MACRec: a Multi-Agent Collaboration Framework for Recommendation

Zhefan Wang*
DCST, Tsinghua University
Beijing 100084, China
wzf23@mails.tsinghua.edu.cn

Yuanqing Yu*
DCST, Tsinghua University
Beijing 100084, China
yyq23@mails.tsinghua.edu.cn

Wendi Zheng
DCST, Tsinghua University
Beijing 100084, China
zhengwd23@mails.tsinghua.edu.cn

Weizhi Ma†
AIR, Tsinghua University
Beijing 100084, China
mawz@tsinghua.edu.cn

Min Zhang†
DCST, Tsinghua University
Beijing 100084, China
z-m@tsinghua.edu.cn

ABSTRACT

LLM-based agents have gained considerable attention for their decision-making skills and ability to handle complex tasks. Recognizing the current gap in leveraging agent capabilities for multi-agent collaboration in recommendation systems, we introduce **MACRec**, a novel framework designed to enhance recommendation systems through multi-agent collaboration. Unlike existing work on using agents for user/item simulation, we aim to deploy multi-agents to tackle recommendation tasks directly. In our framework, recommendation tasks are addressed through the collaborative efforts of various specialized agents, including *Manager*, *User/Item Analyst*, *Reflector*, *Searcher*, and *Task Interpreter*, with different working flows. Furthermore, we provide application examples of how developers can easily use MACRec on various recommendation tasks, including rating prediction, sequential recommendation, conversational recommendation, and explanation generation of recommendation results. The framework and demonstration video are publicly available at https://github.com/wzf2000/MACRec.

CCS CONCEPTS

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

KEYWORDS

Multi-agents; Large Language Models; Recommender Systems

ACM Reference Format:
Zhefan Wang, Yuanqing Yu, Wendi Zheng, Weizhi Ma, and Min Zhang. 2024. MACRec: a Multi-Agent Collaboration Framework for Recommendation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '24), July 14–18, 2024, Washington, DC, USA*. ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3626772.3657669

*Both authors contributed equally to this research.
†Corresponding author. This work is supported by the Natural Science Foundation of China (Grant No. U21B2026, 62372260).

SIGIR '24, July 14–18, 2024, Washington, DC, USA

ACM ISBN 979-8-4007-0431-4/24/07.
https://doi.org/10.1145/3626772.3657669

1 INTRODUCTION

Recommender systems (RSs) play a vital role in improving user experience and platform economic benefits, which have become an essential part of various domains, such as e-commerce, social media, and so on. Currently, the advancement of Large Language Models (LLMs) [1, 9, 15, 22] has introduced LLM-based agents [7, 10, 21] capable of completing complex tasks. These agents' semantic understanding, planning, and decision-making skills unlock new potentials for more nuanced and context-aware recommendations.

Researchers have started to utilize the capabilities of agents to solve recommendation tasks. Existing work like [17, 23, 25] primarily focuses on employing agents for simulating user or item behaviors, providing insights into user preferences but falling short of integration into RSs. On the other hand, some studies [5, 18] attempt to leverage the capabilities of agents to directly build a recommender, primarily using one single agent with planning and memory components and auxiliary tools (e.g., search engine). However, there are various complex decision-making tasks in recommendation scenarios [13, 14], on which single-agent instances are unable to perform well. Multi-agent collaboration, which is near to human workflows, is believed to accomplish complex tasks better with collective intelligence. Although work [11] proposes a multi-agent recommendation framework, it only has limited agent types and a fixed collaboration mode.

To better unleash the potential of multi-agent collaboration for recommendation tasks, we propose **MACRec**, a novel **M**ulti-**A**gent **C**ollaboration framework for recommender systems, designed to harness the diverse capabilities of each agent. Notably, this framework differs from studies for simulation with agents but focuses on building a recommender directly. MACRec provides customizable agents with abilities powered by LLMs and useful tools. For example, we offer *Manager* to plan and manage task execution, *Reflector* to reflect on previous errors, *User/Item Analysts* to analyze user/item characteristics, *Searcher* to search more information using the search tool, and *Task Interpreter* to translate the dialogs into executable recommendation tasks. These agents with different roles work collaboratively to tackle a specific recommendation task.

