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

Anonymous ACL submission

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 requirement. 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 testing from three perspectives: model, data, and features. Additionally, we found that the data generated by A/B Agent can effectively enhance the capabilities of recommendation models. Our code is public abailable [1].

1 Introduction

In real-world industrial settings, recommender systems often use online A/B testing to optimize and evaluate model performance in real time (Kohavi and Longbotham, 2015; Xu et al., 2015; Nandy et al., 2021; Fabijan et al., 2018). In A/B testing pipeline, users are randomly divided into experimental and control groups, with the experimental group receiving recommendations from a new system and the control group using the existing system as a baseline. However, A/B testing has several challenges: 1) High cost: It requires significant server resources and data analysis efforts, especially for high-traffic products (Gilotte et al., 2018). 2) Degraded user experience: Frequent A/B testing can disrupt user interaction and lower satisfaction (Li et al., 2012). 3) Time-consuming: It takes time to collect sufficient data for statistical analysis, delaying quick evaluation. Therefore, it is essential to develop reliable offline evaluation methods to simulate A/B testing results.

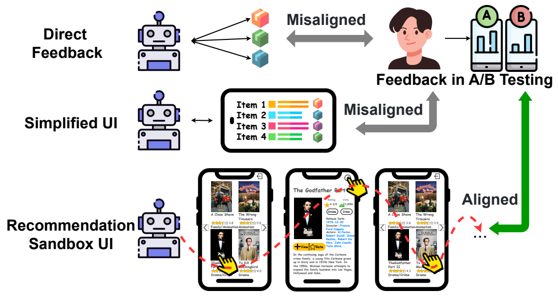


Figure 1: Comparison of alignment between simulated user feedback and real user feedback in A/B testing.

The complex interactions between users and the environment in A/B testing present challenges for user simulator design. Specifically, users perceive rich multimodal information from different interfaces, actively explore content of interest, disengage when fatigued, and generate feedback data across interfaces to evaluate models.

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 (Wang et al., 2024; Lin et al., 2023; Zhu et al., 2023; Wang et al., 2023a). For instance, iEvaLM (Wang et al., 2023c) employs an LLM-based user agent to evaluate conversational recommender systems, providing flexible natural language interactions. RecAgent (Wang

[1] https://anonymous.4open.science/r/MMAgent-D8E2/

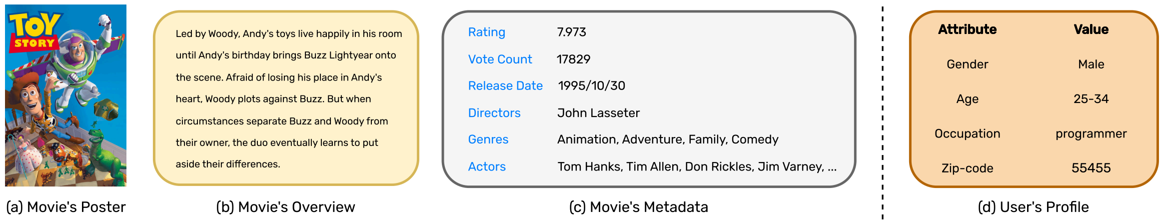


Figure 2: Data example from Multimodal-Movielens-1M dataset.

et al., 2023b) designs a Web recommendation scenario where user agents can interact with each other, exploring the impact of human social behavior on recommendation results. Agent4Rec (Zhang et al., 2024) 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 the actual human perception process. Figure1 illustrates the simulation methods used in existing approaches, which either directly simulate user-item interaction feedback or model user behavior in simplified, text-based UI environments. However, these approaches fail to accurately reflect real user behavior on recommendation platforms. Moreover, the agents lack multimodal perception and multi-layered interface simulation, which limits their ability to replicate human interaction pathways. To address this gap, we propose A/B Agent to simulate the 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) Agent design in the sandbox environment.** In the sandbox environment, users interact with movie information at varying levels of granularity across different interfaces, which results in long and complex behavioral chains. However, existing user agent designs struggle to achieve fine-grained perception and human-like exploration. To address these challenges, we have 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 are as follows:

- We propose A/B Agent to simulate the entire human perception process and interactive behavior chains for A/B testing. The agent possesses multimodal information perception and can perform human-like exploration within the sandbox recommendation environment.
- We developed 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 assess the agent's simulation capabilities in sandbox environments. Both the A/B testing for recommendation models and data augmentation experiments using agent feedback data confirm the effectiveness of A/B Agent simulation.

