

---

# RecAgent: A Novel Simulation Paradigm for Recommender Systems

---

Lei Wang Jingsen Zhang Xu Chen Yankai Lin  
 Ruihua Song Wayne Xin Zhao Ji-Rong Wen  
 Gaoling School of Artificial Intelligence  
 Renmin University of China, Beijing, China

## Abstract

Recommender system has deeply revolutionized people’s daily life and production, bringing a large amount of business value. In the recommendation domain, simulation and real data-based studies are two typical research paradigms, with each having different advantages. Previously, real data-based studies occupy more important positions, since accurately simulating the user preference is quite difficult. Recently, large language models (LLM) have shown great potential to achieve human-like intelligence, which provides new opportunities to overcome the shortcomings of simulation-based studies and thus highlight their advantages, such as much more application scenarios and cheaper data acquisition strategies. To shed lights on this direction, in this paper, we introduce an LLM-based recommender simulator called RecAgent. Our simulator is composed of two modules: (1) the user module and (2) the recommender module. The user module can browse the recommendation website, communicate with other users and broadcast messages on the social media. The recommender module is designed to provide search or recommendation lists to the users, and one can design different models to implement the recommender. All the users take actions based on LLMs, and can freely evolve like in the real world. We present several case studies to demonstrate that the users in our simulator can indeed behave in a reasonable manner as expected. Our project has been released at <https://github.com/RUC-GSAI/YuLan-Rec>.

## 1 Introduction

As an active research field, recommender system exemplifies a dynamic intersection of academia and industry, contributing a wealth of insights and innovations [25, 11, 10, 26, 6, 5, 26]. The bedrock of these systems is an accurate understanding of user preferences [25, 26, 32, 5, 28, 10, 11], playing a pivotal role in enhancing the design and evaluation of system strategies.

Traditional works in the recommender system field generally adhere to real data-based studies [10, 11, 26, 31, 6], grounded in accumulating user behavioral data through either interaction with online environments or enlistment of annotators. However, this paradigm confronts two salient challenges. Firstly, this paradigm proves resource-intensive and lacks sustainability. This limitation restricts its utility to a narrow band of fixed problems, thereby impeding swift adaptation to burgeoning new problems in Web 2.0 (*e.g.*, RL-based recommendation and explainable recommendation [7, 27, 34, 2, 8]). Second, the richness of user interaction data in real-world contexts can be difficult to capture comprehensively. For instance, a user might choose to watch a film based on a friend’s casual mention in conversation, an influencing factor that is difficult to capture through the lens of a recommender system. These hurdles significantly impact the trajectory of our in-depth investigation into recommender systems.

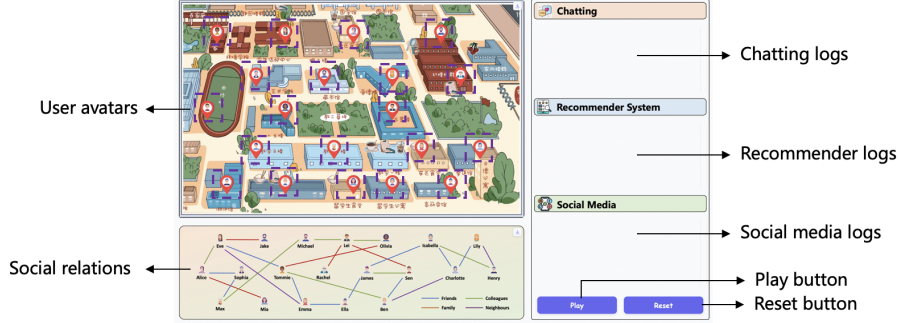


Figure 1: Visual interface of RecAgent

Broadening our perspective on artificial intelligence research, we discern that simulation-based studies serve as a crucial counterpoint to research primarily reliant on real-world data-based studies. These simulation-based approaches [35, 33, 12, 20, 1] are often more cost-effective and adaptable to various application contexts. For instance, in the field of reinforcement learning (RL), given the substantial costs associated with agent-environment interactions in real-world scenarios, numerous works are assessed leveraging environmental simulators (*e.g.*, gym and MuJoCo) [35, 16, 24, 3]. This evidences the value of simulation-based studies in advancing our understanding and development of complex systems. Within the realm of recommender system, the utility of simulation-based studies [13, 23, 15, 17, 2, 8] can potentially be contentious. Unlike other domains where simulators can be intuitively constructed with relative ease (*e.g.*, simulating car moving process based on objective physical laws), simulating user subjective preferences in the domain of recommendation presents a far greater complexity. This highlights the unique challenges inherent to the recommendation field, where understanding and modeling user behavior can be significantly more intricate.

