Highlight actors whose performances you appreciate.
(4) **Release Date Patterns:** Note any trends in the release years of the movies you watch.
(5) **Rating Tendencies:** Describe my typical ratings for different types of movies or score ranges.
(6) **Poster Style Preference:** Based on the movie interaction history, summarize the user's preference for movie poster aesthetics, such as color schemes, compositions, and character depictions.
(7) **Conciseness:** Summarize the preferences in a manner that is clear and to the point.
If no clear preferences are found, indicate this with 'not found'. Do not give any information that is not related to user preference.

Input (User History): `{history}`
Response Requirement: Remember, don't blindly repeat the contexts verbatim.
LLM Output: `{Summarized Preference}`

Figure 7: The prompt template for user preference generation.

The resulting user preferences are shown in Figure 8. The generated preferences capture fine-grained user tendencies specific to individual interaction interfaces.

User Preference Case Study

My Movie Preferences Summary
Genres: I have a strong preference for Drama and Romance, often enjoying films that blend these genres with elements of History, Mystery, and Crime. I also appreciate well-crafted Comedies and Thrillers.
Directors: I consistently enjoy the works of Krzysztof Kieślowski, Steven Spielberg, and James Ivory. Their storytelling and direction resonate with me deeply.
Actors: I appreciate the performances of Audrey Hepburn, Daniel Day-Lewis, and Helena Bonham Carter. Their ability to bring characters to life adds a significant value to the films they are in.
Release Date Patterns: I tend to favor films from the 1990s, with a noticeable appreciation for classics from the 1950s to the 1970s as well.
Rating Tendencies: I generally rate Drama and Romance films highly, often giving them 4 or 5 stars. Comedies and Thrillers also receive favorable ratings, typically around 4 stars. However, I am more critical of films that fail to engage me.
Poster Style Preference: I prefer movie posters with classic and elegant designs, often featuring well-composed images of the main characters.
Summary: In summary, my taste in movies leans heavily towards well-crafted dramas and romances, with a particular appreciation for strong performances and thoughtful direction.

Figure 8: An example of a generated user preference summary.

B Recommendation System Implementation

We implement all models using the DeepCTR [33] framework with binary cross-entropy loss, Adam optimizer ($lr = 1 \times 10^{-3}$), batch size 1024, and early stopping based on validation AUC. Unless specified otherwise, models use an embedding dimension of 64, DNN layers of (256, 128), dropout of 0.3, and L2 regularization of 0.003. Specific adjustments are as follows: FM uses no DNN with linear L2 regularization 10^{-5}. DeepFM adds a third layer (64) with dropout 0.5. Wide & Deep uses layers (256, 64) with dropout 0.5. xDeepFM and DCN (2 cross layers) use dropout 0.5 and linear L2 10^{-5}. AFN employs a log layer (256) without dropout. AutoInt uses two self-attention layers (2 heads). PNN and NFM follow the default configuration. All models are trained on an NVIDIA GeForce RTX 4060 and converge within 20 epochs.