EvalAgent: Towards Evaluating News Recommender Systems with LLM-based Agents

Guangping Zhang
Fudan University
Shanghai, China
gpzhang20@fudan.edu.cn

Peng Zhang*
Fudan University
Shanghai, China
zhangpeng_@fudan.edu.cn

Jiahao Liu
Fudan University
Shanghai, China
jiahaoliu21@m.fudan.edu.cn

Zhuoheng Li
Fudan University
Shanghai, China
zhuohengli22@m.fudan.edu.cn

Dongsheng Li
Microsoft Research Asia
Shanghai, China
dongshengli@fudan.edu.cn

Hansu Gu
Independent
Seattle, United States
hansug@acm.org

Tun Lu*
Fudan University
Shanghai, China
lutun@fudan.edu.cn

Ning Gu
Fudan University
Shanghai, China
ninggu@fudan.edu.cn

Abstract

Online news platforms have become the primary source of information consumption, with recommender systems serving as critical gateways that shape public discourse through their algorithmic power, necessitating rigorous evaluation methodologies. Traditional offline evaluation methods struggle with evolving user behavior and dynamic system adaptation, while online experiments are costly, time-consuming, and ethically challenging. To address these challenges, this paper introduces EvalAgent, a large language model agent system for simulating real-world online news recommender systems. EvalAgent employs Stable Memory (StM) to model users' exploration-exploitation dynamics, mitigating noise from irrelevant interactions by analyzing the distribution density of news articles within the short-term memory, and incrementally maintains the long-term memory to capture users' high-level preferences, thereby enabling a consistent and reliable simulation of sustained interactions. It further incorporates an Environment Interaction Framework (EIF) to enable seamless engagement with real-world recommender systems. This approach yields a precise, scalable, and ethically responsible evaluation framework for news recommender systems. Comprehensive experiments and user studies substantiate EvalAgent's efficacy, with publicly available code to support ongoing research in recommender system evaluation.

CCS Concepts

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

*Corresponding author.

CIKM '25, Seoul, Republic of Korea

ACM ISBN 979-8-4007-2040-6/2025/11
https://doi.org/10.1145/3746252.3761127

Keywords

Recommender System, Large Language Model Agent, Simulation-based Evaluation.

ACM Reference Format:
Guangping Zhang, Peng Zhang, Jiahao Liu, Zhuoheng Li, Dongsheng Li, Hansu Gu, Tun Lu, and Ning Gu. 2025. EvalAgent: Towards Evaluating News Recommender Systems with LLM-based Agents. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management (CIKM '25), November 10–14, 2025, Seoul, Republic of Korea.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3746252.3761127

1 Introduction

Online news platforms have become the primary source of information consumption, with over 86% of Americans regularly getting news from digital devices [1]. Recommender systems serve as the critical gateway that determines what information users encounter, giving them significant "algorithmic power" in shaping public discourse [7, 15]. **Effective evaluation of news recommender systems has become increasingly important for multiple stakeholders.** For platform operators, accurate evaluation directly translates to higher user engagement and commercial revenue. For individual users, evaluation enables informed platform selection based on their information needs. Furthermore, an independent third-party evaluation serves broader societal interests through detecting platform bias and algorithmic risks.

Current evaluation methods primarily rely on offline and online experiments [17]. Offline evaluation, utilizing large-scale datasets and historical user interaction data, is fast and cost-effective, suitable for early-stage algorithm validation [8]. However, it struggles with evolving user behavior and dynamic system adaptation. Online evaluation, often conducted through A/B testing in production environments, provides accurate feedback on user interactions but is costly, time-consuming, and ethically challenging [6]. Both methods typically require developers to have access to the system's source code or API, further constraining the third-party evaluation, such as competitive analysis and research studies.

To overcome these limitations, **simulation-based evaluation** [2, 4, 5, 19] has been proposed, using user simulators to approximate online experiments in offline settings. This approach balances the convenience of offline evaluation with the accuracy of online methods, but it faces challenges in modeling complex user behavior [20]. **The advent of large language model agents (LLMAs) has provided a powerful tool for simulation-based evaluation** [21, 27]. LLMAs excel in language understanding and generation, and with memory mechanisms and self-reflection strategies, they can simulate individual user behavior and even self-organizing social dynamics [13, 29]. However, evaluating continuous user interactions with real online news recommender systems through LLMAs still faces challenges in the following two aspects: **(1) Modeling Exploration-Exploitation Dynamics:** Current approaches often face difficulties in distinguishing between exploratory behavior (seeking novelty) and exploitative behavior (engaging with established interests). Absent this differentiation, the agent's memory may gradually accrue noise from unproductive exploratory actions, which could compromise the precision and consistency of simulating sustained user interactions, potentially affecting the reliability of the evaluation. **(2) Representation of Real-World Environments:** Limited access to the source code or APIs of operational recommender systems frequently leads researchers to employ simplified virtual sandbox environments. Such environments struggle to reflect the dynamic and adaptive nature of real-world systems, which respond in real-time to user interactions and feedback loops, thereby constraining the authenticity and utility of the simulated interactions.

In this paper, we propose EvalAgent, a large language model agent system for evaluating real-world news recommender systems. Leveraging the capabilities of LLMA, EvalAgent introduces Stable Memory (StM) to model users' exploration-exploitation dynamics, thereby ensuring consistent and reliable interest representation. Furthermore, EvalAgent incorporates an Environment Interaction Framework (EIF), which enables seamless interaction between the agent and operational recommender systems.

Our contributions are summarized as follows:

- We introduce EvalAgent, an innovative large language model-based agent system designed to simulate continuous user interactions within real-world online news recommender systems. The proposed Stable Memory (StM) effectively models exploration-exploitation dynamics, ensuring improved stability and precision in interaction simulations.
- We develop the Environment Interaction Framework (EIF), a framework that facilitates seamless and realistic interactions between the agent and operational recommender systems, thereby improving the applicability of simulation-based evaluation.
- We demonstrate the efficacy of EvalAgent through comprehensive experiments and user studies, with open-source code [1] provided to support further research and development in recommender system evaluation.

[1] https://github.com/aSeriousCoder/EvalAgent

2 Related Work

Early efforts in simulation-based evaluation primarily utilized learning-based simulators [10–12, 31], which employed tailored model architectures and learning techniques to model user behavior in specific contexts, such as reinforcement learning and data augmentation.

