# A Survey on LLM-powered Agents for Recommender Systems

Anonymous ACL submission

## Abstract

Recently, Large Language Models (LLMs) have demonstrated remarkable capabilities in natural language understanding, reasoning, and generation, prompting the recommendation community to leverage these powerful models to address fundamental challenges in traditional recommender systems, including limited comprehension of complex user intents, insufficient interaction capabilities, and inadequate recommendation interpretability. This survey presents a comprehensive synthesis of this rapidly evolving field. We consolidate existing studies into three paradigms: (i) recommender-oriented methods, which directly enhance core recommendation mechanisms; (ii) interaction-oriented methods, which conduct multi-turn conversations to elicit preferences and deliver interpretable explanations; and (iii) simulation-oriented methods, that model user-item interactions through multi-agent frameworks. Then, we dissect a four-module agent architecture: profile, memory, planning, and action. Then we review representative designs, public datasets, and evaluation protocols. Finally, we give the open challenges that impede real-world deployment, including cost-efficient inference, robust evaluation, and security.

## 1 Introduction

In the era of information explosion, recommender systems have become an indispensable component of digital platforms, helping users navigate through massive amounts of content across various domains. While traditional recommendation approaches (He et al., 2017) have achieved considerable success in providing personalized recommendations, they still face significant challenges, such as limited understanding of complex user intents, insufficient interaction capabilities, and the inability to provide interpretable recommendations (Zhu et al., 2024b).

Recent advancements in Large Language Models (LLMs) (Achiam et al., 2023) have sparked increasing interest in leveraging LLM-powered agents (Wang et al., 2024a) to address the aforementioned challenges in recommender systems. The integration of LLM-powered agents into recommender systems offers several compelling advantages over traditional approaches (Zhu et al., 2024b). First, LLM agents can understand complex user preferences and generate contextual recommendations through their sophisticated reasoning capabilities, enabling more nuanced decision-making beyond simple feature-based matching. Second, their natural language interaction abilities facilitate multi-turn conversations that proactively explore user interests and provide interpretable explanations, enhancing both recommendation accuracy and user experience. Third, these agents revolutionize user behavior simulation by generating more realistic user profiles that incorporate emotional states and temporal dynamics, enabling more effective system evaluation. Furthermore, the pre-trained knowledge and strong generalization capabilities of LLMs facilitate better knowledge transfer across domains, addressing persistent challenges such as cold-start (Shu et al., 2024) with minimal additional training.

In this survey, we present a comprehensive review of LLM-powered agents for recommender systems. We argue that the core of LLM-powered agents for recommender systems should be systematically analyzed through four key dimensions: Method objective (the fundamental objectives and strategies of different approaches), Agent Architecture (the structural components and their interactions in the recommendation method), Dataset (the comprehensive analysis of recommendation experimental data), and Evaluation methodologies (the metrics and frameworks for recommendation performance assessment). Hence, we first systematically examine how LLM-powered agents address these challenges through three main paradigms: recommender-oriented (e.g., (Wang

| | Our | (Zhu et al., 2024b) | (Zhang et al., 2025) |
|---|---|---|---|
| **Method Objective** | ✓ | × | ✓ |
| **Agent Architecture** | ✓ | ✓ | ✓ |
| **Dataset** | ✓ | ✓ | × |
| **Evaluation** | ✓ | × | × |

Table 1: Comparison with Existing Surveys. ✓ indicates that the corresponding aspect is covered, whereas × indicates that it is not.

et al., 2024b,c)), interaction-oriented (e.g., (Zeng et al., 2024; Friedman et al., 2023)), and simulation-oriented (e.g., (Yoon et al., 2024; Guo et al., 2024)) approaches. Then, we utilize a unified agent architecture consisting of four core modules (Profile (Cai et al., 2024; Zhang et al., 2024c), Memory (Shi et al., 2024; Fang et al., 2024), Planning (Wang et al., 2023b; Shi et al., 2024), and Action (Zhu et al., 2024a; Zhao et al., 2024)) and analyze how existing methods implement these components. Afterwards, we compile comprehensive comparisons of datasets and evaluation methodologies, encompassing both standard recommendation metrics and novel evaluation approaches. Finally, we explore several promising future directions in this field.

**Comparison with existing surveys** Recent surveys have made valuable contributions to understanding LLM agents in information retrieval and recommender systems. Zhu et al. (Zhu et al., 2024b) presented a comprehensive survey on how LLM agents and recommender systems form a symbiotic relationship. Zhang et al. (Zhang et al., 2025) provided an even wider examination of LLM-empowered agents across both recommendation and search tasks. In Table 1, we report a general comparison between the related works. We can find that our survey provide analysis across all these critical aspects, which can enable researchers to develop a more complete understanding of the LLM-powered agents for recommender systems.

(1) We propose a systematic categorization of LLM-powered recommender agents, identifying three fundamental paradigms: recommender-oriented, interaction-oriented, and simulation-oriented approaches. This taxonomy provides a structured framework for understanding.