Recently, large language models (LLM) [30, 29, 9, 4, 18, 21] have shown great potential to achieve human-like intelligence. Given this advancement, we posit that it is opportune to shift our focus toward simulation-based studies in the field of recommender system. LLMs could potentially offer a more profound comprehension of user preferences, thereby enabling the simulation of subjective information, a previously challenging endeavor. This underscores the potentially transformative impact of LLMs in the recommender system field. In this paper, we open the direction of LLM-based recommendation simulation studies. In specific, we build a recommender simulator (see Figure 1), which is composed of two components. The first one is a user module, which can visit the recommendation website, communicate with the other users and send message on the social media. The second one is a recommender module, which is designed to return the search or recommendation lists to the users. All the users behave and produce their thoughts based on LLM, and can freely evolve in the simulator. In the following, we first introduce the detailed architecture of our simulator and then present several case studies on the simulated user behaviors. At last, we introduce many potential opportunities brought by our simulator in the recommendation domain.

## 2 RecAgent

There are two major components in our simulator, that is, the user module and the recommender module. The user module can (1) browse the recommendation website, (2) chat with the other users or (3) broadcast messages on the social media. The recommender module is responsible for generating the search or recommendation results for the users. We summarize the interactions between the user and recommender and different users in Figure 2. In the following, we first introduce the basic environment setup of our simulator, and then detail the user and recommender modules separately.

### 2.1 Environment Setup

The core of our simulator includes a server implemented based on LangChain<sup>1</sup> and a front-end realized based on Gradio<sup>2</sup>. The virtual world in the simulator evolves in a round by round manner. In every round, each user takes an action, and the server organizes the action information, and sends it

<sup>1</sup><https://github.com/hwchase17/langchain>

<sup>2</sup><https://github.com/gradio-app/gradio>

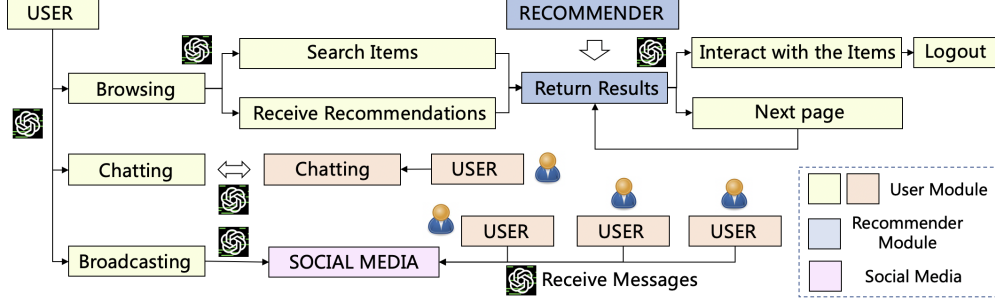


Figure 2: Overall framework of RecAgent. We use different colors to represent different roles in our simulator. The positions where we need to query LLMs are highlighted by the labels of ChatGPT

to the front-end, which is responsible for rendering the action in the visual interface. For example, if a user decides to chat with her friend, then the server delivers this information to the front-end, which highlights the user icon and present a label indicating that she is chatting. The front-end includes the user avatars, the simulator logs and the relations between different users. We also set the buttons of start, stop and reset to control the running process of our simulator. The complete interface of the front-end is presented in Figure 1.

## 2.2 The User Module

Intuitively, the user behaviors in a real-world recommender system are influenced by both internal and external factors. Examples of the internal factors include the user temper, habits, and gender, which can be explicitly described as a prompt to query the LLM. For the external factors, we mainly consider the influence of the other users. To simulate the user influences, we design two scenarios, that is, one-to-one chatting, and one-to-many broadcasting on social media.