With the advent of large language models (LLMs), simulation approaches leveraging LLM-based agents have gained prominence due to their superior accuracy, interpretability, and adaptability across diverse scenarios [21, 26]. These works highlight that sophisticated memory systems, paired with the generative and reasoning capabilities of LLMs, enable highly accurate and human-like behavioral simulations [13, 22]. For example, Agent4Rec [28] introduces a dual-memory architecture incorporating factual and emotional components to simulate user experiences in consumption domains like movies, books, and games. RecAgent [23] simultaneously models and evaluates user behaviors in both recommender systems and social networks. AgentCF [30] introduces a collaborative learning framework with autonomous LLMAs to enhance the system's perception and modeling of both the users and items. iEvalLM [24] and CSHI [32] propose interactive simulation frameworks to evaluate conversational recommender systems.

Furthermore, user-simulation LLMAs have been employed as personalized recommendation assistants to enhance recommendation quality. For instance, RAH [16] introduces a multi-agent system-based recommendation assistant that empowers users with greater control over recommended content. RecMind [25] proposes an LLMA with external knowledge integration and tool-using capabilities, enabling zero-shot personalized recommendations. Similarly, DiscomfortFilter [9] leverages an LLMA to filter out undesirable content, thereby improving the quality of recommender systems.

However, these approaches often underexplore the underlying motivations of user interactions, particularly the exploration-exploitation balance, which can limit their capacity to ensure consistency and robustness in long-term interaction simulations. Furthermore, the absence of comprehensive modeling for real-world recommender system environments restricts these methods primarily to offline datasets, posing challenges for validation in real-world contexts.

3 Preliminaries

Definition 1 (News Recommender System). *A news recommender system is a system that recommends news articles to users based on their interests and preferences. This work specifically focuses on the prevalent news feed format, common to platforms like Toutiao and Twitter, where personalized recommendations derived from users' historical and real-time interactions are presented as a continuously updated, single-column waterfall stream of article cards.*

Definition 2 (Large Language Model Agent). *A Large Language Model Agent (LLMA) leverages a foundational large language model to autonomously perceive, reason, and interact within an environment. Its core components typically comprise the LLM (for reasoning and learning), memory (for retaining contextual history), and actions (for engaging with the environment). In this study, we focus on designing an action-decision workflow and memory architecture, alongside developing an automated perception and operation manager tailored for real-world application environments. Our objective*

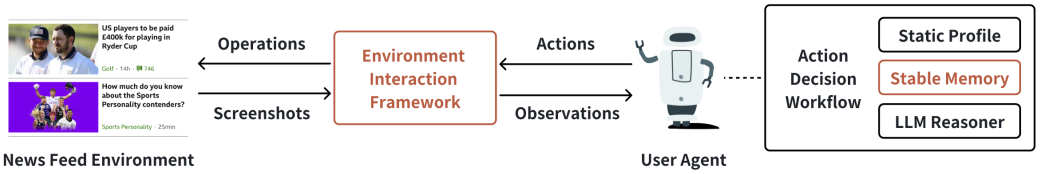

Figure 1: The overall architecture of the EvalAgent. The red components (Stable Memory and Environment Interaction Framework) represent the core contributions of this paper.

is to create an LLMA that simulates user behavior in news feeds, specifically determining whether to scroll or select articles based on its internal objectives and preferences.

Definition 3 (Noise in User Memory). *In the context of user behavior simulation, we define "noise" in user memory as irrelevant or inconsistent information accumulated in the memory system that does not align with the user's preferences or behavioral patterns. Specifically, noise arises from exploratory actions, which are user interactions with news articles that deviate from their established interests, often driven by curiosity or external influences (e.g., trending topics or social recommendations). Exploratory interactions may introduce variability in the memory system, potentially decreasing the accuracy of preference modeling if not properly managed.*

4 Methodology

This section introduces the overall architecture of EvalAgent, along with detailed descriptions of its two core components: Stable Memory and the Environment Interaction Framework.

4.1 Overall Architecture

As illustrated in Figure 1, EvalAgent conforms to the standard LLMA architecture defined in Definition 2, integrating static profiles, dynamic memories, and LLM reasoner within an action-decision workflow. It engages dynamically with the recommender system environment through the Environment Interaction Framework (EIF). The process initiates with the EIF parsing the news feed into textual news cards, which serve as the agent's environmental observations. Subsequently, the agent retrieves pertinent information from its memory: short-term memory recalls previously read related articles, while long-term memory retrieves relevant user preferences. Informed by these dynamic memory retrievals and static user profiles, the agent employs the prompt outlined in Figure 2 to decide whether to click on or swipe past a news card. The selected action is executed as a sequence of device operations through the EIF's operation API, completing the "observation-decision-action" cycle.

4.2 Stable Memory (StM)

The Stable Memory module introduces an innovative approach to modeling user interaction patterns by adeptly distinguishing between exploratory and exploitative behaviors, thereby facilitating stable and consistent interaction simulations for large language model agents within recommender systems. Comprising short-term and long-term memory components, the StM module is designed to capture both immediate and evolving user interests. Short-term

Action-decision Prompt of EvalAgent

You are a user browsing a news app, with your personal information defined as {agent profile}.
You are currently viewing a news article:
{candidate news card}
You have previously read the following related articles:
{retrieved news cards from short-term memory}
Your long-term preferences include:
{retrieved preference description from long-term memory}

Describe your reaction to this news article and decide whether to click for more information.
Provide your response in the following format:
{ "reflection": 2-3 sentences describing your reaction, "action": 1 for click, 0 for skip }

Figure 2: Prompt of action decision in EvalAgent.

memory records user interactions by storing and forgetting clicked news articles, while long-term memory maintains a set of dynamically updated preference descriptions that reflect the user's sustained interests. Operationally, the StM module begins by generating semantic embeddings for clicked news articles. It then assesses the exploratory or exploitative nature of each click by evaluating the semantic density distribution within the short-term memory. By incorporating a temporal decay factor, StM implements a dynamic forgetting mechanism to maintain memory stability during continuous interactions. Additionally, the module incrementally updates long-term memory, adeptly tracking the evolution of user preferences over time.