(2) We utilize an architectural framework for analyzing LLM-powered agent recommender, decomposing them into four essential modules: Profile Construction, Memory Management, Strategic Planning, and Action Execution. Through this, we systematically examine how existing methods integrate and implement these components.

(3) We provide a comprehensive comparative analysis of existing methods, benchmark datasets, and evaluation methodologies, encompassing both traditional recommendation metrics and emerging evaluation approaches specifically designed for LLM-powered agent recommender.

## 2 Background

### 2.1 LLM as Agent

The LLMs as agents is an emerging research direction that has garnered significant attention (Park et al., 2023; Yao et al., 2023; Schick et al., 2023; Shen et al., 2024). By transcending the traditional static prompt-response paradigm, it establishes a dynamic decision-making framework (Patil et al., 2023) capable of systematically decomposing complex tasks into manageable components. A typical LLM-powered agent architecture integrates four fundamental modules (Wang et al., 2024a): (1) the Profile module, which constructs and maintains comprehensive user feature representations; (2) the Memory module, which orchestrates historical interactions and preserves contextual information for systematic experience accumulation; (3) the Planning module, which formulates strategic policies through sophisticated task decomposition and multi-objective optimization; and (4) the Action module, which executes decisions and facilitates environment interaction.

### 2.2 LLM Agents for Recommendation

In LLM-powered agent for recommender systems, we formulate the recommendation process through an agent-centric framework. Let $a \in \mathcal{A}$ denote an agent equipped with a set of functional modules $\mathcal{F} = \mathcal{F}_1, \mathcal{F}_2, ..., \mathcal{F}_K$, where each module $\mathcal{F}_k$ represents a specific capability. The recommendation process for a user $u$ can be formally expressed as:

$$\hat{\mathbf{y}}_u = f(\mathcal{F}_k(X_u)), k = 1 \cdots K \,, \qquad (1)$$

where $X_u \in \mathcal{X}$ represents the input space containing user-specific information (e.g., interaction history, contextual features), and $\hat{\mathbf{y}}_u \in \mathbb{R}^N$ denotes the predicted preference distribution over the item space. The integration function $f : \mathcal{F}_k(X_u) \rightarrow \mathbb{R}^N$ synthesizes module outputs to generate final recommendations. Building upon the previously introduced four functional module (Profile, Memory, Planning, and Action), this formulation provides a flexible framework that can accommodate


Figure 1: Illustration of Different Method Objectives.

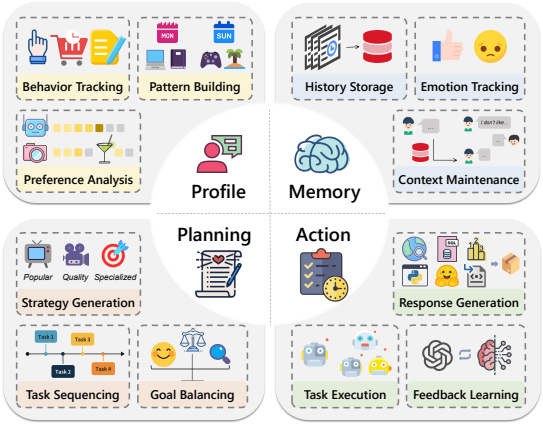


Figure 2: Illustration of Agent Components and Corresponding Functions.

various LLM-powered agent recommendation approaches. These modules operate in a closed-loop framework, where interaction data continuously enriches user profiles and system memory, informing planning strategies that ultimately manifest as personalized recommendations through action execution and feedback collection.

## 3 Methods

In this section, we sort out existing LLM-powered agent recommendation works based on the overall objective of the method and the agent components of different methods.

### 3.1 Method Objective

In Table 2, we classify method objectives of existing methods into three categories: recommender-oriented approaches, interaction-oriented methods, and simulation-oriented methods. The illustrations of categories are shown in Figure 1.

**(1) Recommender-oriented** approaches focus on developing intelligent recommendation equipped with enhanced planning, reasoning, memory, and tool-using capabilities. In these approaches, LLMs leverage users' historical behaviors to generate direct recommendation decisions. For instance, as shown in Figure 1, the model will build and present multi-level content recommendations based on the user's historical preference patterns. This paradigm demonstrates how agents can effectively combine their core capabilities to deliver direct item recommendations. For example, RecMind (Wang et al., 2024b), which develops a unified LLM agent with comprehensive capabilities to generate recommendations directly through LLM outputs.

Despite their significant potential, these approaches face two major challenges: (1) Inconsistency in objectives: the language modeling objective optimized by LLM differs from the recommendation relevance objective, which may result in fluent language but poor recommendation quality; (2) Computational efficiency bottleneck: the high computational cost of directly using LLM to generate recommendation decisions limits the real-time recommendation capability and feasibility of large-scale deployment.

**(2) Interaction-oriented** methods focus on enhancing the natural language interaction capabilities and explainability of recommendation systems through conversational interactions. This type of method uses LLM to conduct human-like conversations and provide recommendation explanations to build a richer user experience. As shown in Figure 1, LLM can track user preferences and naturally express recommendation reasons in conversations, making the recommendation process more transparent and personalized. For example, AutoConcierge (Zeng et al., 2024) uses natural language conversations to understand user needs and collect user preferences, and uses LLM to understand and generate language, ultimately providing explainable personalized restaurant recommendations.