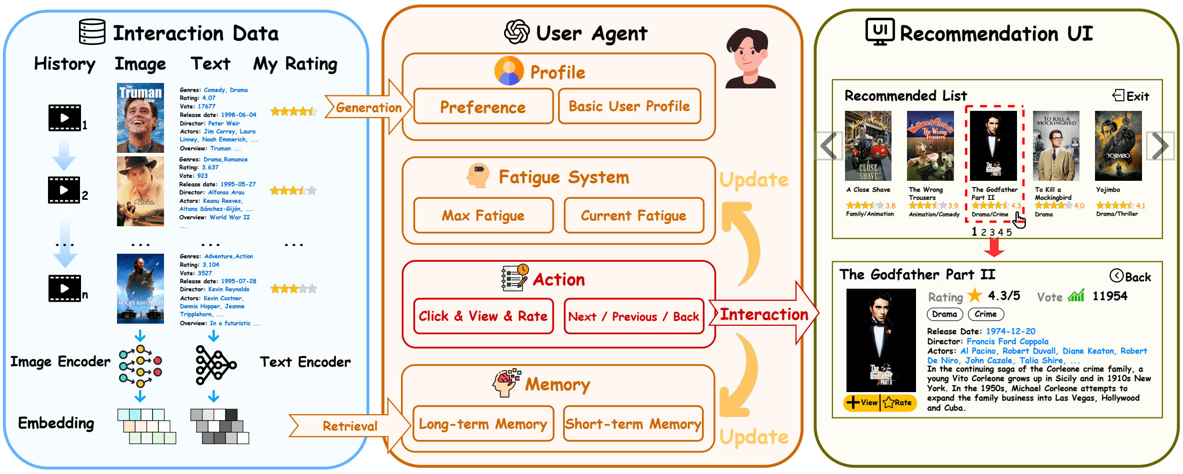


Figure 3: The Framework of A/B Agent. Our agent design involves three components: multimodal User Agent (Orange Section), Recommendation UI (Green Section), and Interaction Data (Blue Section). The Recommendation UI provides a multimodal, multi-interface sandbox for the agent. Based on Interaction Data, 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.

2 A/B Agent Framework

2.1 Framework Overview

As shown in Figure 3, 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 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 (Harper and Konstan, 2015) is devoid of multimodal data, and many datasets like Criteo (Zhu et al., 2021) and Avazu (Zhu et al., 2021) have anonymized feature characteristics. To address these limitations, we introduce the Multimodal-MovieLens-1M (MM-ML-1M) dataset[2], an extension of the original MovieLens-1M (Harper and Konstan, 2015), enhanced with additional movie posters and metadata. Figure 2 provides a detailed illustration of the data structure, with comprehensive dataset statistics available in Appendix C. 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 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 4.