Following the above considerations, we allow each user in our simulator to take three types of actions: (1) entering the recommender system, (2) entering the social network (for one-to-one chatting or one-to-many broadcasting) and (3) do nothing. Each user maintains a memory bank, which can be updated by all the above actions or the reflections summarized by the user herself. At the beginning of each round, the user can choose to do something or nothing using the following prompt<sup>3</sup>:

```
<user> must take one of the actions below:
(1) Enter the recommender system. If so, <user> will be
    recommended some movies, from which <user> can watch some
    movies, or search for movies by himself.
(2) Enter the Social Network website. <user> can chat
    privately with friends or publish a post to all people <user>
    knows.
(3) Do Nothing.
What action would <user> like to take? Respond in one line.
If <user> want to enter the Recommender website, write:
[RECOMMENDER]:: <user> enter the Recommender website.
If <user> want to enter the Social Network website, write:
[SOCIAL]:: <user> enter the Social Network website.
If <user> want to do nothing, write: [NOTHING]:: <user> does
nothing.
```

### 2.2.1 Entering the recommender system

Intuitively, a user may visit a website by two different manners: (1) the user has specific goals, and directly search the website for his desired information, and (2) the user does not have explicit

<sup>3</sup>It should be noted that, for saving the space, we do not present the basic information of the user (*e.g.*, the user memory) in the prompt

purposes, and aimlessly browses the website, hoping to obtain the information of interest by the recommendations from the website. Inspired by such intuitions, we allow the users to interact with the recommender system by both searching items and receiving recommendations.

In specific, once a user enters the recommender system, she will be firstly presented with an initial homepage recommendation. Then there are four following actions that the user can take, including: (1) searching movies, (2) watching movies, (3) going to the next page and (4) leaving the system. We ask the user which action she would like to take based on the following prompt:

```
<user> must take one of the four actions below:
(1) Watch some movies in the item list returned by the
recommender system.
(2) See the next page.
(3) Search items.
(4) Leave the recommender system.
If <user> has recently heard about a particular movie on a
social networking website, <user> might want to search for
that movie on the recommender system.
What action would <user> like to take? Respond in one line.
If <user> wants to watch movies on the recommender system,
write:
[WATCH]:: movie names in the item list returned by the
recommendation system, only movie names, separated by
semicolons.
If <user> wants to see the next page, write:
[NEXT]:: <user> see the next page.
If <user> wants to search for a specific item, write:
[SEARCH]:: single, specific item name <user> wants to search.
If <user> wants to leave the recommender system, write:
[LEAVE]:: <user> leaves the recommender system.
```

In the following, we present more details on the above four actions in the recommender system.

**Searching movies:** This action corresponds to the scenario where the user has specific goals. The user will search the system for the movies that she would like to watch. We use the following prompt to obtain the query of the user:

```
<user> is searching in recommender system. If you are going
to pretend to be <user>, what movies would you be interested
in and search for in the system?
```

According to the user query, the recommender system will return the search results based on the similarities between the queried movies and candidates in the database.

**Watching movies:** This action is triggered once the user has found the movies of interests (either in the context of searching or recommendation). Intuitively, a real user may produce much feelings after watching a movie, which will be stored in her memory and influence her future cognition and behaviors. In our simulator, we also query the user feelings on the watched movies, and leverage such information to update the user memory bank, where we use the following prompt to accomplish these operations:

```
<user> has just finished watching items. <user> has not seen
these movies before. If you are going to pretend to be <user>,
how will you feel about each movie after watching it?
```

**Going to the next page:** In our simulator, the recommender system returns the search or recommendation result including five movies each time. If the user do not satisfy with any movie in the current results, she may decide to ask for the next page results. If this action is taken in the context of searching, then the results are produced based on the movie similarities. Otherwise, the results

are generated based on the recommendation algorithms embedded in the recommender module. It should be noted that one can flexibly design different recommendation algorithms in our simulator to evaluate their influence on the user behaviors.

**Leaving the system:** To simulate more reliable user behaviors, we allow the user to exit the recommender system at any step, for example, after the user watches a movie or the user does not find any movie of interest.