Despite its promising prospects, this approach faces two major challenges: (1) Implicit preference extraction: Accurately identifying and quantifying user preference signals from unstructured conver-

sations is more complex than traditional explicit feedback; (2) Conversation strategy optimization: Achieving a dynamic balance between information acquisition, recommendation quality, and user experience, and determining the optimal decision sequence for when to ask questions, when to recommend, and how to transition naturally remains difficult.

**(3) Simulation-oriented** methods are committed to using LLM to reproduce real user behavior and preference patterns, which focus on using agents to simulate user behaviors and item characteristics in RSs. As shown in Figure 1, the system can simulate the user's decision-making process and generate feedback that conforms to their interest characteristics, providing high-quality simulation data for the recommender systems. For example, UserSimulator proposes (Yoon et al., 2024) an evaluation protocol to assess LLMs as generative user simulators in conversational recommendation through five tasks to measure how closely these simulators can emulate authentic user behaviors.

Although such methods have shown great potential in the evaluation of recommendation systems, they still face the problem of difficulty in modeling complex situations: real user decisions are affected by environmental, emotional, and social factors. These complex situational factors are difficult to fully reproduce in a simulated environment, limiting the simulation system's ability to model users.

### 3.2 Agent Components

The LLM-based agent recommendation architecture consists of four main modules: Profile Module, Memory Module, Planning Module, and Action Module. Figure 2 illustrates the core components of the architecture and corresponding functions.

**(1) Profile Module** is a fundamental component that constructs and maintains dynamic representations of users and items in recommender systems. This module analyzes historical interaction data, identifies user behavior patterns, and forms structured representations to support personalized recommendations. For example, MACRec (Wang et al., 2024c) incorporates a user and item analyst, which play a crucial role in understanding user preferences and item characteristics. AgentCF (Zhang et al., 2024c) constructs natural language-based user profiles to capture dynamic user preferences and item profiles to represent item characteristics and potential adopters' preferences, enabling personalized agent-based collaborative filtering.

Despite the progress, current methods still have key limitations: the representation structure lacks flexibility and is difficult to adapt to emerging user behavior patterns; the temporal modeling capability is insufficient and there is a lack of effective mechanisms to balance long-term preferences with short-term interests; and the profile update strategy is overly simplified and fails to differentiate based on the importance of information.

**(2) Memory Module** serves as a contextual brain that manages and leverages historical interactions and experiences to enhance recommendation quality. This module usually adopts a hierarchical structure design, including different types such as short-term/long-term memory and perceptual memory, forming a multi-level memory storage and retrieval mechanism. The structured memory system enables the system to distinguish and process instant interactive information, accumulate personalized preferences and maintain long-term consistency, providing comprehensive contextual support for decision-making. For example, RecAgent (Wang et al., 2023a) comprises three hierarchical levels: sensory memory, short-term memory, and long-term memory. The sensory memory processes environmental inputs, while short-term memory serves as an intermediate layer that can be transformed into long-term memory through repetitive reinforcement.

However, it also faces the following problems: (1) Retrieval efficiency: The accumulation of historical data leads to a decrease in the efficiency of locating key information in large-scale memory libraries, which is particularly evident in real-time recommendation scenarios; (2) Memory bloat: The lack of an effective forgetting mechanism causes the system to accumulate outdated information, increasing the computational burden and introducing noise, which affects the quality of recommendations.

**(3) Planning Module** outputs intelligent recommendation strategies by designing multi-step action plans that balance immediate user satisfaction with long-term engagement goals. It dynamically formulates recommendation trajectories through careful strategy generation and task sequencing. For example, in video recommendation, the system might construct a strategic plan: "first recommend a popular video to establish user interest, and then gradually introduce niche but high-quality related content, while maintaining the diversity of genres, and ultimately achieve the goal of both satisfying

