Temporal-Aware User Behaviour Simulation with Large Language Models for Recommender Systems

Xinye Wanyan
xinye.wanyan@student.rmit.edu.au
RMIT University
Melbourne, VIC, Australia

Danula Hettiachchi
danula.hettiachchi@rmit.edu.au
RMIT University
Melbourne, VIC, Australia

Chenglong Ma
chenglong.ma@rmit.edu.au
RMIT University
Melbourne, VIC, Australia

Ziqi Xu
ziqi.xu@rmit.edu.au
RMIT University
Melbourne, VIC, Australia

Jeffrey Chan
jeffrey.chan@rmit.edu.au
RMIT University
Melbourne, VIC, Australia

Abstract

Large Language Models (LLMs) demonstrate human-like capabilities in language understanding, reasoning, and generation, driving interest in using LLM-based agents to simulate human feedback in recommender systems. However, most existing approaches rely on static user profiling, neglecting the temporal and dynamic nature of user interests. This limitation stems from a disconnect between language modelling and behaviour modelling, which constrains the capacity of agents to represent sequential patterns. To address this challenge, we propose a Dynamic Temporal-aware Agent-based simulator for Recommender Systems, DyTA4Rec, which enables agents to model and utilise evolving user behaviour based on historical interactions. DyTA4Rec features a dynamic updater for real-time profile refinement, temporal-enhanced prompting for sequential context, and self-adaptive aggregation for coherent feedback. Experimental results at group and individual levels show that DyTA4Rec significantly improves the alignment between simulated and actual user behaviour by modelling dynamic characteristics and enhancing temporal awareness in LLM-based agents.[1]

CCS Concepts

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

Keywords

Recommender System, Large Language Model, Generative Agents

[1] The source code can be found at https://github.com/WennyXY/DyTA4Rec.

1 Introduction

Modern recommender systems (RSs) demand real-time interactions to enable effective evaluation and iterative system improvement [7, 19, 21]. To meet this need, recent studies have explored the use of large language models (LLMs), which exhibit human-level capabilities in general-purpose understanding, reasoning, and decision making [13, 26], as credible proxies for interacting with RSs [3, 12, 16, 20, 22, 25, 28]. LLM-based agents support dynamic and nuanced user profiling and enable real-time, diverse evaluation with minimal task-specific training. Specifically, LLMs function as the central reasoning module, equipped with components for user profiling, memory management, and action selection, to simulate user behaviour in RSs [23].

Building upon this general LLM-based agent framework, various agent architectures are developed and customised for specific application scenarios to enhance the fidelity of user behaviour simulation in RSs. Agent4Rec [20] performs statistical analysis on user interaction patterns and incorporates the extracted insights into user profiles, thereby improving the realism of simulated behaviours. Its action module also accounts for both user preferences and emotion-driven responses. Extending beyond individual user modelling, AgentCF [22] introduces user and item agents that engage in interactive processes, enabling them to collaboratively learn and adapt through mutual interaction. In contrast, RecAgent [16] focuses on modelling the influence of social context by integrating external relationships and social activities into the simulation of user responses to recommendations.

Existing frameworks primarily focus on inferring user preferences and characteristics from static historical data, overlooking the dynamic evolution of user interests and the temporal patterns underlying user behaviour. Prior research has shown that incorporating temporal dynamics into user modelling can significantly improve the accuracy and effectiveness of personalisation strategies [1]. However, recent work by Hou et al. [10] shows that LLMs have notable limitations in modelling sequential patterns, which can hinder their ability to detect temporal shifts in user interests or behavioural dynamics without dedicated architectural or training interventions. The findings highlight the need for an enhanced agent design that explicitly accounts for temporal information and models user behaviour as a dynamic and sequential process.