4.2.1 Semantic Encoding. Upon a user clicking on a news article, the article's textual content is converted into a dense vector $\mathbf{v}_i \in \mathbb{R}^d$ using a pre-trained sentence embedding model, such as Sentence-BERT [14]. This embedding captures the article's semantic content within a high-dimensional space, enabling the quantification of article relatedness through vector similarity. Each click is represented by the article's semantic vector and a timestamp t_i recording the interaction time.

4.2.2 Explore-Exploit Modeling. User interactions can be categorized into exploitative actions, which align with established preferences, and exploratory actions, which reflect curiosity-driven engagement with less familiar topics. Exploratory actions, while valuable for discovering new interests, introduce noise into the memory system by adding information that may not be representative of the user's long-term preferences. This noise can accumulate over time, leading to degraded performance in preference alignment. To address this, EvalAgent employs a k-nearest neighbor (k-NN) approach combined with a Gaussian kernel to compute the local density ρ_i of each news article n_i in the short-term memory.

Long-term Memory Incremental Update Prompt of EvalAgent

You are an intelligent user modeling assistant. Your goal is to incrementally update the user's high-level information based on the news reading record.
The high-level information is organized as a list of concise descriptions reflecting the user's interests, preferences, and behavioral patterns, detailed as:
{recalled descriptions}
The user just read the news below:
{triggered news article}
Rules:
1. Descriptions should be concise and clear, avoiding unnecessary complexity.
2. You can update the descriptions by modify, add or merge.
3. The output should be a list of updated descriptions with json format

Figure 3: Prompt of long-term memory incremental update in EvalAgent.

Compared to global density estimation methods, k-NN-based local density estimation can alleviate overfitting under the sparsity constraint of user memory, where the number of memory items N is much smaller than the embedding dimension D. The Gaussian kernel further ensures smooth density estimates, balancing sensitivity to local patterns with generalization across diverse topics. This process can be formulated as:

$$\rho_i = \frac{1}{k} \sum_{j \in \mathcal{N}_k(i)} \exp\left(-\frac{\sigma(\|\mathbf{v}_i - \mathbf{v}_j\|^2)}{\sigma(h^2)} \right), \tag{1}$$

where $\mathbf{v}_i$ is the semantic embedding of the news article n_i, $\mathcal{N}_k(i)$ is the set of k nearest neighbors of n_i in the semantic space, and $\sigma(\cdot)$ is a sigmoid function to avoid extreme values. The bandwidth h of the Gaussian kernel is estimated using Silverman's Rule of Thumb [18]:

$$h = \sqrt{n^{-1/(d+4)} \operatorname{tr}(\mathbf{S})/d}, \tag{2}$$

where n is the current number of articles in the short-term memory, d is the embedding dimension, and $\mathbf{S}$ is the covariance matrix of the current set of article embeddings. The explore-exploit tendency e_i is subsequently defined as:

$$e_i = 1 - \rho_i, \tag{3}$$

where $e_i \in [0, 1)$, with higher values indicating a stronger inclination towards exploratory behavior, which is less aligned with the user's core interests and more likely to introduce noise into the memory system.

4.2.3 Adaptive Forgetting. Motivated by the fact that the human brain actively forgets unimportant or infrequently used information to optimize storage space [3], StM incorporates an adaptive forgetting mechanism, to keep memory stability and prioritize the retention of information most relevant to the user's evolving preferences.

This mechanism comprises two components: a balanced forgetting strategy to constrain the memory size within a reasonable range, and a natural forgetting process to clear out outdated memories, thereby preventing overall distribution collapse. The probability p_i of forgetting article n_i is formulated as:

$$p_i = \begin{cases} \lambda \cdot (1 - \exp(-\frac{t - t_i}{\tau})), & \text{if } n \le N \\ \frac{(n-N)e_i}{\sum_{j=1}^{n} e_j} + \lambda \cdot (1 - \exp(-\frac{t - t_i}{\tau})), & \text{if } n > N \end{cases} \tag{4}$$

where N is the expected maximum memory capacity. The first term under the $n > N$ condition implements the balanced forgetting strategy, where articles exhibiting a higher exploratory tendency are more likely to be removed. The balanced forgetting strategy dynamically determines the forgetting intensity, aiming for an expected memory size after its execution of:

$$E[n'] = n - \sum_{i=1}^{n} \frac{(n-N)e_i}{\sum_{j=1}^{n} e_j} = N. \tag{5}$$

The second term introduces a natural forgetting process, where λ is a weight parameter (a small positive value) controlling the influence of time, t is the current time, and τ is a time constant that determines the rate of decay.

4.2.4 Long-Term Memory Updating. We utilize semantic-density-based probabilistic triggers to integrate short-term interactions into long-term memory. For each article i in the short-term memory, the probability of it triggering an update to the long-term memory, $p_{trigger}(i)$, is determined by a Sigmoid function centered around the average density $\bar{\rho}$ of all articles currently in the short-term memory:

$$p_{trigger}(i) = \frac{1}{1 + \exp(-\beta(\rho_i - \bar{\rho}))} \tag{6}$$

Here, $\bar{\rho} = \frac{1}{n} \sum_{j=1}^{n} \rho_j$, and β is a parameter that controls the steepness of the Sigmoid curve with a default value of 1. Articles residing in denser semantic regions (higher ρ_i), which are indicative of exploitative clicks aligned with established interests, have a higher likelihood of triggering a long-term memory update. Each article is marked after contributing to long-term memory to prevent repeated updates.

For each triggered article $n_i \in M_s$, the system recalls descriptions from the long-term memory M_l with a cosine similarity exceeding a threshold ϵ. This process can be formated as:

$$D_{recalled}(n_i) = \{d_j \in M_l \mid \text{similarity}(n_i, d_j) > \epsilon\} \tag{7}$$

The recalled descriptions are then updated incrementally with the prompt outlined in Figure 3.

4.2.5 Memory Retrieval. Given a query news article n_q,StM retrieves the most relevant items from both short-term and long-term memory. From the short-term memory, it selects the top-K news articles based on the inverse Gaussian distance. From the long-term memory, it retrieves the top-T preference descriptions using cosine similarity. The retrieved sets of K articles and T descriptions are then used by the agent's decision-making process outlined in Figure 2.