Additionally, we provide application examples to use MACRec on various recommendation tasks, including rating prediction, sequential recommendation, conversational recommendation, and explanation generation of recommendation results. Considering the

Table 1: Comparison between previous work and our MACRec. Note that *Single-type Agents* indicate all agents serve the same role (e.g., users), while *Multi-type Agents* refer to agents having multiple roles and capabilities (e.g., managers, reflectors).

Model	Objectives	Single-type Agents	Multi-type Agents	Diverse Rec. Scenarios	Open-source
RecAgent [17]	User Simulation	✓			✓
Agent4Rec [23]	User Simulation	✓			✓
AgentCF [25]	U-I Inter Simulation		✓		
RAH [11]	Recommender		✓		
RecMind [18]	Recommender	✓		✓	
InteRecAgent [5]	Recommender	✓			
MACRec	Recommender	✓	✓	✓	✓

varying requirements for agents in different scenarios, we showcase examples of selecting and customizing agents to collaborate on diverse recommendation tasks. Furthermore, we developed an online web interface for our MACRec, providing a user-friendly visualization of the agents' collaboration process. The main strengths of this work can be summarized as follows:

- **A New Multi-agent Collaboration Framework for Recommendation.** Unlike previous studies focused on user/item simulation with agents, we propose a new multi-agent collaboration framework for recommendation **MACRec**. In this framework, agents with different abilities, work collaboratively are involved to tackle specific recommendation tasks.
- **Diverse Applications on Recommendation Scenarios.** We present application examples on various recommendation scenarios, including rating prediction, sequential recommendation, explanation generation, and conversational recommendation.
- **A User-friendly Online Web Interface.** We developed an online web interface for MACRec, visualizing how agents collaboratively tackle tasks.

2 RELATED WORK

2.1 Agents-based Recommendation

Currently, research on integrating LLM-based agents for recommendation can be categorized into two primary orientations: *simulation-oriented* and *recommender-oriented* approaches. Table 1 compares our MACRec and previous agents-based work.

The *simulation-oriented* work focuses on using agents to simulate user behaviors and item characteristics in RSs. RecAgent [23] and Agent4Rec [17] both propose to use agents as user simulators to empower the evaluation of RSs, which feature single-type agents (as users). AgentCF [25] explores the simulation of user-item interactions through user-agents and item-agents. It belongs to a multi-type agent system, with only two types and simple interactions. This line of research aims to provide a deeper understanding of user preferences but falls short of integration into RSs.

The goal of *recommender-oriented* studies is to build a "recommender agent" with planning and memory components to tackle recommendation tasks. InteRecAgent [5] and RecMind [18] primarily focus on improving a single recommender agent's planning and reflection ability. RAH [11] proposes a human-centered framework using LLM Agents as assistants. It supports collaboration among different types of agents, yet only in a fixed mode, whereas MACRec enables adaptable collaboration for various uses. Moreover, RAH lacks publicly accessible code or demos. To the best of our knowledge, MACRec is the first open-source framework supporting multi-type agents for diverse recommendation scenarios.

2.2 Multi-agent Collaboration

Multi-agent systems, initially grounded in DAI [2] and MAS [12], evolved with foundational concepts of agent coordination and communication by Wooldridge and Jennings [19]. The advent of powerful LLMs [1, 9, 15, 22] has shifted focus towards their application in multi-agent collaboration. Brown et al. [1] demonstrated LLMs' potential in human-like dialogues, applicable to agent-agent communication. Nascimento et al. [8], Vinyals et al. [16] illustrates how LLM agents can collaborate for shared objectives, achieving specific and complex task solutions. Recent work [3, 4, 24] leverage multi-agent collaboration to achieve better performance on complex tasks. CAMEL [6] and AutoGen [20] focus on communicative agent systems for complex task solutions through inter-agent dialogue. However, existing research on multi-agent collaboration has not investigated its potential in recommendation scenarios.

3 THE MACREC FRAMEWORK

3.1 Framework Overview

Figure 1 illustrates our proposed multi-agent collaboration recommendation framework. A sequential recommendation task is given as an example.