To address the challenges, we propose DyTA4Rec, a novel LLM-based simulator for recommendation systems that models both

static and dynamic features and captures temporal behaviour patterns for informed decision-making. Our main contributions are:

- DyTA4Rec presents a temporal-aware simulation framework that improves the behavioural fidelity and temporal consistency of agent interactions through sequential reasoning.
- The framework integrates a dynamic profile updater to capture real-time short-term user modelling and a self-adaptive aggregator for combining multi-dimensional behavioural signals.
- Experiments on the MovieLens-1M dataset show that DyTA4Rec aligns agent behaviour with real user patterns in terms of accuracy and believability at both group and individual levels.

2 DyTA4Rec

The architecture of DyTA4Rec is shown in Fig. 1. It comprises four main modules: (1) dynamic profile updater, (2) temporal pattern extractor, (3) self-adaptive aggregator, and (4) memory module. The following subsections provide a detailed explanation.

2.1 Dynamic Profile Updater

The user profile integrates both static, long-term attributes (e.g., demographic information, preferences, and behavioural patterns) and dynamic, short-term features that reflect recent interactions and evolving interests. Long-term preferences typically represent stable user interests, whereas short-term preferences are more specific and context-dependent, capturing temporary intent [15]. This hybrid representation enables the agent to form a comprehensive and temporal-aware understanding of user behaviour, thereby supporting more accurate and personalised simulation.

Long-term features are initialised based on the MovieLens-1M (ML-1M) dataset [8]. To enrich user profiles beyond basic metadata, we employ LLMs to extract personalities and item-level preferences from each user's historical interactions. In addition, we compute statistical patterns to capture rating tendencies (e.g., distributions over popular or highly rated items) and viewing behaviours (e.g., preferences for popular or highly rated content). These patterns guide agent's decision-making during simulation. The modular design of the dynamic updater enables flexible incorporation of dataset-specific features, enhancing its adaptability to a wide range of application scenarios. Specifically, we leverage LLMs to extract and summarise users' recent behaviours and evolving interests from their most recent interaction history, serving as short-term features updated every n rounds.

2.2 Temporal Pattern Extractor

We propose a Temporal Pattern Extractor (TPE) to capture temporal dynamics and sequential patterns in user behaviour by analysing most recent interaction histories.

The process of TPE can be formulated as:

$$\text{TPE}(H, C) = f^{act \cdot}(f^{cluster \cdot}(H)) + f^{seq \cdot}(\text{ICL}(H)) \tag{1}$$

where f denotes utilising LLMs for clustering, prediction, and decision-making. The function $f^{act \cdot}$ represents the action mechanism of the agent, which may involve ranking candidates, selecting the next interaction, or performing other task-specific operations.

For each agent, we define the most recent n interactions as $H = \{h_1, h_2, ..., h_n\}$, where each $h_t = (i_t, r_t, f_t)$ represents an interaction involving item i_t, a rating r_t, and a generated feeling f_t. During simulation, the recommender system presents content to the agent in a page-by-page manner, with each page containing a candidate set $C = \{c_1, c_2, ..., c_m\}$. Temporal information of H is obtained from clustering $f^{cluster \cdot}$ and sequential prediction $f^{seq \cdot}$. $f^{cluster \cdot}$ identifies recurring behaviour patterns and detects potential shifts in user interests. $f^{seq \cdot}$ captures ordered dependencies in H to infer the most likely next interaction. This combination enables agents to make time-consistent decisions that align with recent interaction trajectories.

Additionally, we adopt a two-step prompting approach for clustering pattern extraction to more effectively leverage the reasoning capabilities of LLMs. Specifically, the LLM performs an independent analysis, identifying key patterns or insights. Then, this intermediate analysis is provided as contextual information for the LLM to execute the specific downstream task, such as prediction or ranking. This structured process enhances the quality of reasoning and improves task performance by separating understanding from decision-making (as shown in Figure 2a).

We leverage LLMs through in-context learning (ICL) to enable such temporal reasoning without additional training. LLMs have demonstrated the ability to adapt to new tasks by conditioning on a few prompt-based examples, eliminating the need for model retraining [6, 18]. In our framework, both temporal clustering and sequential prediction functions are instantiated directly from the interaction history using LLMs.