4.3 Environment Interaction Framework (EIF)

The Environment Interaction Framework serves as a bridge facilitating the agent's interaction with real-world news recommender systems. As illustrated in Figure 4, it comprises three key components: the Device Manager, which oversees the execution and operation of news recommender system applications on the device; the News Feed Parser, which transforms real news feed screenshots from mobile devices into standardized news cards; and the Device Operation Chains, which translate the agent's action decision into a sequence of executable operations on the device. EIF provides a

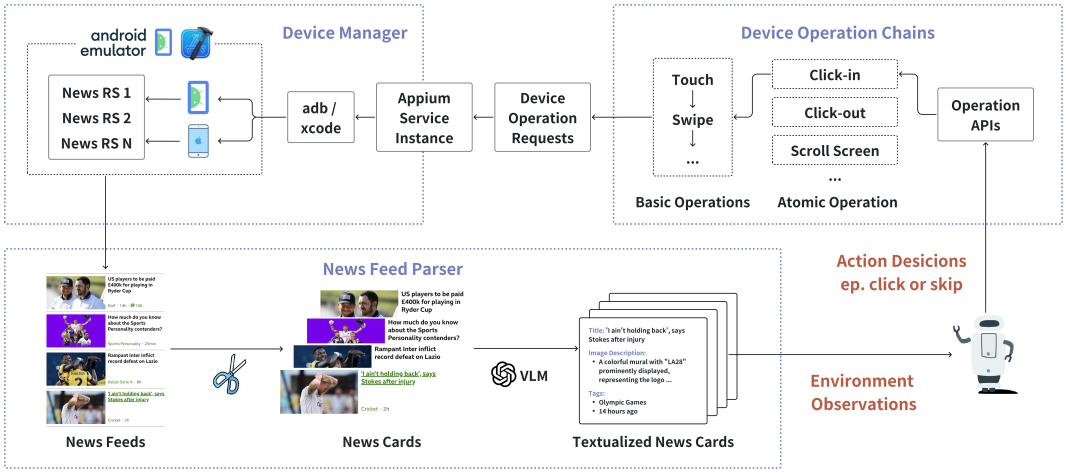

Figure 4: The architecture of Environment Interaction Framework (EIF), in which EvalAgent parses real news feeds on mobile devices into standardized news cards, and maps the agent's actions into a sequence of operational commands executed on the mobile device.

unified environment modeling for different single-column waterfall style news recommender systems and offers a interaction interface for the LLMAs.

4.3.1 Device Manager. To enhance the applicability of EvalAgent, the Device Manager provides unified management for devices running different operating systems (iOS or Android) and diverse platforms (e.g., BBC News, Tencent News). Specifically, it facilitates communication with Android or iOS devices using adb or xcode, respectively, while leveraging Appium to supply screenshots to the News Feed Parser, receive operation commands from the Device Operation Chains, and handle exceptions such as pop-ups or crashes.

4.3.2 News Feed Parser. As depicted by the rightward data flow in the lower section of Figure 4, upon receiving a screenshot, the News Feed Parser first segments the page into individual news cards. It then employs a Visual Language Model (VLM), such as GPT-4o, to extract key information from each card, including: 1) title information, 2) image descriptions, and 3) tags such as topic, hashtag, and publish-time, enabling the LLMA to "comprehend" the visual news content.

4.3.3 Device Operation Chains. The Device Operation Chains are structured across three hierarchical layers. The foundational basic operation layer consists of operation commands receivable by the Device Manager, such as touch and swipe. Building upon this, the atomic operation layer defines the fundamental interaction units specific to news recommender systems, including actions like clicking into or out of a news article and scrolling the screen, while adapting to device parameters (e.g., screen size) and recommender system parameters (e.g., news card layout). At the top, the operation API layer interfaces directly with the agent's action decisions, such as clicking the current news or skipping it, translating these into sequences of atomic operations.

5 Evaluation

We first validate the effectiveness of EvalAgent through offline experiments, then conduct a user study to assess whether EvalAgent can accurately reflect user interaction patterns with real-world news recommender systems.

5.1 Offline Experiments

5.1.1 Task. We validate the effectiveness of EvalAgent in aligning with user preferences through a ranking task. Specifically, for a user u, we first construct the Memory modules based on their historical news clicks. Then, for incoming testing news sequence, the agent re-ranks the sequence based on predicted click probabilities. Finally, we evaluate the ranking performance with metrics including AUC, MRR, and nDCG@5. These metrics comprehensively evaluate the agent's capacity to rank content that aligns with the user's genuine interests from multiple perspectives, collectively reflecting the extent to which the agent modelling corresponds to the user's actual preferences.

5.1.2 Dataset. We conduct offline experiments using the MIND and Adressa datasets, which are widely adopted in the news recommendation domain. Following previous works [9], we sampled a subset of users from each dataset for experimentation to constrain API call costs. To demonstrate the stability of EvalAgent in modeling continuous interactions, we first group users based on their historical interaction frequency into categories: $[0, 10)$, $[10, 20)$, and up to $[100, +\infty)$ for the MIND dataset, up to $[40, +\infty)$ for the Adressa dataset due to their different user distributions. Then, from each group, we randomly selected users, ensuring a total of at least 1,000 positive testing samples per group. We resample the testing data to achieve a positive-to-negative sample ratio of $1 : 4$ for standard metrics calculation. For the Adressa dataset, which only contains positive samples, we construct negative samples for the candidate news sequences through global random sampling.

Table 1: User preference alignment accuracy comparison against different baselines and density estimation methods.