### 2.2.2 Chatting with the other users and broadcasting messages on the social media

In the real world, the behaviors of the users in a recommender system may be not only determined by the factors inside the system. There can be also many external factors, among which, we believe, the information dissemination between different users is an important one, for example, a user may watch a movie because of her friends' recommendation or the movie is widely discussed on the social media. In reality, people may disseminate information in different ways, and we abstract them into two types in our simulator: (1) one-to-one chatting and (2) one-to-many broadcasting.

More specifically, at each round of the simulator, if the user decide to enter the social network, then we determine whether she would like to chat or broadcast based on the following prompt:

```
<user> must take one of the two actions below:
(1) Chat with one acquaintance about movies recently watched
    on the recommender system, or movies heard about on a social
    networking website.
(2) Publish a post to all acquaintances about movies recently
    watched on the recommender system, or heard about on a social
    networking website.
What action would <user> like to take? Respond in one line.
If <user> wants to chat with an acquaintance, write:
[CHAT]::acquaintance's name;what to say.
If <user> wants to publish a post to all acquaintances, write:
[POST]::what to post.
```

For one-to-one chatting, a straightforward method is recurrently adding the previous dialog history into the prompt to query the next response. However, this method is quite inefficient, since one needs to frequently access the LLMs. To overcome this shortcoming, we firstly initialize the chatting background, and let LLMs generate the complete dialog at one time, which can greatly enhance the simulation efficiency.

For one-to-many broadcasting, we obtain the message that the user would like to post based on the following prompt:

```
<user> wants to publish a post to all acquaintances.
The post should be related to recently watched movies on the
recommender system.
If you are going to pretend to be <user>, what will you post?
Respond in one line.
```

Once the user has posted a message, all her friends on the social media can receive this message, and may chat with her, which launches another one-to-one chatting thread.

## 2.3 The Recommender Module

In our simulator, the recommender module is implemented with the real search and recommendation algorithms, which aims to respond to the user queries and generate recommendation lists for the users. The recommender module is highly flexible, where one can easily change the search/recommendation strategies, and initialize the candidate items with different public available or simulated datasets.

While our current recommender module is designed for general recommendation, it can be easily extended to the other scenarios by slightly change existing settings. For example, one can simulate conversational recommendation by allowing chats between the users and system.

After simulating the user behaviors, all the information in the recommender module can be exported for offline studies, and one can also import offline datasets into the recommender module for user behavior simulation.

### 3 Case Studies

In this section, we show some examples of the user behaviors in our simulator.

#### A user browses the recommendation website, watches several movies and generates feelings.

- Sophia enters the Recommender website. Sophia is going to recommender system website.
- Sophia is recommended [<Naked Gun 33 1/3: The Final Insult(1994)> . "<All the Rage (a.k.a. It's the Rage)(1999)> ", <Police Academy 3: Back in Training(1986)> ' <Dead Zone, The (1983)> <Bandits (1997)> ']. Sophia is looking next page.
- Sophia is recommended [ ' <It Happened One Night(1934)> , <Hard 8(akaSydney,a.k.a.Hard Eight)(1996)> ' , <Third Miracle,The(1999)> ' <Tie Me Up! Tie Me Down!(1990)> <Sunset Park(1996)> ].
- Sophia is looking next page.
- Sophia is recommended [ <Little Princess,The(1939)> <Love Stinks(1999)> <Once Upon a Time... When We Were Colored(1995)> , <Frank and Ollie(1995)> ' , <Corruptor, The(1999)> '].
- Sophia is looking next page.
- Sophia is recommended [ ' <Thing From Another World,The(1951)> <Chamber, The(1996)> ' , <Coma(1978)> ' , <How to Make an American Quilt(1995)> ' , <One Little Indian(1973)> '].
- Sophia watched <Thing From Another World,The(1951)> ; <Coma(1978)>
- Sophia feels: <Thing From Another World,The(1951)> : I found it to be an intriguing and suspenseful sci-fi classic with great performances and direction. <Coma(1978)> : It was a thrilling and thought-provoking medical drama that kept me on the edge of my seat.