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

[2] https://anonymous.4open.science/r/MM-ML-1M-9805/

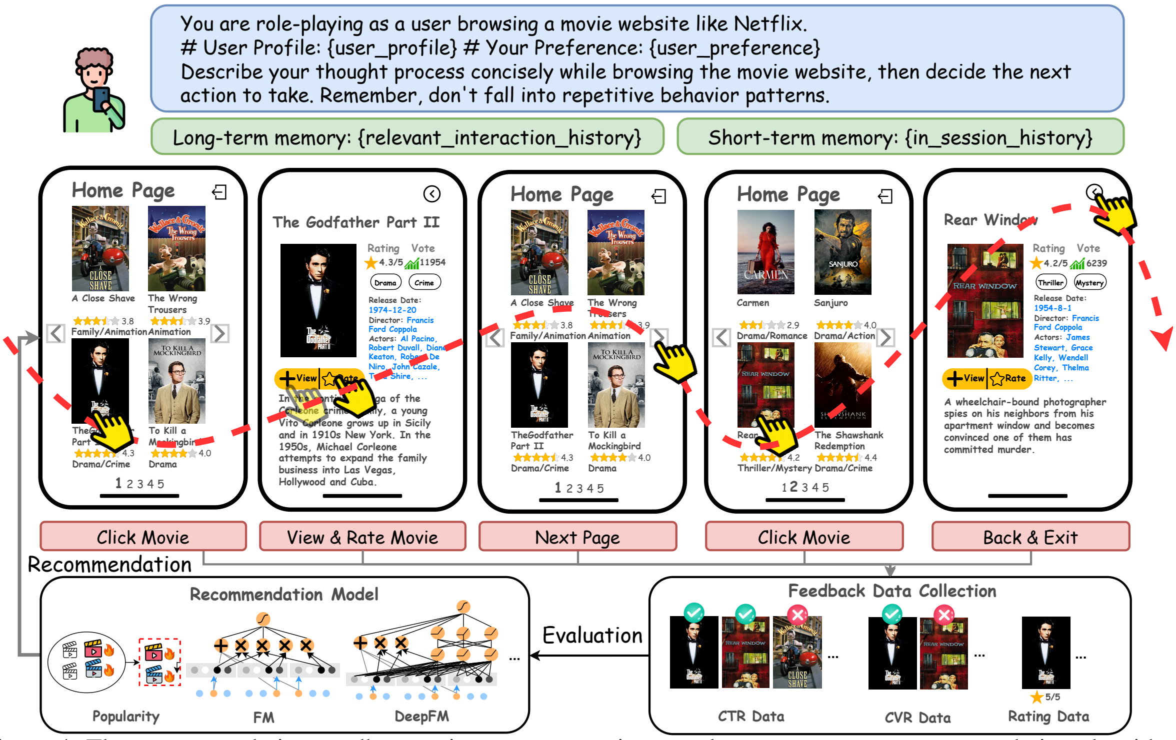


Figure 4: The recommendation sandbox environment comprises two key components: recommendation algorithms and a user interface. The recommendation algorithms generate recommendation lists displayed to users on the home page and individual movie detail pages. User interaction with this interface, including click-through rate (CTR), conversion rate (CVR), and ratings, provides data for recommendation model evaluation.

interactive options.

Home Page. As depicted in Figure 4, 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 4. This page offers comprehensive information about the chosen movie, including 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 detailed 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 4, offering scalability for developing and optimizing recommender systems. It includes collaborative filtering algorithms such as random recommendation, popularity-based models (Steck, 2011), Factorization Machines (FM) (Rendle, 2010), and DeepFM (Guo et al., 2017). 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.

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 3.

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 (Li et al., 2024). 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. For more details, see Appendix A.1.

2.4.2 Memory Module

The memory module is essential for retaining interaction history and supporting decision-making in recommendation systems (Huang et al., 2024). While existing memory designs provide a basic framework, they neglect visual modality retrieval (Zhang et al., 2024). To achieve the multi-modal 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 The long-term memory component is designed to store item interaction histories and encode detailed information to enhance decision-making. This module seamlessly integrates both textual and visual data for effective memory retrieval. 1). *Textual Retrieval*: The system employs the OpenAI `text-embedding-3-small` model to encode comprehensive movie meta-information into embeddings $\boldsymbol{e}_{\text{text}}$. When encountering a new interface, the agent analyzes the page's textual content to generate queries $\boldsymbol{q}_{\text{text}}$. These queries are used to retrieve relevant interaction records, thereby providing historical context to guide decision-making. 2). *Visual Retrieval*: Visual information is also leveraged by encoding movie posters into embeddings $\boldsymbol{e}_{\text{image}}$ using the CLIP model (Radford et al., 2021). The agent formulates queries $\boldsymbol{q}_{\text{image}}$ based on visual elements such as color schemes and character depictions. It then retrieves pertinent memories through cosine similarity.

By integrating both textual and visual retrieval processes, the long-term memory module effectively captures detailed information across modalities, establishing a human-like, multimodal memory mechanism.

Short-Term Memory Short-term memory records all in-session interactions between the agent and the recommendation environment within the current session, recording recent experiences and observations. This memory enables the agent to maintain continuity in its actions, avoiding repetitive behaviors and supporting a coherent exploration of the current session. To store in session records, new experiences are recorded in a structured format, including the current location, key observations (whether movies catch the agent's attention), a numerical interest level (1-5), and the actions taken. This process ensures that the relevant context of each interaction is preserved, enabling the agent to refer back to specific details as needed.