| Category | Methods | Profile Module | Memory Module | Planning Module | Action Module |
|---|---|---|---|---|---|
| Recommender-oriented Method | RAH (Shu et al., 2024) | × | ✓ | ✓ | ✓ |
| | ToolRec (Zhao et al., 2024) | × | ✓ | × | ✓ |
| | PMS (Thakkar and Yadav, 2024a) | ✓ | × | × | ✓ |
| | DRDT (Wang et al., 2023b) | × | × | ✓ | × |
| | BiLLP (Shi et al., 2024) | × | ✓ | ✓ | ✓ |
| | RecMind (Wang et al., 2024b) | × | ✓ | ✓ | ✓ |
| | MACRec (Wang et al., 2024c) | ✓ | × | ✓ | ✓ |
| Interaction-oriented Method | AutoConcierge (Zeng et al., 2024) | × | ✓ | ✓ | ✓ |
| | MACRS (Fang et al., 2024) | ✓ | ✓ | ✓ | ✓ |
| | RecLLM (Friedman et al., 2023) | ✓ | ✓ | × | ✓ |
| | InteRecAgent (Huang et al., 2023) | ✓ | ✓ | ✓ | ✓ |
| | MAS (Thakkar and Yadav, 2024b) | ✓ | ✓ | ✓ | ✓ |
| | H-MACRS (Nie et al., 2024) | ✓ | ✓ | × | ✓ |
| | Rec4Agentverse (Zhang et al., 2024b) | ✓ | × | ✓ | × |
| Simulation-oriented Method | KGLA (Guo et al., 2024) | ✓ | ✓ | × | ✓ |
| | CSHI (Zhu et al., 2024a) | ✓ | ✓ | × | ✓ |
| | SUBER (Corecco et al., 2024) | ✓ | ✓ | × | × |
| | LUSIM (Zhang et al., 2024d) | ✓ | ✓ | × | × |
| | FLOW (Cai et al., 2024) | ✓ | ✓ | × | ✓ |
| | Agent4Rec (Zhang et al., 2024a) | ✓ | ✓ | × | ✓ |
| | AgentCF (Zhang et al., 2024c) | ✓ | ✓ | × | ✓ |
| | UserSimulator (Yoon et al., 2024) | ✓ | × | × | ✓ |
| | RecAgent (Wang et al., 2023a) | ✓ | ✓ | × | ✓ |

Table 2: Comparative analysis of LLM-powered agent recommendation methods, detailing their methodological orientation (Recommender, Interaction, or Simulation-oriented) and the incorporation of core architectural modules (Profile, Memory, Planning, Action).

user interest and expanding horizons". Through this planning approach, the module optimizes resource allocation and adapts recommendation sequences to achieve both user engagement and item discovery.

BiLLP (Shi et al., 2024) planning mechanism employs a hierarchical structure with two levels: macro-learning (Planner and Reflector LLMs) generates high-level strategic plans and guidelines from experience, while micro-learning (Actor-Critic) translates these plans into specific recommendations. MACRS (Fang et al., 2024) uses a multi-agent planning system where a Planner Agent coordinates three Responder Agents (Ask, Recommend, Chat) through multi-step reasoning. The system adjusts its dialogue strategy through a feedback mechanism, enabling reflective planning based on user interactions.

**(4) Action Module** serves as the execution engine that transforms decisions into concrete recommendations through systematic interaction with various system components. For example, in an e-commerce scenario, when receiving the directive "recommend entry-level camera for new user" from the Planning Module, the Action Module executes a coordinated sequence: analyzing purchase patterns of similar users, querying the product database with specific price and feature constraints, generating targeted recommendations, and capturing user feedback. This execution enables the system to deliver contextually appropriate recommendations while continuously learning from interaction outcomes.

RecAgent (Wang et al., 2023a) orchestrates naturalistic agent interactions within recommender systems and social environments through a unified prompting framework, incorporating six action modalities (encompassing search, browse, click, pagination, chat, and broadcast functionalities). InteRecAgent (Huang et al., 2023) action module integrates three core tools (information querying, item retrieval, and item ranking) while leveraging a Candidate Bus for sequential tool communication, enabling an end-to-end interactive process from user queries to final recommendations.

## 4 Datasets and Evaluations

### 4.1 Datasets

The evaluation of LLM agent-based recommendation systems usually uses two key datasets: tradi-

tional recommendation datasets and conversational recommendation datasets. The former provides large-scale user-item interaction records, while the latter contains multi-round conversation scenarios, which together constitute a comprehensive evaluation framework.

**Traditional Recommendation Dataset** In Table 3, we list several traditional recommendation datasets for evaluating model performance. Several state-of-the-art methods have demonstrated their effectiveness using these datasets.

For instance, the "Books" dataset (10.3M users, 4.4M items) from **Amazon Review data** (McAuley et al., 2015) has been used to evaluate Agent4Rec (Zhang et al., 2024a) and BiLLP (Shi et al., 2024) performance on large-scale tasks, while the "Video Games" dataset (2.8M users, 137.2K items) has validated DRDT (Wang et al., 2023b) and RAH (Shu et al., 2024) capabilities. The "Beauty" dataset (632K users, 112.6K items) has been utilized by IntcRecAgent (Huang et al., 2023) and DRDT (Wang et al., 2023b) to demonstrate their proficiency in recommendation. These diverse applications underscore the datasets' crucial role in advancing LLM-powered agent recommender systems and providing a foundation for evaluating various of algorithms.

The **Steam**, **Lastfm**, **Anime**, and **Yelp** datasets provide diverse domain-specific evaluation scenarios for LLM-powered agent recommender systems. The Steam dataset, introduced by (Kang and McAuley, 2018), contains 3.7M interactions between 334.7K users and 13K gaming items, and has been extensively used by methods such as Agent4Rec (Zhang et al., 2024a), BiLLP (Shi et al., 2024), FLOW (Cai et al., 2024), and InteRecAgent (Huang et al., 2023) to validate their effectiveness in game recommendation. The Lastfm dataset (Cantador et al., 2011), focusing on music recommendation, comprises 73.5K interactions from 1.2K users on 4.6K music items, and has been specifically utilized by FLOW (Cai et al., 2024) to demonstrate its capabilities in the music domain. Additionally, the Yelp dataset, containing 316.3K interactions between 30.4K users and 20.4K items, has been employed by RecMind (Wang et al., 2024b) to evaluate its performance in recommendations. These domain-specific datasets offer unique evaluation opportunities in specialized recommendation contexts.