2.3 Self-Adaptive Aggregator

We design an aggregation approach self-adaptive aggregator (SAA) for integrating complementary behavioural information (i.e., long- and short-term user traits and interests, sequential patterns, and temporal clustering patterns and balancing their relative importance. We implement and compare Borda Count (BC) and Reciprocal Rank Fusion (RRF) [5] as aggregation methods. Compared to averaging or majority voting with fixed weights, our method assigns personalised weights based on whether the interaction history contains consistent temporal or sequential patterns. This enables more context-sensitive information fusion and decision generation.

User profile and memory information serve as the foundation of decision-making, hence the corresponding weight W_l is set to 1. For temporal reasoning, LLMs are employed to determine whether the interaction history exhibits clear sequential patterns or temporal clusters. A weight W_s is allocated to the sequential prediction output when sequential dependencies are observed, while a weight W_c is applied to the clustering-based outcome in the presence of discernible temporal clustering patterns.

The aggregation process is defined as:

$$\text{Aggregator}(R) = W_l \times r_l + W_s \times r_s + W_c \times r_c \tag{2}$$

where $R = \{r_l, r_s, r_c\}$ denotes the results generated by different modules. The term $r_l = w_i \times r_l^1$ represents the output of the user profile and memory module, r_s represents the output of sequential prediction, and r_c represents the output of clustering analysis.
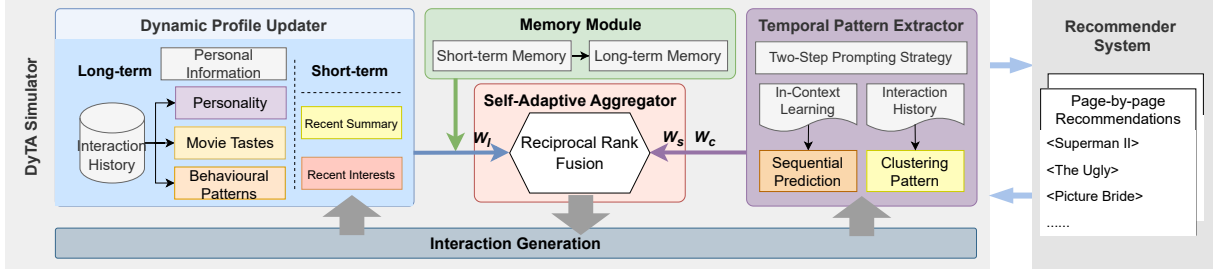
Figure 1: An overview of the proposed DyTA4Rec, comprising an LLM-based generative agent and a recommender system. The recommender system is designed as a flexible component that can be reconfigured to support different objectives. During simulation, the agent replicates real user behaviour, enabling realistic interactions with the recommender system.

2.4 Memory Module

Memory constitutes the core of the human decision-making process [17, 24]. Inspired by previous work [16, 20], we design a specialised memory module for LLM-based simulation in recommendation. Our memory module consists of two components: short-term memory and long-term memory. Specifically, the short-term memory captures the recent interaction history directly and tracks the emotional feelings, facilitating the system to adapt to the dynamic environment. In contrast, the long-term memory stores high-level and informative patterns that remain consistent over time, which are extracted and summarised from short-term memory by LLM.

3 Experiments

3.1 Experimental Setup

We conduct experiments on the benchmark recommendation dataset ML-1M [8]. We adopt the GPT-4o-mini, due to budget constraints, for simulations by default. To reduce output variability from the language model, the temperature is set to 0.1 and the top-p value is set to 0.9. All reported results are averaged over three independent runs to ensure reliability and consistency.