2.4.3 Action Module

The action module defines the agent's workflow and permissible actions within the recommendation environment (Zhang et al., 2024), 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*. Further details are provided in Appendix A.3.

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

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

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

3 Experiment

3.1 Experimental Setting

Rating records are split into training, validation, and test sets in a 7:2:1 ratio based on the timestamp. The training sets are used to initialize the profile module within A/B Agent and train the recommendation model. The performance of A/B Agent is evaluated within the recommendation sandbox using Click-Through Rate (CTR), Conversion Rate (CVR), and Average Rating (AR). For validation of the agent simulation, real-world model performance is evaluated using recall and Normalized Discounted Cumulative Gain (NDCG) (Järvelin and Kekäläinen, 2002) on the test set.

For implementation details, user preferences are summarized using GPT-4o (Islam and Moushi, 2024), and for agent simulation, we use GPT-4o and GPT-4o mini as the backbone LLMs. The memory module employs the text-embedding-small model, while image processing is handled by CLIP (Radford et al., 2021) with the VIT-L-14 architecture. Each recommender generates a list of 20 movies, with 5 displayed on the home page.

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 1 presents the results of our A/B testing from the model perspective. The results demonstrate the ability of A/B Agent to evaluate recommendation models. We observe that (1) A/B Agent validates a clear and consistent performance ranking among the models across different metrics, with performance progressively increasing from random, pop, and FM to DeepFM. (2) A/B Agent effectively evaluates the model performance based on different backbones, demonstrating its versatility and applicability to multimodal large models with varying parameters. (3) The simulation results of A/B Agent are consistent with real user feedback.

3.2.2 Data Scale Impact

Table 2 shows the impact of the training data scale on the performance of the DeepFM model. These results illustrate that A/B Agent can effectively capture the performance improvements resulting from increased training data. We observe a clear positive correlation between the amount of training data and all three metrics. A/B Agent validated the performance gains achieved by increasing data scale in recommendation systems.

3.2.3 Feature Importance Evaluation

Table 3 presents the results of our feature importance analysis for the DeepFM model. These results demonstrate that A/B Agent can discern the impact of different feature sets on recommendation performance. We observe that the model using all features yields the best performance across all metrics, while the model using only User ID performs better in terms of CTR and CVR compared to the model using only Movie ID. This suggests that A/B Agent can evaluate the relative importance of the features of the user and the item in the recommendation simulation.

Our experiments demonstrate the ability of A/B Agent to effectively evaluate recommender system performance across various aspects, including model differentiation, data scale impact, and feature set importance. This makes A/B Agent a valuable tool for evaluating and comparing recommender systems within a realistic, multimodal interactive environment.

3.3 Agent Alignment

3.3.1 User Taste Alignment

The consistency between simulated agent behavior and real user behavior is evaluated by comparing user agent satisfaction with recommendations that demonstrate varying satisfaction rates derived from real user feedback. Specifically, we regard

Evaluation	A/B Agent (GPT-4o)			A/B Agent (GPT-4o-mini)			Real-World	
Model	CTR	CVR	AR	CTR	CVR	AR	Recall	NDCG
random	0.2330	0.1147	4.27	0.0872	0.0461	4.01	0.0066	0.0222
pop	0.3077	0.1835	4.30	0.1886	0.1181	4.2	0.0216	0.0881
FM	0.3635	0.2940	4.70	0.2642	0.1984	4.52	0.0353	0.0987
DeepFM	**0.4453**	**0.3458**	**4.75**	**0.2891**	**0.2094**	**4.52**	**0.0429**	**0.1130**

Table 1: Performance of recommendation model evaluation 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 2: Performance of DeepFM with various training data scale within the A/B Agent Framework.

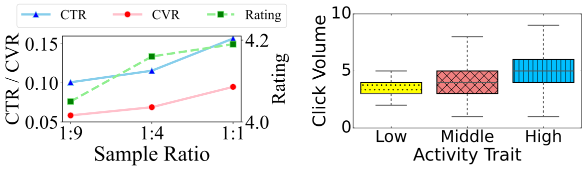
Evaluation	A/B Agent (GPT-4o-mini)			Real-World	
Feature	CTR	CVR	AR	Recall	NDCG
User ID Only	0.2754	0.1982	4.47	0.0359	0.0981
Movie ID Only	0.2850	0.2097	4.51	0.0372	0.0966
All	**0.2891**	**0.2094**	**4.52**	**0.0429**	**0.1130**

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

the movies that users actually clicked in the test set as positive samples, and the movies that users have not interacted with as negative samples, and recommend 20 movies for each user. With different sampling ratio settings, we obtain recommendation results with positive and negative sample ratios of 1:1, 1:4, and 1:9, respectively. Figure 5(a) demonstrates the simulation results. We observe that the agent shows a higher degree of acceptance for the recommendation results with a higher positive sample ratio, whether in terms of CTR, CVR, or AR, indicating that the agent-simulated results are consistent with the preferences of the real users.