**Conversational Recommendation Dataset** In addition to the above traditional recommendation datasets, some works (Zhu et al., 2024a) evaluate the model performance on conversational datasets. In Table 3, we list three widely-adopted datasets: **ReDial** (Li et al., 2018), **Reddit** (He et al., 2023), and **OpenDialKG** (Moon et al., 2019). CSHI (Zhu et al., 2024a) employs ReDial (movie domain, including 10006 dialogues) and OpenDialKG (multiple domains, including 13802 dialogues) for performance evaluation. These authentic human-human conversations serve as crucial benchmarks for assessing the model capabilities of LLM-powered agents recommender systems.

However, these datasets face three significant challenges in the context of LLM agent-based recommendation research: (1) Existing benchmarks were primarily designed for traditional recommendation algorithms rather than agent-based systems, making it difficult to comprehensively evaluate unique agent capabilities such as reasoning, memory utilization, and strategic planning. This misalignment limits our ability to accurately assess the true advantages of LLM agent approaches over conventional methods. (2) The inherent need for frequent LLM API calls during both training and evaluation creates significant computational bottlenecks. This has led researchers to adopt sampling strategies—as evidenced by AgentCF's 100-user subsets (Zhang et al., 2024c) and DRDT's 200-user evaluation protocol—which (Wang et al., 2023b), while practical, may compromise the statistical robustness of performance assessments and potentially obscure algorithm behaviors on long-tail distributions. (3) Many benchmark datasets likely overlap with LLM pre-training corpora, creating potential data leakage. This contamination risk is particularly problematic for fair evaluation, as it becomes difficult to distinguish between genuine reasoning capabilities and mere regurgitation of memorized patterns, potentially leading to overly optimistic conclusions about model effectiveness.

### 4.2 Evaluation

In Table 4, we summary the evaluation metrics used by recent representative methods.

**Standard Recommendation Metrics** Most existing methods employ standard recommendation evaluation metrics to assess model performance. The commonly utilized metrics including Normalized Discounted Cumulative Gain (NDCG@K),

| Category | Datasets | Reference | Users | Items | Interactions | Conversations | Turns | Methods |
|---|---|---|---|---|---|---|---|---|
| Traditional Recommendation Dataset | Books | (McAuley et al., 2015) | 10.3M | 4.4M | 29.5M | - | - | Agent4Rec, BiLLP, RAH, SUBER |
| | CDs and Vinyl | | 1.8M | 701.7K | 4.8M | - | - | AgentCF, KGLA, ToolRec |
| | Video Games | | 2.8M | 137.2K | 4.6M | - | - | DRDT, RAH, LUSIM |
| | Beauty | | 632.0K | 112.6K | 701.5K | - | - | InteRecAgent, DRDT, RecMind |
| | Clothing | | 22.6M | 7.2M | 66.0M | - | - | DRDT |
| | Movies | | 6.5M | 747.8K | 17.3M | - | - | RAH, LUSIM |
| | Office Products | | 7.6M | 710.4K | 12.8M | - | - | AgentCF |
| | Music | | 101.0K | 70.5K | 130.4K | - | - | LUSIM |
| | Movielens-100K | (Harper and Konstan, 2015) | 0.9K | 1.6K | 100K | - | - | FLOW, MACRS, SUBER |
| | Movielens-1M | | 6K | 3.7K | 1.0M | - | - | Agent4Rec, RecAgent, DRDT, MACRS, ToolRec |
| | Movielens-10M | | 69.9K | 10.6K | 10M | - | - | InteRecAgent |
| | Movielens-20M | | 138.5K | 27.3K | 20M | - | - | MACRS, UserSimulator |
| | Steam | (Kang and McAuley, 2018) | 334.7K | 13K | 3.7M | - | - | Agent4Rec, BiLLP, FLOW, InteRecAgent |
| | Lastfm | (Cantador et al., 2011) | 1.2K | 4.6K | 73.5K | - | - | FLOW |
| | Yelp | https://www.yelp.com/dataset | 30.4K | 20.4K | 316.3K | - | - | RecMind, ToolRec, LUSIM |
| | Anime | https://www.kaggle.com/datasets | 73.5K | 12.2K | 1.05M | - | - | LUSIM |
| Conversational Recommendation Dataset | ReDial | (Li et al., 2018) | 0.9K | 51.6K | - | 10K | - | UserSimulator, CSHI |
| | Reddit | (He et al., 2023) | 36.2K | 51.2K | - | 634.4K | 1.6M | UserSimulator |
| | OpenDialKG | (Moon et al., 2019) | - | - | - | 15.6K | 91.2K | CSHI |

Table 3: Summary of Used Experimental Datasets.