	Method	MIND Dataset			Adressa Dataset		
		AUC	MRR	nDCG@5	AUC	MRR	nDCG@5
Baseline Methods	Latest N	0.6466	0.5634	0.6719	0.7062	0.6085	0.7068
	Similarity	0.6620	0.5881	0.6923	0.7188	0.6254	0.7163
	Emotional	0.6864	0.5958	0.7078	0.7325	0.6429	0.7318
	Hierarchical	0.7107	0.6221	0.7243	0.7518	0.6835	0.7614
	EvalAgent	**0.7283**	**0.6368**	**0.7331**	**0.7681**	**0.7027**	**0.7818**
Density Estimation Methods	Global KDE	0.7066	0.6179	0.7204	0.7441	0.6765	0.7566
	k-NN + Cosine	0.7180	0.6276	0.7245	0.7605	0.6925	0.7759
	k-NN + Distance	0.7225	0.6313	0.7287	0.7631	0.6972	0.7780
Ablation Variants	EvalAgent w/o EE	0.7121	0.6258	0.7267	0.7539	0.6876	0.7663
	EvalAgent w/o LTM	0.6663	0.5913	0.6948	0.7201	0.6323	0.7233

5.1.3 Baselines. We select four baseline memory architectures of LLM-based User Simulators for a comprehensive comparison: **Latest N Memory**: A memory module that maintains the N most recent news articles with the First-In-First-Out (FIFO) rule. This approach represents a simple, recency-focused memory design, widely used for its computational efficiency and ability to capture short-term user interests; **Similarity Memory**: A memory module that preserves all clicked news articles and retrieves them based on their similarity to the querying news. This design is representative of context-aware memory architectures, commonly employed in retrieval-augmented generation (RAG) tasks for its ability to dynamically align recommendations with historical user interactions; **Emotional Memory**: A dual-structure memory module, proposed by Agent4Rec [28], that integrates factual memories of clicked items with emotional responses. This design is a representative work of item-level memory enhancement, capturing fine-grained user preferences by modeling subjective affective responses alongside objective interaction history. **Hierarchical Memory**: A hierarchical memory module consisting of short-term and long-term components, employing a self-reflection mechanism to build upward. Widely adopted by advanced social agents like Generative Agents [13] and RecAgent [23], this architecture represents a classic approach to capturing complex, long-term user behavior patterns. EvalAgent extends the design of Hierarchical Memory by introducing Stable Memory, which supports explore-exploit modeling, to enhance accuracy and stability, and building Environment Interaction Framework, which is used to link the agent with real-world news recommender systems.

5.1.4 Implementation Details. The number of k-nearest neighbors is set as 5. The expected short-term memory scale N is set as 20. The weight parameter λ and time constant τ in the natural forgetting process are set as 0.1 and 3600 respectively. The long-term memory update threshold ϵ is set as 0.5. To eliminate the influence of different llm models on evaluation results, we conduct experiments using three widely-used models as the base: *GPT-4o*, *Deepseek-V3*, and *Qwen3-235b-a22b*, and report the average performance across these three base models. The embedding model is set as *text-embedding-3-small* from openai for all experiments.

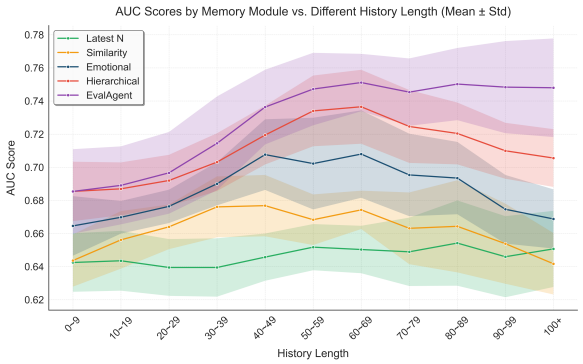

Figure 5: AUC comparison of different Memory modules with different historical interaction frequency on the MIND dataset.

5.1.5 Results.

A. Baseline Comparison.

Table 1 presents the comprehensive performance comparison of user preference alignment accuracy across different memory modules. The results demonstrate that **EvalAgent consistently achieves superior performance across all evaluation metrics, substantiating its effectiveness in modeling user preferences and maintaining alignment with user interests**. To elucidate the underlying mechanisms of EvalAgent's performance enhancement, we further analyze the AUC curves across different Memory modules, stratified by user historical interaction sequence length on the MIND dataset, as shown in Figure 5, which yields two key insights:

Explore-exploit modelling bolsters stability in user preference modeling during prolonged interactions: While baseline models suffer from systematic performance degradation as historical interaction sequences surpass a critical threshold, EvalAgent maintains consistent performance. This empirical evidence supports our hypothesis that exploratory clicks may lead to systematic noise accumulation in memory systems. Specifically, within the expected short-term memory scale N, EvalAgent and Hierarchical

operate on similar theoretical foundations and deliver comparable results. However, when sequence lengths exceed N, EvalAgent demonstrates superior resilience against memory noise accumulation, thereby confirming its enhanced stability.

User Reflection serve as a Pivotal Mechanism for Enhancing User Preference Alignment Accuracy. The results also demonstrate that reflection mechanisms at various levels significantly improve the accuracy of user preference alignment. Specifically, EvalAgent and Hierarchical leverage reflection to construct high-level long-term memory of user interaction patterns, while Emotional employs reflection to build news-level emotional response memory. Despite differences in their mechanisms and levels, experimental results consistently show that these reflection-based approaches outperform non-reflective methods, such as Latest N Memory and Similarity Memory.

B. Sensitivity Analysis.

This subsection analyzes the impact of different density estimation methods on the accuracy of EvalAgent. We compare our approach (k-NN + Gaussian Kernel) against three other density estimation methods: (A) **Global KDE**: A global density estimation method implemented via Gaussian Mixture Models (with the number of components set to 5). (B) **k-NN + Cosine**: A local density estimation using k-NN, where the distance metric is defined as 1 minus the cosine similarity. (C) **k-NN + Distance**: A local density estimation using k-NN with the Euclidean distance directly.

Table 1 presents the comparison results, where our approach achieves the best performance. Due to the sparse memory constraint, where the number of memory items N is much smaller than the embedding dimension D, Global KDE suffers from overfitting caused by insufficient samples, leading to lower performance. k-NN + Cosine exhibits slightly lower performance than k-NN + Distance due to the nonlinear mapping relationship between similarity and distance, which distorts the spatial density estimation. EvalAgent builds upon k-NN + Distance by further modeling the density with a Gaussian kernel, effectively smoothing fluctuations and achieving the best overall performance.

C. Ablation Study.

We further analyze the impact of different components on the accuracy. Specifically, we ablate the explore-exploit modeling (along with the adaptive forgetting mechanism) and the long-term memory from EvalAgent. EvalAgent w/o EE can be considered as an incrementally updated Hierarchical Memory, and due to the finer update granularity, it achieves slightly better performance. EvalAgent w/o LTM loses the reflection of users' high-level interests, leading to a significant drop in performance. But it can also be viewed as Similarity Memory with explore-exploit modeling, which achieves a clear performance improvement.