3.3.2 Activity Trait Alignment

To evaluate the impact of activity traits, we analyzed the movie click distribution of A/B Agent under different activity trait settings, and the results are depicted in Figure 5(b). When the activity attribute is set to high, medium, and low, the click distribution of the Agent shows a change from high to low in terms of peak value and total click volume, which indicates that our fatigue design can effectively control the activity level of the agent.



(a) User taste alignment experiments under various sample ratios.

(b) Click distribution among user groups with different activity traits.

Figure 5: Agent alignment experiments.

3.4 Data Augmentation

To evaluate the effectiveness of the A/B Agent simulation, we collected the agent's simulated behavior under DeepFM recommendations. Specifically, we gathered 2,518 click behaviors on the home page and 1,884 watching records from the movie detail page, which are significant less than the original training set of 700,146 samples. These data are then concatenated with the original training set to form a new training dataset for offline recommendation system evaluation. Table 4 reports the data augmentation results of various models including NFM (He and Chua, 2017), xDeepFM (Guo et al., 2017), Wide&Deep (Cheng et al., 2016), DCN (Wang et al., 2017), DeepFM (Guo et al., 2017). Recommendation results show significant improvements in both click data and viewing records across the five recommendation models. For click data, all models achieve AUC improvements exceeding 0.002, with Wide&Deep showing the largest gain at 0.0032, highlighting the quality of the simulated data. The improvements are even more significant for CVR data. NFM, xDeepFM, Wide&Deep, and DeepFM see AUC increases of 0.0037, 0.0039, 0.0052, and 0.0022, respectively. DCN achieves a 0.002 improvement, similar to that from click data. In conclusion, the agent simulation produces high-quality data that leads to significant performance improvements, even with a small dataset of around 2,000 points, much smaller than the original 700,000 samples.

Model	NFM	xDeepFM	Wide&Deep	DCN	DeepFM
Original	74.38	74.78	74.69	75.17	75.20
+Simulated Click Data(w/o vision)	74.58	74.82	74.85	75.33	75.01
+Simulated View Data(w/o vision)	74.55	74.86	74.82	75.34	75.28
+ Simulated Click Data(w/ vision)	<u>74.64</u>	<u>75.02</u>	<u>75.01</u>	**75.39**	<u>75.41</u>
+ Simulated View Data (w/ vision)	**74.75**	**75.17**	**75.21**	<u>75.37</u>	**75.42**

Table 4: 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 (Li et al., 2022)

3.5 Ablation Study

To verify the importance of the image modality in the agent simulation process, we conducted an ablation experiment on the image modality. Specifically, we display the DeepFM recommendation results while no movie poster is displayed in the UI interface. The data augmentation results of the interaction data simulated under this setting are recorded in the Table 4 with (w/o vision) for both click and view data. We can observe that under the condition of eliminating visual information, the simulated results can still provide some performance improvement. However, the enhancement is significantly lower compared to the results obtained by agent simulation using visual information.

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 (Ie et al., 2019; Rohde et al., 2018; Shi et al., 2019; Chen et al., 2019; Bai et al., 2019). RecSim (Ie et al., 2019) 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 (Zhao et al., 2021) 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.