| Category | Metrics | Methods |
|---|---|---|
| Standard Recommendation | NDCG@K, Recall@K, HR@K, Hit@K, MRR, Acc, F1-Score, MAP | DRDT, RecMind, InteRecAgent, RAH, MACRS, PMS, Agent4Rec, AgentCF, KGLA, FLOW, CSHI, ToolRec, SUBER |
| | RMSE, MAE, MSE | RecMind |
| Language Generation Quality | BLEU, ROUGE | RecMind, PMS |
| Reinforcement Learning | Rewards | LUSIM, BiLLP, SUBER |
| Conversational Efficiency | Average Turn (AT), Success Rate (SR) | InteRecAgent, MACRS, CSHI |
| Custom Indicators | Proactivity, Economy, Explainability, Correctness, Consistency, Efficiency | AutoConcierge |
| | Simulated user behaviors believability, Agent memory believability | RecAgent |

Table 4: Summary of Used Evaluation Metrics.

Recall@K and Hit Ratio@K (HR@K), etc. For instance, AgentCF (Zhang et al., 2024c) evaluates its performance using NDCG@K and Recall@K on the MovieLens-1M dataset. Similarly, DRDT (Wang et al., 2023b) conducts comprehensive evaluations using Recall@10,20 and NDCG@10,20 across multiple datasets including ML-1M, Games, and Luxury datasets. Hit Ratio@K (HR@K) is another crucial metric for evaluating recommendation performance. RecMind (Wang et al., 2024b) employ that for evaluating the recommendation tasks on Amazon Reviews (Beauty) and Yelp datasets.

**Language Generation Quality** Some methods (Wang et al., 2024b) consider the evaluation of language generation quality (e.g., recommendation explanation generation, review summarization), which primarily rely on BLEU and ROUGE metrics. BLEU measures the precision of generated text against references, while ROUGE evaluates recall-based similarity, enabling comprehensive assessment of language generation capabilities in recommendation scenarios. PMS (Thakkar and Yadav, 2024a) utilizes the ROUGE to evaluate the quality of its generated textual recommendations.

**Reinforcement Learning Metrics** In evaluating LLM-powered agent recommender systems for long-term engagement, BiLLP (Shi et al., 2024) employs three key metrics adopted from reinforcement learning: trajectory length, average single-round reward, and cumulative trajectory reward. Similarly, LUSIM (Zhang et al., 2024d) uses the total reward to reflect the overall user engagement during the entire interaction process, and the average reward to represent the average quality of a single recommendation. These metrics are to eval-

uate both immediate recommendation quality and long-term engagement effectiveness.

**Conversational Efficiency Metrics** Recent research has introduced more comprehensive metrics to evaluate the efficiency of conversational interactions in recommender systems. For instance, MACRS (Fang et al., 2024) employs key interaction-focused metrics such as Success Rate (proportion of successful recommendations) and Average Turn (AT) (number of interaction rounds needed to reach a recommendation) per session. These metrics assess how effectively the system can understand user preferences and deliver accurate recommendations while minimizing the number of interaction turns.

**Custom Indicators** Beyond conventional metrics, some methods (Yoon et al., 2024) propose customized evaluation frameworks. AutoConcierge (Zeng et al., 2024) presents six evaluation metrics for task-driven conversational agents: proactivity, economy, explainability, correctness, consistency, and efficiency. RecAgent (Wang et al., 2023a) proposes simulated user behaviors believability and Agent memory believability, to assess the credibility of LLM-simulated user interactions and memory mechanism effectiveness. These metrics assess system engagement, dialogue efficiency, answer interpretability, response accuracy, requirement fulfillment, and response time, respectively.

This diversity of evaluation methodologies reflects the complexity of LLM-powered agent recommenders but also introduces significant challenges. The lack of standardization across studies makes direct comparison between different approaches difficult. Many custom metrics remain unvalidated across diverse datasets and use cases, raising questions about their generalizability. Furthermore, existing evaluation frameworks often assess individual aspects of performance in isolation, failing to capture the inherent trade-offs between recommendation accuracy, language quality, interaction efficiency, and user experience.

## 5 Related Research Fields

**LLM-powered Recommender Systems** In recent years, recommender systems based on Large Language Models (LLMs) have attracted widespread attention. Such systems make full use of the powerful language understanding and generation capabilities of LLMs, bringing a new paradigm to traditional recommender systems. Most existing methods are primarily designed for rating prediction (Bao et al., 2023) and sequential recommendation (Hou et al., 2024; Zheng et al., 2024). CoLLM (Zhang et al., 2023) captures and maps the collaborative information through external traditional models, forming collaborative embeddings used by LLMs. LlamaRec (Yue et al., 2023) finetunes Llama-2-7b for list-wise ranking of the preselected items. However, these methods would face significant limitations: the inability to simulate authentic user behaviors for enhanced personalization, the lack of effective memory mechanisms for long-term context awareness, and the rigid pipeline structure that prevents flexible task decomposition and seamless integration with external tools.

## 6 Future Directions

**Refinement of Evaluation Framework** There is a notable absence of unified and comprehensive evaluation standards for accurately measuring dialogue quality and recommendation effectiveness. Future research necessitates the establishment of robust evaluation frameworks, development of novel performance metrics, and consideration of privacy and security concerns in practical applications.