5.2 User Study

5.2.1 Experiment Design. To evaluate the effectiveness of EvalAgent in real-world applications, we designed a comprehensive experiment comprising user recruitment, data collection, user interaction simulation, and user evaluation. The experiment was conducted on Tencent News [2], one of China's most widely used news platforms. The detailed experimental procedure is outlined below:

Step 1: Participant Recruitment and Data Collection. Participants were recruited through a combination of open online calls and snowball sampling. The participants' ages range from 18 to 63 years, with 11 participants being male and 9 being female, detailed demographics are shown in Table 2. Each participant was required to submit a video of at least 30 minutes, capturing their usage of Tencent News. From these videos, we extracted the historical news click data of each participant as the historical data for building user-specific agents, enabling the modeling of individual user preferences based on real-world interactions.

Step 2: User Interaction Simulation. To simulate user interactions, agents interacted with the feed stream of Tencent News for one hour via the Environment Interaction Framework (EIF). To eliminate potential affect from device caching or account histories, we created an independent virtual device (Google Pixel 8, Android 14.0, provided by Android Studio) for each agent, ensuring no account login was used. The interaction simulation was divided into two phases: **Synchronization Phase (First 30 Minutes):** During this phase, agents conveyed preference signals to the recommendation system through click behaviors, with their memory states kept frozen; **Validation Phase (Second 30 Minutes):** Agents actively interacted with the feed and updated their memory. The interaction data from this phase were used to assess the agents' accuracy in the following steps. In addition to the outlined action-decision prompt in Section 4.1, agents were required to provide explanations for their decisions. These explanations facilitated qualitative analysis of the alignment between the agents' decision-making processes and users' actual preferences. For each participant, we conducted parallel experiments by constructing agents based on EvalAgent and Hierarchical (the best-performing baseline from offline experiments).

Step 3: User Evaluation. Using the data collected from previous step, participants were invited to complete two evaluation tasks to assess the agents' performance: **Preference Test:** Participants were presented with 20 news pairs (10 from EvalAgent and 10 from Hierarchical). Each pair consisted of one news item the agent simulated clicking and one it did not (presented in random order). Participants selected the news item in each pair that better aligned with their preferences; **Explanation Test:** Participants reviewed 10 news items, each accompanied by decision explanations from both the EvalAgent and Hierarchical. They rated their preferences on the two explanations with a 7-point Likert scale and provided textual justifications for their ratings.

5.2.2 Results.

A. Results of the Preference Test.

We analyzed discrepancies in news topic distribution between historical data and simulation results. Figure 6 illustrates the distribution of clicked news topics across participant groups, segmented by gender and age, comparing historical data (orange) with simulated distributions from the Hierarchical Memory (red) and EvalAgent (purple). **EvalAgent demonstrates superior alignment**

[2] https://news.qq.com/

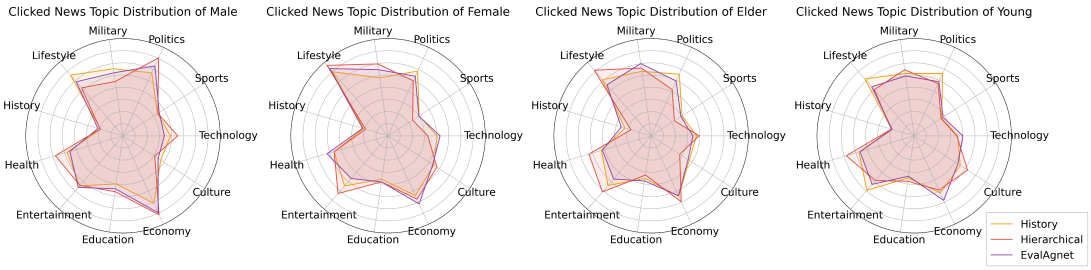

Figure 6: Clicked news topic distribution of participants grouped by gender and age. Historical distribution is colored in orange, simulated distribution of Hierarchical is colored in red, and simulated distribution of EvalAgent is colored in purple.

Table 2: Demographics of participants in the user study.

ID	Age	Gender	ID	Age	Gender
P01	24	Male	P11	63	Male
P02	26	Male	P12	18	Male
P03	25	Male	P13	42	Female
P04	30	Female	P14	45	Male
P05	23	Male	P15	56	Female
P06	32	Male	P16	22	Female
P07	34	Female	P17	55	Male
P08	46	Male	P18	53	Female
P09	20	Female	P19	36	Male
P10	41	Female	P20	50	Female

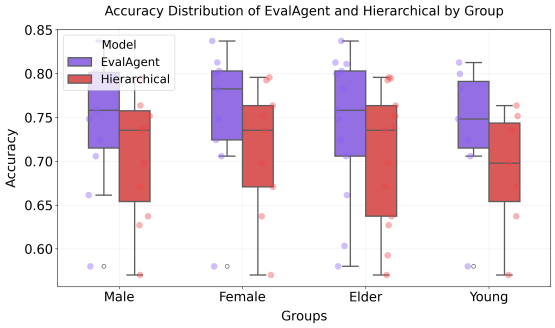

Figure 7: Accuracy comparison of EvalAgent and Hierarchical on the preference test grouped by gender and age. Results of EvalAgent are colored in purple, and results of Hierarchical are colored in red.

with historical data across all groups, particularly for topics with varying degrees of user preference clarity. For example, in the female group, both models accurately capture strong engagement with Lifestyle topics and minimal interest in History, where preferences are well-defined. However, for topics like Technology, where user preferences are less distinct, EvalAgent generally outperforms Hierarchical. This pattern underscores the efficacy of EvalAgent's Explore-Exploit modeling approach, which enhances the accuracy and stability of user preference alignment.

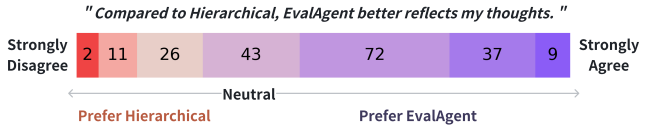

Figure 8: User preference of Hierarchical vs. EvalAgent on accurately reflecting their minds over the 20×10 cases in Explanation Test.