As shown in the example in Figure 1, the *Task Interpreter* first translates the task in a better way to understand. Then, as the central component of the entire system, the *Manager* starts calling other agents to obtain detailed analyses of the user and items. These agents, including the *Searcher* and the *User/Item Analyst*, support the call of some tools, e.g., the *Searcher* has access to the search engine and the *User/Item Analyst* can access detailed information about users and items. After receiving responses from the *Searcher* and *Analyst*, the *Manager* will attempt to provide an answer, i.e., give a ranking order of the candidate sets. The *Reflector* will be responsible for analyzing and reflecting on the *Manager*'s answer in the last trial and giving suggestions, e.g., modifying the answer format to follow the task requirements. Eventually, the Manager

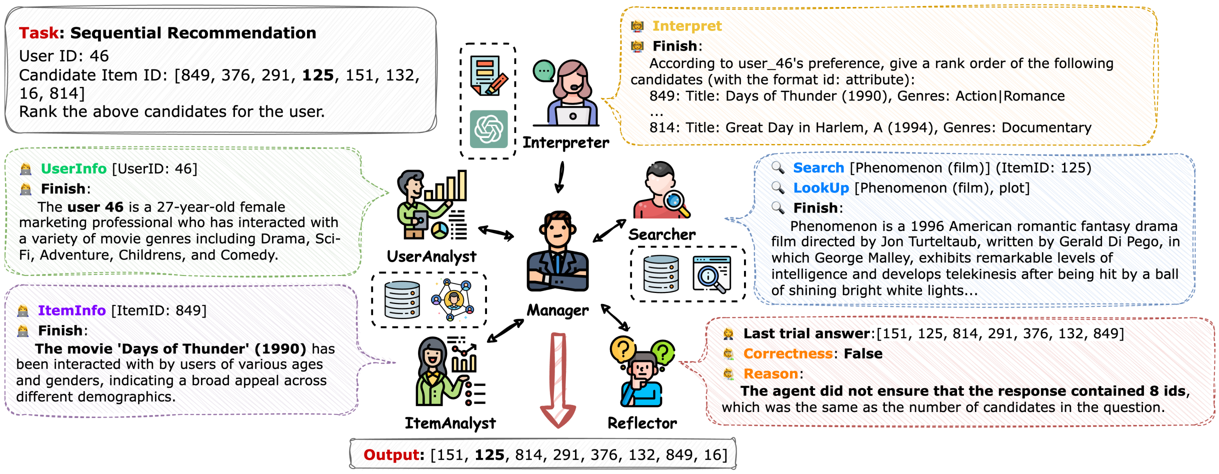


Figure 1: The Framework of MACRec. We take a sequential recommendation task as an example to show how these agents work collaboratively.

will reattempt to solve the task based on the reflections and provide a more reasonable answer, e.g., adding the missed item ID.

The following sections will detail each agent's specific characteristics and functions.[1]

3.2 Agent Roles

3.2.1 ***Manager***. For the given task, the *Manager* would assign sub-tasks to other agents and complete the main execution process. It oversees the collaboration among all other agents.

The *Manager* always performs the three steps of *Thought*, *Action*, and *Observation* alternately. In the *Thought* phase, the *Manager* reasons about the current situation of the task (e.g., whether the analysis is sufficient, whether additional information is needed, etc.). During the *Action* phase, the *Manager* can choose to give an answer to end the task or seek help from other agents (under a particular interface format). Responses given by other agents will be given in the *Observation* phase of the *Manager*.

3.2.2 ***Reflector***. The *Reflector* is responsible for judging the correctness of the answer given by the Manager. A further reflection will be given if the *Reflector* determines the answer is correct.

The *Reflector* will step in when the *Manager* is about to perform the second or more runs on the same task input. If the *Reflector* judges that the answer given by the *Manager* has no room for improvement, the *Manager* will no longer perform the current run. Otherwise, the *Reflector* will further summarize where the *Manager* can be improved, e.g., not considering the few highly rated items/movies in the user's historical interactions.