We consider four baseline models: (1) Random, a random ranking strategy; (2) RecAgent [16], an LLM-based agent simulation framework; (3) BM25 [14], a text-based retrieval algorithm that ranks items based on their textual similarity to user preferences; and (4) UniSRec [9], a pre-trained sequential recommendation model.

The evaluation metrics include the normalised discounted cumulative gain (nDCG@K), where $K \in \{5, 10\}$, and the top-N hit rate (HR@3). Following previous studies [9, 22, 27], we adopt the leave-one-out evaluation strategy, where the last item in each historical interaction sequence is treated as the ground-truth item.

We construct ranking candidates by including one ground-truth item and nine negative samples. Empirical research [4, 10] identifies position bias in RSs, where users are more likely to select items near the top of the recommendation list. To mitigate this bias, we apply a simple Direct Prompting strategy to explicitly instruct that candidate positions carry no significance.

3.2 Main Results

We assess the **ranking ability** of each agent. Main results are presented in Table 1. DyTA4Rec consistently outperforms the baseline models in next-item prediction. The traditional retrieval-based

Table 1: Performance comparison on the ML-1M dataset. The first block reports baseline results. A tailored version of RecAgent is evaluated on ML-1M since the social relationship information of dataset is not provided. The second block presents ablation studies of DyTA4Rec by varying behavioural inputs and removing the SAA. The third block shows the performance of the DyTA4Rec using different aggregation strategies. The best result in each row is highlighted in bold, and the second-best result is <u>underlined</u>.

Model	nDCG@5	nDCG@10	HR@3
Random	0.326	0.509	0.240
BM25	0.260	0.428	0.242
UniSRec	0.340	0.484	0.365
RecAgent	0.431	0.571	0.391
Long-term	0.434	0.568	0.360
Long- & Short-term	0.467	0.599	0.400
Sequential	0.466	0.584	0.418
Clustering	0.462	0.593	0.409
Sequential+Long-term	0.494	0.622	0.458
DyTA4Rec w/o SAA (BC)	0.489	0.602	0.427
DyTA4Rec w/o SAA (RRF)	0.457	0.585	0.396
DyTA4Rec (BC)	<u>0.514</u>	<u>0.632</u>	<u>0.467</u>
DyTA4Rec (RRF)	**0.551**	**0.639**	**0.480**

method BM25 performs poorly across all metrics, reflecting the limitations in capturing user intents. UniSRec performs better than BM25 due to its temporal modelling ability. RecAgent shows further improvement by incorporating generative reasoning. Notably, DyTA4Rec with RRF achieves the best results, with 0.551 in nDCG@5, 0.639 in nDCG@10, and 0.480 in HR@3, demonstrating the effectiveness of our temporal-aware and adaptive aggregation design.

Beyond ranking performance, we further evaluate the **behavioural alignment** between simulated agents and real users at a macro level. Following the setup in [20], we assess rating distribution consistency on ML-1M. Figure 2b compares the ground-truth rating distribution with those generated by the baseline simulator [16] and our proposed simulator. DyTA4Rec outperforms the baseline models in aligning with actual ratings, showing stronger consistency between simulated agents and real human behaviour at the group level.
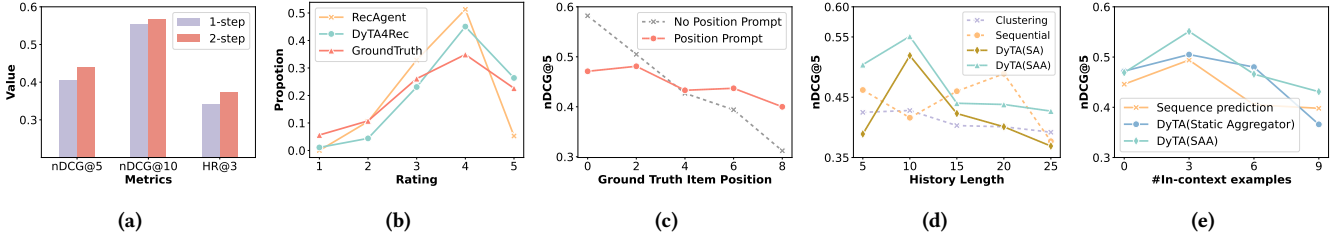
Figure 2: Performance comparisons across various settings: (a) ranking performance under single-step and two-step prompting strategies; (b) rating distribution alignment between ground truth and simulated data (RecAgent vs. DyTA4Rec); (c) impact of ground-truth item position on simulation outcomes; (d) effect of interaction history length; (e) effect of ICL example count.