4.2 LLM User Simulator

Equipped with prior open-world knowledge, LLM-based user simulators can provide flexible interaction feedback and explainable thought processes (Xi et al., 2023). Several works apply LLM-based user simulators in recommender systems. ToolRec (Zhao et al., 2024) designs a user simulator to evaluate user preference and provide recommendations by tool learning. RecAgent (Wang et al., 2023b) and S^3 (Gao et al., 2023) introduce a recommendation agent with social networks environment. Agent4Rec (Zhang et al., 2024) 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 (Wang et al., 2023c; Yang et al., 2024; Yoon et al., 2024; Zhu et al., 2024). 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 multimodal LLM-based user agent framework (A/B Agent) for A/B testing. To tackle the inconsistency between the online recommendation platform and the current simulation environment, we construct a multimodal multi-interface recommendation sandbox. To simulate human perception and interaction patterns in online A/B testing, we design a multimodal agent that uses multimodal information perception, fine-grained user preference, and integrates profiles, action memory retrieval, and a fatigue system to simulate complex decision-making. Both recommendation system evaluation and data augmentation experiments demonstrate that the agent simulation align with online A/B testing results.

Limitations

Although A/B Agent presented in this paper shows potential in A/B testing for recommender systems, we acknowledge several limitations that may affect its broader application in real-world settings. These limitations are primarily related to two aspects:

User Interaction Beyond the Recommender Environment. The current agent-based model focuses mainly on user behavior within the recommender system environment but does not fully account for the multiple information sources that influence user decisions in the real world. In real-world A/B testing scenarios, the user action are not only influenced by the recommender system itself but also by social media, feedback from others, and other external information channels. Users may acquire information through interactions with peers, browsing external websites, or using social platforms, which could affect their acceptance of recommended content or their choice behavior. Since these external interactions and information sources are not effectively captured in the current simulation framework, there may be discrepancies between the simulated user behavior and real-world behavior.

Hallucination Issues with LLMs. LLMs face the hallucination issues where the model generates irrelevant or inconsistent data. LLMs may produce inconsistent reasoning, offering contradictory conclusions based on similar inputs, thus undermining the reliability of the simulation. LLMs may also exhibit repetitive behavior, failing to adapt to new contexts or changing environments, which may not accurately reflect real-world decision-making processes. We will explore methods to mitigate hallucination issues during the simulation process in the further direction.

References

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.

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*, pages 1052–1061. PMLR.

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*, pages 7–10.

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)*, pages 68–72. IEEE.

Chen Gao, Xiaochong Lan, Zhihong Lu, Jinzhu Mao, Jinghua Piao, Huandong Wang, Depeng Jin, and Yong Li. 2023. S^3: Social-network simulation system with large language model-empowered agents. *arXiv preprint arXiv:2307.14984*.

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*, pages 198–206.

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*.

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 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*, pages 355–364.

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*.

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*.

Raisa Islam and Owana Marzia Moushi. 2024. Gpt-4o: The cutting-edge advancement in multimodal llm. *Authorea Preprints*.

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

Ron Kohavi and Roger Longbotham. 2015. Online controlled experiments and a/b tests. *Encyclopedia of machine learning and data mining*, pages 1–11.

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*, pages 19–36. JMLR Workshop and Conference Proceedings.

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*, pages 3292–3301.

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*.

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*.

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*, 34:6466–6477.

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*, pages 8748–8763. PMLR.

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

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*.

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*, volume 33, pages 4902–4909.

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

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. *arXiv preprint arXiv:2305.16291*.

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):186345.

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

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

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

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*.

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*, pages 2227–2236.

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.

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*.

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.

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*, pages 3582–3589.

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, pages 1796–1806.

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*, pages 2759–2769.

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*, pages 1726–1732.

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*.

Table 5: 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

A Implementation Details

In this section, we firstly detailed the fatigue setting on MM-ML-1M and the prompt case in Table 5. Then, we display our prompt design in our agent framework and give a typical example of user preference construction.

A.1 User Preference Generation

User Preference Generation

Prompt: 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.
Here is your movie interaction history:
{history}
Remember, don't blindly repeat the contexts verbatim.
LLM: {Summarized Preference}.

User Preference Case

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. This suggests a preference for both modern classics and timeless older films.
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, as seen with lower ratings for certain Comedies and Dramas.
Poster Style Preference: I prefer movie posters with classic and elegant designs, often featuring well-composed images of the main characters. Posters with a balanced color scheme and a focus on character depiction appeal to me the most.
In summary, my taste in movies leans heavily towards well-crafted dramas and romances, with a particular appreciation for strong performances and thoughtful direction. I enjoy films that offer a blend of historical context, mystery, and emotional depth."