3.2.3 ***User/Item Analyst***. *User/Item Analyst* specializes in examining and understanding the characteristics and preferences of users, as well as the attributes of items.

The *Analyst* will be given access to two tools to assist in the analysis, including info database and interaction retriever. The *Analyst* can get the user profile of each user and the attributes of each item through the info database. Through the interaction retriever, the *Analyst* can get the user/item interaction history before the current time. With the combination of these two tools, the *Analyst* can have an in-depth analysis of the user or the item.

3.2.4 ***Searcher***. The *Searcher* is responsible for searching under the requirements given by the *Manager* with the search tool, and finally summarizing the text reply to the *Manager*.

Take Wikipedia as an example of a search tool. The *Searcher* can give a search query to get the most relevant entry in Wikipedia. The *Searcher* can further retrieve passages in a specific entry where the given keywords exist. Eventually, the *Searcher* is asked to summarize the paragraph to respond the the *Manager*'s query.

3.2.5 ***Task Interpreter***. The *Task Interpreter* translates the dialogs into executable recommendation tasks.

The *Task Interpreter* will get the conversation history when starts running. Since conversation histories can be long, the *Task Interpreter* will only get the last part of the history. The *Task Interpreter* also has access to call the text summarization tool to get a more concise overview of the history. Eventually, the *Task Interpreter* will give a specific description of the task requirements that will be used to guide the subsequent runs of the *Manager*.

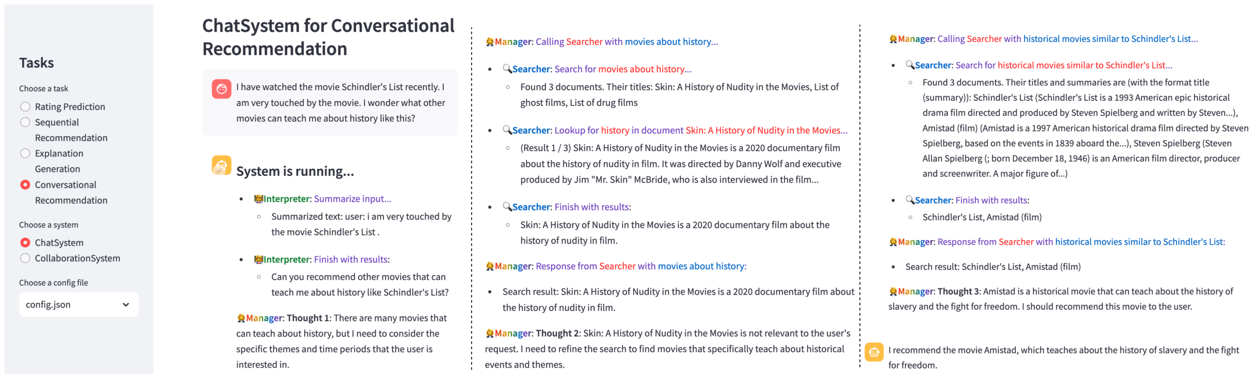
4 APPLICATIONS ON RECOMMENDATION SCENARIOS

Here, we present the applications of MACRec on four recommendation scenarios. Table 2 summarizes the agents' selection for each scenario.

Task	U.Analy.	I.Analy.	Reflector	Searcher	Interpreter
RP	☑	☑	☐		
SR	☑	☐	☑		
EG	☑	☑	☐	☑	
CR				☑	☑

Table 2: The agents' selection for four applications supported by MACRec. ☑ means required and ☐ means optional.

[1] Code is available at https://github.com/wzf2000/MACRec.



(a) Interprete the dialog into a task. (b) Search for movies about history. (c) Search for movies similar to Schindler's List.

Figure 2: The web interfaces of our MACRec, along with a case of how three agents collaboratively address a conversational recommendation task. The interface is composed by the leftmost configuration panel and the main interaction panel.

4.1 Rating Prediction (RP)

Rating prediction task involves predicting the numerical rating a user might give to an item, such as a movie or a product, based on their preferences and historical interactions.