#### A user browses the next page of the recommendation system.

- Olivia enters the Recommender website.
- Olivia is going to recommender system website.
- Olivia is recommended [ ' <Fantastic Voyage (1966)> ' , ' <Straight Story, The (1999)> ' , ' <Pajama Game, The (1957)> ' , ' <Money Talks (1997)> ' , ' <Wings (1927)> '].
- Olivia is looking next page.
- Olivia is recommended [ ' <Calendar Girl (1993)> ' , ' <Kolya (1996)> ' , ' <Fatal Attraction (1987)> ' , ' <Graveyard Shift (1990)> ' , ' <Montana (1998)> '].
- Olivia watched <Calendar Girl (1993)> ; <Kolya (1996)> ; <Fatal Attraction (1987)> ; <Graveyard Shift (1990)>

#### A user searches a specific movie in the recommendation system.

- Olivia is recommended [ ' <Surf Nazis Must Die (1987)> ' , ' <Shanghai Surprise (1986)> ' , ' <My Son the Fanatic (1998)> ' , ' <Avengers, The (1998)> ' , ' <SubUrbia (1997)> '].
- Olivia is searching "The Lion King" in the recommender system..
- Olivia is recommended [ 'Lion King, The (1994)' , 'Jungle Book, The (1994)' , 'Jungle Book, The (1967)' , 'Cheetah (1989)' , 'Tarzan (1999)' ].
- Olivia watched <Lion King, The (1994)> ; <Jungle Book, The (1994)> ; <Jungle Book, The (1967)>

#### A user chats with the other ones on her recently watched movies.

- Tommie watched <A Chef in Love (1996)>; <Killing, The (1956)>; <Mary Shelley's Frankenstein (1994)>; <Repulsion (1965)>; <Mariachi, El (1992)>
- Tommie feels:<A Chef in Love (1996)>: Impressed by the beautiful cinematography and the unique love story set in the culinary world.
- Eve enters the Social Network website
- Eve is going to social media.
- Eve is chatting with Tommie. Said: Hi Tommie, have you watched any good romantic or mystery movies lately? I'm always on the lookout for recommendations.
- Tommie said "Actually, I recently watched a really unique movie called <A Chef in Love>. It's not a typical romantic movie, but it's definitely worth checking out if you're looking for something different. And as for mystery, I haven't watched anything in that genre recently, but I'll keep an eye out for you, Eve."
- Eve said "Thanks for the recommendation, Tommie! I'll definitely check out <A Chef in Love>. Let me know if you come across any good mystery movies in the future. It was nice chatting with you!"

#### **A user broadcasts messages to share her feelings on the recently watched movies.**

- Eve enters the Social Network website
- Eve is going to social media.
- Eve is posting.
- Eve posted: "Just watched <Gone Girl> and <A Walk to Remember> and they were both amazing! Any other romance or mystery recommendations for my next movie night? #movienight #romance #mystery #drama"
- ['Tommie', 'Alice', 'Michael', 'Jake', 'Olivia', 'Lei'] get this post.

#### **A user responds to the post of the other users.**

- Tommie enters the Social Network website.
- Tommie is going to social media.
- Tommie is posting.
- Tommie posted: "I'm feeling anxious today, but watching a good comedy always helps me feel better! Any recommendations?"
- Eve enters the Social Network website
- Eve is going to social media.
- Eve is chatting with Tommie. Said: Hi Tommie, I saw your post about feeling anxious. Have you seen the movie "Crazy, Stupid, Love"? It always makes me laugh and feel better when I'm feeling down.
- Tommie said "Thanks for the recommendation! I'll definitely check it out. Do you have any other favorite comedies?"
- Eve said "Sure, I have a few more favorites. Have you seen 'The Hangover' or 'Bridesmaids'? They're both hilarious. Let me know what you think of 'Crazy, Stupid, Love' after you watch it. Take care!"
- Tommie said "Thanks, I'll add those to my list! And I'll definitely let you know what I think of 'Crazy, Stupid, Love'. Thanks for the chat, Eve. Have a great day!"
- Eve said "You're welcome, Tommie! Glad I could help. Have a great day too!"

#### **A user searches movies recommended from the other users.**

- Tommie said: "Actually, Eve, I have a question. Do you have any recommendations for movies with great design? I really appreciate good visual aesthetics in movies."
- Eve said: "Of course, Tommie! I love movies with great design too. One movie that comes to mind is <Blade Runner 2049>. The visuals are stunning and really transport you into the world of the movie. Another movie that has great design is

<The Grand Budapest Hotel>, which has a whimsical and colorful aesthetic. Have you seen either of those?"