We subsequently evaluated the accuracy of EvalAgent and Hierarchical in simulating user click behavior. Figure 7 presents a detailed comparison of their performance in the Preference Test, with accuracy distributions visualized across gender and age groups. The results highlight that **EvalAgent consistently outperforms the Hierarchical model in click simulation accuracy across all evaluated groups**. Demographically, EvalAgent demonstrates better performance among female and younger users, achieving higher accuracy or lower variance in capturing their click patterns. In contrast, the Hierarchical model exhibits relatively better performance among female and elderly users. This difference may indicate that female and younger users exhibit a stronger curiosity toward a diverse range of news topics, and the Explore-Exploit modeling strategy employed by EvalAgent may deliver a more significant improvement for these groups.

B. Results of the Explanation Test.

Figure 8 illustrates user preferences regarding the accuracy of the Hierarchical model versus EvalAgent in reflecting their thought processes across the 20×10 cases in the Explanation Test. **The majority of respondents (118 out of 200, or 59%) favor EvalAgent**, with 9 strongly agreeing, 37 agreeing, and 72 weakly agreeing that it more effectively mirrors their cognitive processes. In contrast, only 39 respondents (19.5%) prefer the Hierarchical model, with 2 strongly agreeing, 11 agreeing and 25 weakly agreeing. Additionally, a notable portion of participants (43 responses, or 21.5%) adopt a neutral stance, indicating some uncertainty in their preference. This distribution highlights EvalAgent's superior ability to align with user cognition, likely attributable to its enhanced memory management capabilities, thereby outperforming the Hierarchical model in this evaluative context.

To further explore the reasons for participants' preference for EvalAgent and their views on LLM-based recommendation interaction agents, we conducted a thematic analysis of the textual justifications collected from the Explanation Test. This analysis revealed key insights explaining the superiority of EvalAgent over Hierarchical and highlight areas for improvement:

Insight 1: Interest misalignment is more frequent in Hierarchical. Participants frequently noted that Hierarchical's explanations misrepresented their interests, suggesting a bias in its interest modeling. Specifically, 7 out of 20 participants (35%) reported at least one instance where Hierarchical's explanation referenced a topic they were not interested in, compared to only 4 participants (20%) for EvalAgent. For example, one participant stated, "*Hierarchical assumed I care about tech innovations multiple times, but I don't*" (P02).

Insight 2: Overly objective decision-making of LLM-based agents. Both EvalAgent and Hierarchical produced explanations that participants perceived as overly objective, often prioritizing analytical reasoning over subjective user preferences. Participants acknowledged the agents' strong analytical capabilities, with one noting, "*The explanation about the importance of political news was quite logical and well-reasoned*" (P17). However, many felt the explanations failed to account for personal curiosity or emotional drivers of news consumption. For instance, a participant commented, "*It's like the agent is writing a thesis on why this news matters, but it misses why I'd actually click on it—because it sounds fun*" (P09). Another stated, "*The reasoning feels like an academic abstract, not my thought process*" (P04). This theme aligns with offline experiment findings, where Emotional Memory improved interest alignment by explicitly modeling users' emotional responses.

Insight 3: Hallucinations in explanations affect user experience. Despite the use of memory mechanisms to mitigate hallucinations, both agents occasionally produced explanations that included fabricated or misaligned content, undermining their credibility. Participants reported two types of hallucinations: experiential hallucinations, where agents referenced non-existent user experiences, and value-based hallucinations, where agents attributed irrelevant motivations. For example, one participant noted, "*The explanation mentioned a past experience I never had*" (P15). Another remarked, "*It said I'd want to read this to 'gain social influence at work,' which isn't something I think about*" (P13).

6 Discussion

6.1 Potential Application Value

EvalAgent establishes a framework for simulating user interactions with news recommender systems, seamlessly integrated with real-world platforms through the Environment Interaction Framework. This simulation-based approach provides a dynamic and controllable alternative to traditional offline and online evaluation methods. Potential application scenarios may include: **Early-Stage System Design**: EvalAgent facilitates realistic simulations of user behavior during the development phase, enabling developers to optimize recommendation algorithms without disrupting the users. **Evaluation of Sensitive Content**: By simulating interactions for specific demographics (e.g., elderly or teenagers) or with sensitive content (e.g., hate speech), EvalAgent offers a controlled environment to address ethical concerns and recruitment challenges for these scenarios. **Third-Party Evaluation**: EvalAgent provides a tool for external entities, such as research institutions, to independently assess recommender systems' performance without requiring proprietary API access.

6.2 Robustness of EIF

EIF targets single-column waterfall news feeds, common in mobile apps such as BBC News and Tencent News. Cross-platform UI variations are handled by the VLM, which semantically parses screenshots with minimal configuration changes. Its main limitation is the inability to process video content, now prevalent in mobile news. While keyframe extraction could enable basic video parsing, it may miss temporal continuity and key details, highlighting the need for more advanced video understanding methods.

6.3 Limitations and Future Directions

6.3.1 Emotional Modeling. While LLMAs excel at objective analysis, they currently struggle to accurately simulate human emotional responses and empathy. Future work may focus on enhancing LLMs' inherent emotional understanding capabilities for more realistic news consumption behavior simulation.

6.3.2 Perception of Out-of-System Events. A key limitation leading to LLMA hallucinations stems from their restricted perception, primarily limited to events observed within the recommender system. Real-world behavior is shaped by external factors such as social interactions or personal changes. Introducing a "group shared memory" could infer such influences from aggregated user data, enriching contextual awareness.

6.3.3 Scalability and Efficiency. Current experiments simulate thousands of users with short interaction histories (<100). Real systems involve far larger scales and longer histories, creating high inference and resource costs. Fine-tuned Small Language Models (SLMs) may deliver comparable performance with significantly improved efficiency.

7 Conclusion

This paper introduces EvalAgent, a large language model agent system designed for real-world news recommender system evaluation. By employing Stable Memory (StM) for modeling user exploration-exploitation dynamics and an Environment Interaction Framework (EIF) for direct engagement with real systems, EvalAgent aims to provide a accurate, stable, and ethically responsible evaluation framework. Comprehensive experiments and user studies validate EvalAgent's efficacy and potential application value.