Our study also investigate the impact of position bias within simulations. Figure 2c shows that the performance of LLM-based agents is highly sensitive to the recommended items position, as the selection is easily influenced by items order. Without explicit prompts indicating that item positions are random or irrelevant, agents often overvalue top-ranked options. This reflects real-world position bias where top-listed items receive disproportionate attention and engagement [2]. To address this, we design an explicit prompt to inform the agent of the randomness of item positions, effectively reducing position bias in its decisions.

3.3 Ablation Study

We conduct an ablation study to examine the contribution of different components in DyTA4Rec (second block, Table 1). The user profile integrates both long-term and short-term features. Long-term features are initialised from real-world data and remain fixed throughout the simulation, while short-term features are dynamically generated and updated by the profile updater based on recent interactions. To assess the effect of dynamic profiling, we compare two configurations: using only long-term features (Long-term) and combining long-term with short-term features (Long & Short-term).

We further study the impact of each temporal reasoning strategy. The Sequential and Clustering variants enable only the sequential prediction or temporal clustering module, respectively, while Sequential + Long-term combines sequential reasoning with long-term profiling. In addition, we disable the SAA and apply static aggregation (SA) strategies, namely BC and RRF, to examine the importance of adaptive weighting.

Table 1 shows that incorporating short-term features yields clear improvements over using long-term profiling alone (e.g., nDCG@5 increases from 0.434 to 0.467), confirming the value of dynamic features tracking. Similarly, enabling sequential or clustering reasoning individually improves performance, and combining sequential signals with long-term profiling leads to further gains. Removing the SAA also results in performance degradation, demonstrating the effectiveness of adaptive strategy integration. These results highlight the importance of both temporal reasoning and dynamic profiling in achieving high-quality simulation.

3.4 Parameter Analysis

We investigate the impact of two key parameters in DyTA4Rec: the length of the interaction history and the number of in-context examples used for sequential behaviour modelling. These parameters influence how effectively the agent can capture temporal dependencies and recent user intents during the simulation.

First, we examine the impact of interaction history length on performance. As shown in Figure 2e, the performance of all three agents tends to decline as the length of the interaction history increases. This may be attributed to two aspects: (1) limited ability of LLMs to extract and reason over long behavioural sequences although they can process long textual inputs [10, 11]; and (2) long sequence may distract LLMs' attention from most recent interactions, which are often more informative for decision-making and more accurately reflect the user's current intent.

Second, we study the influence of ICL by varying the number of in-context examples in the prompt. The interaction history length is set to 10 and the most recent k items are selected as ICL examples, with $k \in \{0, 3, 6, 9\}$. As shown in Figure 2e, using 3 examples achieves the best overall performance. This suggests that a moderate number of examples is sufficient to enhance temporal reasoning, while excessive prompts may introduce noises.

4 Conclusion and Future Work

In this paper, we present DyTA4Rec, a novel LLM-based simulator that enhances the temporal awareness of agents in recommendation scenarios. By integrating a dynamic profile updater, a temporal pattern extractor, and a self-adaptive aggregator, DyTA4Rec captures both static and evolving user behaviour patterns for more realistic simulations. Experiments demonstrate improved alignment with real user preferences at both individual and group levels. However, the effectiveness of LLM-based agents remains sensitive to prompt design and prone to overconfident outputs. Future work will explore integrating an extra model to guide the output generation, aiming to mitigate the prompt sensitivity. We also plan to generalise our simulation framework on different benchmark datasets across diverse domains to assess its generalisability and robustness.