**Security Recommender System** (Ning et al., 2024) reveals the vulnerability of LLM-empowered recommender systems to adversarial attacks. In future, the researchers could develop robust adversarial detection methods, investigate multi-agent defensive architectures, and integrating domain-specific security knowledge into defense.

## 7 Conclusion

Recent, the integration of LLM-powered agents into recommender systems has emerged as a significant advancement. In this survey, we established a systematic taxonomy categorizing existing approaches into three paradigms: recommender-oriented, interaction-oriented, and simulation-oriented. We analyzed these methods through a comprehensive four-module architectural framework and critically examined the datasets and evaluation methodologies employed across the literature. Finally, we identify two promising directions for future exploration.

## 8 Limitation

First, our classification framework, while effective for current approaches, may require extension as

novel hybrid methods continue to emerge at the intersection of our proposed paradigms. Second, due to the limited adoption of LLM-powered recommendation agents in industrial settings thus far, our survey does not extensively explore commercial implementations and their unique challenges.

## References


Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Keqin Bao, Jizhi Zhang, Yang Zhang, Wenjie Wang, Fuli Feng, and Xiangnan He. 2023. Tallrec: An effective and efficient tuning framework to align large language model with recommendation. In *Recsys*, pages 1007–1014.

Shihao Cai, Jizhi Zhang, Keqin Bao, Chongming Gao, and Fuli Feng. 2024. Flow: A feedback loop framework for simultaneously enhancing recommendation and user agents. *arXiv preprint arXiv:2410.20027*.

Iván Cantador, Peter Brusilovsky, and Tsvi Kuflik. 2011. Second workshop on information heterogeneity and fusion in recommender systems (hetrec2011). In *Recsys*, pages 387–388.

Nathan Corecco, Giorgio Piatti, Luca A Lanzendörfer, Flint Xiaofeng Fan, and Roger Wattenhofer. 2024. An llm-based recommender system environment. *arXiv preprint arXiv:2406.01631*.

Jiabao Fang, Shen Gao, Pengjie Ren, Xiuying Chen, Suzan Verberne, and Zhaochun Ren. 2024. A multi-agent conversational recommender system. *arXiv preprint arXiv:2402.01135*.

Luke Friedman, Sameer Ahuja, David Allen, Zhenning Tan, Hakim Sidahmed, Changbo Long, Jun Xie, Gabriel Schubiner, Ajay Patel, and 1 others. 2023. Leveraging large language models in conversational recommender systems. *arXiv preprint arXiv:2305.07961*.

Taicheng Guo, Chaochun Liu, Hai Wang, Varun Mannam, Fang Wang, Xin Chen, Xiangliang Zhang, and Chandan K Reddy. 2024. Knowledge graph enhanced language agents for recommendation. *arXiv preprint arXiv:2410.19627*.

F Maxwell Harper and Joseph A Konstan. 2015. The movielens datasets: History and context. *ACM TIIS*, 5(4):1–19.

Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *The WebConf*, pages 173–182.

Zhankui He, Zhouhang Xie, Rahul Jha, Harald Steck, Dawen Liang, Yesu Feng, Bodhisattwa Prasad Majumder, Nathan Kallus, and Julian McAuley. 2023. Large language models as zero-shot conversational recommenders. In *CIKM*, pages 720–730.

Yupeng Hou, Junjie Zhang, Zihan Lin, Hongyu Lu, Ruobing Xie, Julian McAuley, and Wayne Xin Zhao. 2024. Large language models are zero-shot rankers for recommender systems. In *ECIR*, pages 364–381.

Xu Huang, Jianxun Lian, Yuxuan Lei, Jing Yao, Defu Lian, and Xing Xie. 2023. Recommender ai agent: Integrating large language models for interactive recommendations. *arXiv preprint arXiv:2308.16505*.

Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *ICDM*, pages 197–206. IEEE.

Raymond Li, Samira Ebrahimi Kahou, Hannes Schulz, Vincent Michalski, Laurent Charlin, and Chris Pal. 2018. Towards deep conversational recommendations. *NuerIPS*, 31.

Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. 2015. Image-based recommendations on styles and substitutes. In *SIGIR*, pages 43–52.

Seungwhan Moon, Pararth Shah, Anuj Kumar, and Rajen Subba. 2019. Opendialkg: Explainable conversational reasoning with attention-based walks over knowledge graphs. In *ACL*, pages 845–854.

Guangtao Nie, Rong Zhi, Xiaofan Yan, Yufan Du, Xiangyang Zhang, Jianwei Chen, Mi Zhou, Hongshen Chen, Tianhao Li, Ziguang Cheng, and 1 others. 2024. A hybrid multi-agent conversational recommender system with llm and search engine in e-commerce. In *Recsys*, pages 745–747.

Liang-bo Ning, Shijie Wang, Wenqi Fan, Qing Li, Xin Xu, Hao Chen, and Feiran Huang. 2024. Cheatagent: Attacking llm-empowered recommender systems via llm agent. In *KDD*, pages 2284–2295.

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 *AASUIST*, pages 1–22.