In the rating prediction task, each user will have different rating preferences. The *User Analyst* can provide a detailed analysis of the user's historical interactions and preferences. Meanwhile, the *Manager* also needs characteristic analysis of the target item, which can be provided by the *Item Analyst*. With the help of two types of *Analysts*, the *Manager* can know the user's tendency to rate and the item's recent ratings before giving a prediction.

4.2 Sequential Recommendation (SR)

Sequential recommendation systems analyze the sequence of items a user has interacted with to predict their next likely interest.

Modeling of user's long-term and short-term interests is important in sequential recommendation tasks. Hence, the *User Analyst*'s role is self-evident. The number of relevant items in the sequence is significantly higher than the rating prediction task. It is hard to ask the *Item Analyst* to analyze every item that appeared in either the history or the candidate set. Moreover, given that the answers to the sequential recommendation task are much more complex (i.e., a ranking order of the candidate set), the *Reflector* can help to avoid the *Manager* getting into formatting troubles. A single round of behavioral analysis may omit consideration of long-term user behavior, and reflection on this is something the *Reflector* can do.

4.3 Explanation Generation (EG)

This task involves generating understandable and relevant explanations for the recommendations provided to users.

The explanation generation task also requires a detailed analysis of both the user and the item. In addition, more information about the item may also help the *Manager* understand the user's behavior towards it. For example, a user may have similar preferences for multiple movies by the same director. The information about the director may not be contained in the dataset. Retrieving these extra pieces of information is suitable for the *Searcher* to perform.

4.4 Conversational Recommendation (CR)

Conversational recommender systems engage users in a dialogue to refine their preferences and deliver more accurate suggestions.

In conversational scenarios, the user's input text is not necessarily explicitly instructive. Hence, the *Task Interpreter* can help translate the conversation history into a more concise and clear task prompt. In addition, the user's input requirements may contain information unknown to the *Manager*. In this case, the *Searcher* can help the Manager understand what the user mentioned.

Beyond the abovementioned applications, our framework can support other scenarios by customizing the configurations.

5 INTERFACE DEMONSTRATION

Figure 2 presents the web interfaces of our framework, along with a detailed case study demonstrating the collaborative efforts of three agents in addressing a conversational recommendation task.

The interface can be divided into two main panels. 1) **Configuration panel**, where users can select different tasks to tackle, such as "Rating Prediction." Users can also customize different systems and configuration files for the task execution. 2) **Interaction panel**, where the whole collaboration process takes place. Agents with different abilities would complete the task collaboratively.

In Figure 2, the user has expressed a preference for the movie "Schindler's List" and seeks recommendations for similar historical movies. The *Interpreter* summarizes this input and translates it into a clearer task. Then, the *Manager* calls for the help of the *Searcher* for two rounds, searching for movies about history and movies similar to "Schindler's List". According to all the information, the *Manager* gives the final recommendation movie "Amistad".

6 CONCLUSION

In this work, we propose a novel LLM-based multi-agent collaboration framework for recommendation, called MACRec. Unlike existing studies on using agents for user/item simulation, we directly tackle recommendation tasks through the collaboration of various agents. We present applications of MACRec on four different recommendation tasks. Moreover, We developed an online web interface for MACRec, visualizing how agents work collaboratively.

REFERENCES

[1] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems* 33 (2020), 1877–1901.

[2] Brahim Chaib-Draa, Bernard Moulin, René Mandiau, and Patrick Millot. 1992. Trends in distributed artificial intelligence. *Artificial Intelligence Review* 6 (1992), 35–66.

[3] Weize Chen, Yusheng Su, Jingwei Zuo, Cheng Yang, Chenfei Yuan, Chen Qian, Chi-Min Chan, Yujia Qin, Yaxi Lu, Ruobing Xie, et al. 2023. Agentverse: Facilitating multi-agent collaboration and exploring emergent behaviors in agents. *arXiv preprint arXiv:2308.10848* (2023).

[4] Yilun Du, Shuang Li, Antonio Torralba, Joshua B Tenenbaum, and Igor Mordatch. 2023. Improving Factuality and Reasoning in Language Models through Multiagent Debate. *arXiv preprint arXiv:2305.14325* (2023).