Acknowledgments

This work is supported by National Natural Science Foundation of China (NSFC) under the Grant No. 62172106. Peng Zhang is a faculty of College of Computer Science and Artificial Intelligence, Fudan University. Tun Lu is a faculty of College of Computer Science and Artificial Intelligence, Shanghai Key Laboratory of Data Science, Fudan Institute on Aging, MOE Laboratory for National Development and Intelligent Governance, and Shanghai Institute of Intelligent Electronics & Systems, Fudan University.

GenAI Usage Disclosure

Generative AI tools were used solely for language refinement after the main content of the paper was completed. No GenAI tools were used in the design, implementation, analysis, or writing of the research content itself. The authors are fully accountable for the content presented herein.

References

[1] 2024. News Platform Fact Sheet. https://www.pewresearch.org/journalism/fact-sheet/news-platform-fact-sheet/

[2] Giovanni Luca Ciampaglia, Azadeh Nematzadeh, Filippo Menczer, and Alessandro Flammini. 2018. How algorithmic popularity bias hinders or promotes quality. *Scientific reports* 8, 1 (2018), 15951.

[3] Hermann Ebbinghaus. 2013. [image] Memory: A Contribution to Experimental Psychology. *Annals of neurosciences* 20, 4 (2013), 155.

[4] Daniel Geschke, Jan Lorenz, and Peter Holtz. 2019. The triple-filter bubble: Using agent-based modelling to test a meta-theoretical framework for the emergence of filter bubbles and echo chambers. *British Journal of Social Psychology* 58, 1 (2019), 129–149.

[5] Ray Jiang, Silvia Chiappa, Tor Lattimore, András György, and Pushmeet Kohli. 2019. Degenerate feedback loops in recommender systems. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*. 383–390.

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

[7] Joseph Konstan and Loren Terveen. 2021. Human-centered recommender systems: Origins, advances, challenges, and opportunities. *AI Magazine* 42, 3 (2021), 31–42.

[8] Karl Krauth, Sarah Dean, Alex Zhao, Wenshuo Guo, Mihaela Curmei, Benjamin Recht, and Michael I Jordan. 2020. Do offline metrics predict online performance in recommender systems? *arXiv preprint arXiv:2011.07931* (2020).

[9] Jiahao Liu, Yiyang Shao, Peng Zhang, Dongsheng Li, Hansu Gu, Chao Chen, Longzhi Du, Tun Lu, and Ning Gu. 2025. Filtering Discomforting Recommendations with Large Language Models. In *Proceedings of the ACM on Web Conference 2025*. 3639–3650.

[10] Xufang Luo, Zheng Liu, Shitao Xiao, Xing Xie, and Dongsheng Li. 2022. Mindsim: user simulator for news recommenders. In *Proceedings of the ACM Web Conference 2022*. 2067–2077.

[11] James McInerney, Ehtsham Elahi, Justin Basilico, Yves Raimond, and Tony Jebara. 2021. Accordion: a trainable simulator for long-term interactive systems. In *Proceedings of the 15th ACM Conference on Recommender Systems*. 102–113.

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

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

[14] Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084* (2019).

[15] Francesco Ricci, Lior Rokach, and Bracha Shapira. 2021. Recommender systems: Techniques, applications, and challenges. *Recommender Systems Handbook* (2021), 1–35.

[16] Yubo Shu, Haonan Zhang, Hansu Gu, Peng Zhang, Tun Lu, Dongsheng Li, and Ning Gu. 2024. RAH! RecSys–Assistant–Human: A Human-Centered Recommendation Framework With LLM Agents. *IEEE Transactions on Computational Social Systems* (2024).

[17] Thiago Silveira, Min Zhang, Xiao Lin, Yiqun Liu, and Shaoping Ma. 2019. How good your recommender system is? A survey on evaluations in recommendation. *International Journal of Machine Learning and Cybernetics* 10 (2019), 813–831.

[18] Bernard W Silverman. 2018. *Density estimation for statistics and data analysis*. Routledge.

[19] Wenlong Sun, Sami Khenissi, Olfa Nasraoui, and Patrick Shafto. 2019. Debiasing the human-recommender system feedback loop in collaborative filtering. In *Companion Proceedings of The 2019 World Wide Web Conference*. 645–651.

[20] Petter Törnberg, Diliara Valeeva, Justus Uitermark, and Christopher Bail. 2023. Simulating social media using large language models to evaluate alternative news feed algorithms. *arXiv preprint arXiv:2310.05984* (2023).

[21] 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.

[22] Lei Wang, Jingsen Zhang, Hao Yang, Zhiyuan Chen, Jiakai Tang, Zeyu Zhang, Xu Chen, Yankai Lin, Ruihua Song, Wayne Xin Zhao, Jun Xu, Zhicheng Dou, Jun Wang, and Ji-Rong Wen. 2023. When Large Language Model based Agent Meets User Behavior Analysis: A Novel User Simulation Paradigm. *arXiv preprint arXiv:2306.02552* (2023).

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

[24] 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).

[25] Yancheng Wang, Ziyan Jiang, Zheng Chen, Fan Yang, Yingxue Zhou, Eunah Cho, Xing Fan, Xiaojiang Huang, Yanbin Lu, and Yingzhen Yang. 2023. Recmind: Large language model powered agent for recommendation. *arXiv preprint arXiv:2308.14296* (2023).

[26] 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).

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

[28] 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.

[29] Guangping Zhang, Dongsheng Li, Hansu Gu, Tun Lu, Li Shang, and Ning Gu. 2024. Simulating News Recommendation Ecosystems for Insights and Implications. *IEEE Transactions on Computational Social Systems* (2024).

[30] Junjie Zhang, Yupeng Hou, Ruobing Xie, Wenqi Sun, Julian McAuley, Wayne Xin Zhao, Leyu Lin, and Ji-Rong Wen. 2024. Agentcf: Collaborative learning with autonomous language agents for recommender systems. In *Proceedings of the ACM Web Conference 2024*. 3679–3689.

[31] 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.

[32] Lixi Zhu, Xiaowen Huang, and Jitao Sang. 2025. A LLM-based controllable, scalable, human-involved user simulator framework for conversational recommender systems. In *Proceedings of the ACM on Web Conference 2025*. 4653–4661.