Shishir G Patil, Tianjun Zhang, Xin Wang, and Joseph E Gonzalez. 2023. Gorilla: Large language model connected with massive apis. *arXiv preprint arXiv:2305.15334*.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools. In *NuerIPS*, volume 36.

Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. 2024. Hugginggpt: Solving ai tasks with chatgpt and its friends in hugging face. 36.

Wentao Shi, Xiangnan He, Yang Zhang, Chongming Gao, Xinyue Li, Jizhi Zhang, Qifan Wang, and Fuli Feng. 2024. Large language models are learnable planners for long-term recommendation. In *SIGIR*, pages 1893–1903.

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 TCSS*.

Param Thakkar and Anushka Yadav. 2024a. Personalized recommendation systems using multimodal, autonomous, multi agent systems. *arXiv preprint arXiv:2410.19855*.

Param Thakkar and Anushka Yadav. 2024b. Personalized recommendation systems using multimodal, autonomous, multi agent systems. *arXiv preprint arXiv:2410.19855*.

Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, and 1 others. 2024a. 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, and 1 others. 2023a. User behavior simulation with large language model based agents. *arXiv preprint arXiv:2306.02552*.

Yancheng Wang, Ziyan Jiang, Zheng Chen, Fan Yang, Yingxue Zhou, Eunah Cho, Xing Fan, Yanbin Lu, Xiaojiang Huang, and Yingzhen Yang. 2024b. Recmind: Large language model powered agent for recommendation. In *Findings of NAACL*, pages 4351–4364.

Yu Wang, Zhiwei Liu, Jianguo Zhang, Weiran Yao, Shelby Heinecke, and Philip S Yu. 2023b. Drdt: Dynamic reflection with divergent thinking for llm-based sequential recommendation. *arXiv preprint arXiv:2312.11336*.

Zhefan Wang, Yuanqing Yu, Wendi Zheng, Weizhi Ma, and Min Zhang. 2024c. Macrec: A multi-agent collaboration framework for recommendation. In *SIGIR*, pages 2760–2764.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models. In *ICLR*.

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*.

Zhenrui Yue, Sara Rabhi, Gabriel de Souza Pereira Moreira, Dong Wang, and Even Oldridge. 2023. Llamarec: Two-stage recommendation using large language models for ranking. *arXiv preprint arXiv:2311.02089*.

Yankai Zeng, Abhiramon Rajasekharan, Parth Padalkar, Kinjal Basu, Joaquín Arias, and Gopal Gupta. 2024. Automated interactive domain-specific conversational agents that understand human dialogs. In *ISPADL*, pages 204–222.

An Zhang, Yuxin Chen, Leheng Sheng, Xiang Wang, and Tat-Seng Chua. 2024a. On generative agents in recommendation. In *SIGIR*, pages 1807–1817.

Jizhi Zhang, Keqin Bao, Wenjie Wang, Yang Zhang, Wentao Shi, Wanhong Xu, Fuli Feng, and Tat-Seng Chua. 2024b. Prospect personalized recommendation on large language model-based agent platform. *arXiv preprint arXiv:2402.18240*.

Junjie Zhang, Yupeng Hou, Ruobing Xie, Wenqi Sun, Julian McAuley, Wayne Xin Zhao, Leyu Lin, and Ji-Rong Wen. 2024c. Agentcf: Collaborative learning with autonomous language agents for recommender systems. In *The WebConf*, pages 3679–3689.

Yang Zhang, Fuli Feng, Jizhi Zhang, Keqin Bao, Qifan Wang, and Xiangnan He. 2023. Collm: Integrating collaborative embeddings into large language models for recommendation. *arXiv preprint arXiv:2310.19488*.

Yu Zhang, Shutong Qiao, Jiaqi Zhang, Tzu-Heng Lin, Chen Gao, and Yong Li. 2025. A survey of large language model empowered agents for recommendation and search: Towards next-generation information retrieval. *arXiv preprint arXiv:2503.05659*.

Zijian Zhang, Shuchang Liu, Ziru Liu, Rui Zhong, Qingpeng Cai, Xiangyu Zhao, Chunxu Zhang, Qidong Liu, and Peng Jiang. 2024d. Llm-powered user simulator for recommender system. *arXiv preprint arXiv:2412.16984*.

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 *SIGIR*, pages 1796–1806.

Bowen Zheng, Yupeng Hou, Hongyu Lu, Yu Chen, Wayne Xin Zhao, Ming Chen, and Ji-Rong Wen. 2024. Adapting large language models by integrating collaborative semantics for recommendation. In *ICDE*, pages 1435–1448. IEEE.

Lixi Zhu, Xiaowen Huang, and Jitao Sang. 2024a. A llm-based controllable, scalable, human-involved user simulator framework for conversational recommender systems. *arXiv preprint arXiv:2405.08035*.

Xi Zhu, Yu Wang, Hang Gao, Wujiang Xu, Chen Wang, Zhiwei Liu, Kun Wang, Mingyu Jin, Linsey Pang, Qingsong Wen, and 1 others. 2024b. Recommender systems meet large language model agents: A survey. *SSRN 5062105*.