Acknowledgments

GenAI Usage Disclosure

This research complies with the CIKM 2025 GenAI usage policy. The authors disclose the following use of Generative AI (GenAI) tools during the research process:

- **Writing:** ChatGPT (GPT-4, OpenAI) was used to assist in proofreading, rephrasing technical sentences, and improving the clarity of the manuscript. All substantive content, including ideas, methods, results, and analysis, was written and verified by the authors.
- **Code:** No GenAI tools were used to generate any code in this research. All code was developed by the authors.
- **Data:** No GenAI tools were used to generate or augment the data used in this research. All datasets were obtained from publicly available sources as described in the paper.
- **Experiments and Analysis:** No GenAI tools were used to generate experimental results or statistical analyses.

The authors confirm that all intellectual contributions are original and that the use of GenAI tools did not compromise the scientific integrity or originality of the work.

References

[1] Veronika Bogina, Tsvi Kuflik, Dietmar Jannach, Maria Bielikova, Michal Kompan, and Christoph Trattner. 2023. Considering temporal aspects in recommender systems: a survey. *User Modeling and User-Adapted Interaction* 33, 1 (2023), 81–119.

[2] Jiawei Chen, Hande Dong, Xiang Wang, Fuli Feng, Meng Wang, and Xiangnan He. 2023. Bias and Debias in Recommender System: A Survey and Future Directions. *ACM Transactions on Management Information Systems* 41, 3 (2023), 67:1–67:39.

[3] Luyu Chen, Quanyu Dai, Zeyu Zhang, Xueyang Feng, Mingyu Zhang, Pengcheng Tang, Xu Chen, Yue Zhu, and Zhenhua Dong. 2025. RecUserSim: A Realistic and Diverse User Simulator for Evaluating Conversational Recommender Systems. In *Companion Proceedings of the ACM on Web Conference 2025, WWW*. 133–142.

[4] Andrew Collins, Dominika Tkaczyk, Akiko Aizawa, and Jöran Beel. 2018. Position Bias in Recommender Systems for Digital Libraries. In *Transforming Digital Worlds - 13th International Conference, iConference*, Vol. 10766. 335–344.

[5] Gordon V Cormack, Charles LA Clarke, and Stefan Buettcher. 2009. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*. 758–759.

[6] Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Baobao Chang, Xu Sun, Lei Li, and Zhifang Sui. 2024. A Survey on In-context Learning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP*. 1107–1128.

[7] Chongming Gao, Shijun Li, Wenqiang Lei, Jiawei Chen, Biao Li, Peng Jiang, Xiangnan He, Jiaxin Mao, and Tat-Seng Chua. 2022. KuaiRec: A Fully-observed Dataset and Insights for Evaluating Recommender Systems. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 540–550.

[8] F. Maxwell Harper and Joseph A. Konstan. 2016. The MovieLens Datasets: History and Context. *ACM Transactions on Interactive Intelligent Systems* 5, 4 (2016), 19:1–19:19.

[9] Yupeng Hou, Shanlei Mu, Wayne Xin Zhao, Yaliang Li, Bolin Ding, and Ji-Rong Wen. 2022. Towards Universal Sequence Representation Learning for Recommender Systems. In *KDD '22: The 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 585–593.

[10] Yupeng Hou, Junjie Zhang, Zihan Lin, Hongyu Lu, Ruobing Xie, Julian J. McAuley, and Wayne Xin Zhao. 2024. Large Language Models are Zero-Shot Rankers for Recommender Systems. In *Advances in Information Retrieval - 46th European Conference on Information Retrieval, ECIR*, Vol. 14609. 364–381.