[5] 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* (2023).

[6] Guohao Li, Hasan Abed Al Kader Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. 2023. Camel: Communicative agents for" mind" exploration of large scale language model society. *arXiv preprint arXiv:2303.17760* (2023).

[7] Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. 2021. Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332* (2021).

[8] Nathalia Nascimento, Paulo Alencar, and Donald Cowan. 2023. GPT-in-the-Loop: Adaptive Decision-Making for Multiagent Systems. *arXiv preprint arXiv:2308.10435* (2023).

[9] OpenAI. 2023. GPT-4 Technical Report. *arXiv preprint arXiv:2303.08774* (2023).

[10] Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. 2023. Hugginggpt: Solving ai tasks with chatgpt and its friends in huggingface. *arXiv preprint arXiv:2303.17580* (2023).

[11] Yubo Shu, Hansu Gu, Peng Zhang, Haonan Zhang, Tun Lu, Dongsheng Li, and Ning Gu. 2023. RAH! RecSys-Assistant-Human: A Human-Central Recommendation Framework with Large Language Models. *arXiv preprint arXiv:2308.09904* (2023).

[12] Peter Stone and Manuela Veloso. 2000. Multiagent systems: A survey from a machine learning perspective. *Autonomous Robots* 8 (2000), 345–383.

[13] Peijie Sun, Yifan Wang, Min Zhang, Chuhan Wu, Yan Fang, Hong Zhu, Yuan Fang, and Meng Wang. 2024. Collaborative-Enhanced Prediction of Spending on Newly Downloaded Mobile Games under Consumption Uncertainty. *WWW2024, Industry Track* (2024).

[14] Peijie Sun, Le Wu, Kun Zhang, Xiangzhi Chen, and Meng Wang. 2023. Neighborhood-Enhanced Supervised Contrastive Learning for Collaborative Filtering. *IEEE Transactions on Knowledge and Data Engineering* (2023).

[15] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971* (2023).

[16] Oriol Vinyals, Igor Babuschkin, Wojciech M Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H Choi, Richard Powell, Timo Ewalds, Petko Georgiev, et al. 2019. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature* 575, 7782 (2019), 350–354.

[17] Lei Wang, Jingsen Zhang, Hao Yang, Zhiyuan Chen, Jiakai Tang, Zeyu Zhang, Xu Chen, Yankai Lin, Ruihua Song, Wayne Xin Zhao, et al. 2023. When Large Language Model based Agent Meets User Behavior Analysis: A Novel User Simulation Paradigm. *arXiv preprint ArXiv:2306.02552* (2023).

[18] 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).

[19] Michael Wooldridge and Nicholas R Jennings. 1995. Intelligent agents: Theory and practice. *The knowledge engineering review* 10, 2 (1995), 115–152.

[20] Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Shaokun Zhang, Erkang Zhu, Beibin Li, Li Jiang, Xiaoyun Zhang, and Chi Wang. 2023. Autogen: Enabling next-gen llm applications via multi-agent conversation framework. *arXiv preprint arXiv:2308.08155* (2023).

[21] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629* (2022).

[22] Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, et al. 2022. Glm-130b: An open bilingual pre-trained model. *arXiv preprint arXiv:2210.02414* (2022).

[23] An Zhang, Leheng Sheng, Yuxin Chen, Hao Li, Yang Deng, Xiang Wang, and Tat-Seng Chua. 2023. On generative agents in recommendation. *arXiv preprint arXiv:2310.10108* (2023).

[24] Hongxin Zhang, Weihua Du, Jiaming Shan, Qinhong Zhou, Yilun Du, Joshua B Tenenbaum, Tianmin Shu, and Chuang Gan. 2023. Building cooperative embodied agents modularly with large language models. *arXiv preprint arXiv:2307.02485* (2023).

[25] Junjie Zhang, Yupeng Hou, Ruobing Xie, Wenqi Sun, Julian McAuley, Wayne Xin Zhao, Leyu Lin, and Ji-Rong Wen. 2023. Agentcf: Collaborative learning with autonomous language agents for recommender systems. *arXiv preprint arXiv:2310.09233* (2023).