A.2 Agent System Prompt

Agent System Prompt

Agent System Prompt: You are role-playing as a user browsing a movie website like Netflix.
User Profile
{user_profile}
Your Preferences
{movie_taste}
Response
Describe your thought process concisely while browsing the movie website, then decide the next action to take.
Use the following JSON format to specify your chosen action:
"‘json "action": "name": "<action_name>", "parameters": "param1": "<value1>", ... ,
Remember, don’t fall into repetitive behavior patterns. Keep track of your recent actions and adjust your behavior accordingly.

A.3 Action Prompt

Available Actions on home page

Available Actions
['name': 'click_movie', 'fatigue': 2, 'description': 'View detailed information about a movie that might align with your preferences.', 'parameters': 'type': 'object', 'properties': 'movie_id': 'type': 'int', 'description': 'ID of the movie.', 'required': ['movie_id'], 'name': 'exit', 'fatigue': 0, 'description': "Exit the application when one of the following conditions is met:
1) Your fatigue level has reached its maximum.
2) You've viewed all pages without finding any movies that match your preferences.
3) You've found and watched all the movies that appeal to your tastes.", 'parameters': 'type': 'object', 'properties': 'reason': 'type': 'string', 'description': 'Reason for exiting the application.', 'required': ['reason'], 'name': 'next_page', 'fatigue': 2, 'description': "Navigate to the next page of movie recommendations. Use this action to discover more films that might interest you when you've finished reviewing the current page."]

Case Study for Agent Thought

Search Long Term Memory: args: {"genres": ["drama", "crime"], "directors": ["Frank Darabont"], "actors": ["Tim Robbins", "Morgan Freeman"], "keywords": ["prison", "hope", "friendship"], "style": ["realistic"]}

Short Term Memory: 1. Location: Movie List (Page 1) | Insights: The Shawshank Redemption and Raiders of the Lost Ark have high ratings and align well with my favorite genres. The posters for these movies are visually striking and appealing. | Action: Clicked on The Shawshank Redemption for detailed information.

Thought:
The poster for "The Shawshank Redemption" is visually striking, featuring a man standing in the rain with his arms outstretched, which conveys a sense of freedom and hope. The dark, moody color scheme and the dramatic lighting immediately catch my eye, hinting at the intense and emotional journey the film promises. Given my strong preference for Drama and Crime genres, and the high rating of 4.4, this movie aligns perfectly with my tastes. Additionally, the presence of Morgan Freeman, an actor whose performances I admire, further increases my interest. The storyline about hope and integrity in a prison setting is compelling and thought-provoking, which is exactly what I look for in a film. Considering all these factors, I am inclined to watch and rate this movie.

Figure 6: Case study for thought process

Available Actions on Movie Detail Page

Available Actions
['name': 'back', 'fatigue': 5, 'description': "Return to the previous home page. Use this action when the current movie details don't match your preferences.", 'name': 'watch_and_rate_movie', 'fatigue': 10, 'description': 'Select and watch a movie that strongly appeals to your tastes, then rate it based on your personal viewing experience and preferences. Avoid rewatching previously rated films.', 'parameters': 'type': 'object', 'properties': 'movie_id': 'type': 'int', 'description': 'ID of the movie.', 'rating': 'type': 'float', 'description': 'Rating from 1.0 to 5.0', 'required': ['movie_id', 'rating'], 'name': 'exit', 'fatigue': 0, 'description': "Exit the application when one of the following conditions is met:
1) Your fatigue level has reached its maximum.
2) You've viewed all pages without finding any movies that match your preferences.
3) You've found and watched all the movies that appeal to your tastes.", 'parameters': 'type': 'object', 'properties': 'reason': 'type': 'string', 'description': 'Reason for exiting the application.', 'required': ['reason']]

B Case Study

Figure 6 illustrates A/B Agent's thought process for multimodal information. The agent starts by searching its long-term memory, which includes

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	-

Table 6: Dataset statistics in MM-ML-1M. Note: the range for text type is the length of text content.

textual and visual elements. Unlike traditional simulators that rely solely on text metadata, A/B Agent integrates visual analysis into its decision-making. This multimodal approach allows the agent to evaluate movies by considering both visual and textual aspects throughout memory retrieval and reasoning.

C The statistics of MM-ML-1M

We list the statistics of our collected dataset in Table 6, which reports the type, count, and range for each feature.