[11] Junling Liu, Chao Liu, Peilin Zhou, Renjie Lv, Kang Zhou, and Yan Zhang. 2023. Is ChatGPT a Good Recommender? A Preliminary Study. arXiv:2304.10149 [cs.IR] https://arxiv.org/abs/2304.10149

[12] Chenglong Ma, Ziqi Xu, Yongli Ren, Danula Hettiachchi, and Jeffrey Chan. 2025. PUB: An LLM-Enhanced Personality-Driven User Behaviour Simulator for Recommender System Evaluation. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR*. 2690–2694.

[13] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog* 1, 8 (2019), 9.

[14] Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends® in Information Retrieval* 3, 4 (2009), 333–389.

[15] Ke Sun, Tieyun Qian, Tong Chen, Yile Liang, Quoc Viet Hung Nguyen, and Hongzhi Yin. 2020. Where to Go Next: Modeling Long- and Short-Term User Preferences for Point-of-Interest Recommendation. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI*. 214–221.

[16] Lei Wang, Jingsen Zhang, Hao Yang, Zhiyuan Chen, Jiakai Tang, Zeyu Zhang, Xu Chen, Yankai Lin, Hao Sun, Ruihua Song, Xin Zhao, Jun Xu, Zhicheng Dou, Jun Wang, and Ji-Rong Wen. 2025. User Behavior Simulation with Large Language Model-based Agents. *ACM Transactions on Information Systems* 43, 2 (2025), 55:1–55:37.

[17] Yancheng Wang, Ziyan Jiang, Zheng Chen, Fan Yang, Yingxue Zhou, Eunah Cho, Xing Fan, Yanbin Lu, Xiaojiang Huang, and Yingzhen Yang. 2024. RecMind: Large Language Model Powered Agent For Recommendation. In *Findings of the Association for Computational Linguistics: NAACL*. 4351–4364.

[18] Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2022. Finetuned Language Models are Zero-Shot Learners. In *The Tenth International Conference on Learning Representations, ICLR*.

[19] Ziqi Xu, Chenglong Ma, Yongli Ren, Jeffrey Chan, Wei Shao, and Feng Xia. 2025. Towards Better Evaluation of Recommendation Algorithms with Bi-directional Item Response Theory. In *Companion Proceedings of the ACM on Web Conference 2025, WWW*. 1455–1459.

[20] 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, SIGIR*. 1807–1817.

[21] Guixian Zhang, Guan Yuan, Debo Cheng, Lin Liu, Jiuyong Li, Ziqi Xu, and Shichao Zhang. 2025. Deconfounding representation learning for mitigating latent confounding effects in recommendation. *Knowledge and Information Systems* 67, 7 (2025), 5999–6020.

[22] Junjie Zhang, Yupeng Hou, Ruobing Xie, Wenqi Sun, Julian J. 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 on Web Conference 2024, WWW*. 3679–3689.

[23] 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* (2025).

[24] Zeyu Zhang, Xiaohe Bo, Chen Ma, Rui Li, Xu Chen, Quanyu Dai, Jieming Zhu, Zhenhua Dong, and Ji-Rong Wen. 2024. A survey on the memory mechanism of large language model based agents. *arXiv preprint arXiv:2404.13501* (2024).

[25] Zijian Zhang, Shuchang Liu, Ziru Liu, Rui Zhong, Qingpeng Cai, Xiangyu Zhao, Chunxu Zhang, Qidong Liu, and Peng Jiang. 2025. LLM-Powered User Simulator for Recommender System. In *AAAI-25*. 13339–13347.

[26] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223* 1, 2 (2023).

[27] Kun Zhou, Hui Wang, Wayne Xin Zhao, Yutao Zhu, Sirui Wang, Fuzheng Zhang, Zhongyuan Wang, and Ji-Rong Wen. 2020. S3-Rec: Self-Supervised Learning for Sequential Recommendation with Mutual Information Maximization. In *CIKM '20: The 29th ACM International Conference on Information and Knowledge Management*. 1893–1902.

[28] 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, WWW*. 4653–4661.