- Tommie enters the Recommender website.
- Tommie is going to recommender system website.
- Tommie is recommended ['<Last of the Mohicans, The (1992)>', '<Rocky IV (1985)>', '<Story of Us, The (1999)>', '<Verdict, The (1982)>', '<Chopping Mall (a.k.a. Killbots) (1986)>'].
- Tommie is searching: I would be interested in searching for <The Grand Budapest Hotel>.

## 4 Potential Opportunities Brought by RecAgent

In this section, we discuss the potential utilization of RecAgent across several applications.

**Cold Start Recommendation.** In the recommendation domain, the cold start problem has obsessed people for a long time [14, 22]. We believe that RecAgent may provide new opportunities for alleviating this problem. To begin with, one can align the profiles of the cold start users in the real and virtual worlds. Then, even if the users in the real world have no (or a small number of) interactions with the items, we can observe how their projections in the virtual world behave, and collect these behaviors to augment the real-world data for learning more accurate recommender models.

**Social Recommendation.** In the field of social recommendation, previous models mostly assume that the users with a social connection may behave similar. However, such an assumption is too strong, in our simulator, one can observe that even if two users are close friends, their preferences can be also quite different. In our simulator, the information dissemination process among the users on the social media is completely transparent. Thus, we have sufficient knowledge to learn the influence of the social connections, where we do not need to impose strong assumptions.

**RL-based Recommendation.** Recently, reinforcement learning (RL) based recommender models have attracted increasing attention, aiming to model the user long-term engagement. An important problem in RL-based recommendation is the lack of accurate simulators. RecAgent can be a natural solution to this problem, and one can design different prompts for querying the user feedback on a recommendation list based on LLM. Here, LLM basically plays the role of reward model. Compared with the previous reward models, LLM can better understand the users, and thus can provide more accurate rewards.

**Explainable Recommendation.** Providing explanations for the recommendation results have shown to be significant to improve the user trust and satisfaction to the system. However, how to evaluate the explanations is a very hard problem. Based on RecAgent, we may directly ask LLM about the feeling of the users on the explanations. For example, "without the explanation, would you like to interact with the item?" (the necessity of the explanation), "Does the explanation provide you with sufficient information for the items?" (the informativeness of the explanation), "Is the explanation pervasive enough for you?" (the pervasiveness of the explanation).

We believe that there are many more scenarios (*e.g.*, fairness aware recommendation, debiased recommendation) that RecAgent can be leveraged to improve traditional recommendation studies. As RecAgent becomes more perfect, it may even change existing industry recommender deployment processes. For example, people can maintain an image of their online environment, and then simulate the effect of the newly developed models (by replacing the recommender module in RecAgent). At last, the model with the best simulation results is deployed online.

## 5 Related Work

Simulation-based studies in the recommendation domain are not a new concept, and there have been many previous efforts to build recommender simulators. For example, RecSim [13] is a simulation platform, where the users and system can sequentially interact with each other, and one can freely assign different features to the users or items. Virtual Taobao [23] can simulate users which are aligned with the real-world datasets. MINDSim [15] is a user simulator to produce reasonable user behaviors in a news website. Despite effectiveness, all the above simulators are based on simple human-designed rules, and the user behaviors are usually constrained by strong assumptions. In



contrast, we base our simulator on LLMs, which can better understand the users, and the user behaviors are fully determined by LLMs without any external assumptions.

Our simulator is inspired by a pioneering work called generative agent [19]. However, this work aims to simulate the human’s daily life in a general manner. Another related work is AgentVerse<sup>4</sup>, which simulates the interactions between the students and teacher in the education domain. Different from the previous work, our simulator focuses on the recommendation, which, we believe, is another killer application of the idea of LLM-based simulation. Unlike CV, NLP and many other domains, recommender system is a highly subjective task, and LLMs can better understand humans, which provides new opportunities for the previously unimaginable simulation-based studies.

## 6 Conclusion and Future Work

In this paper, we introduce a novel recommender simulator based on LLM. Our simulator includes two modules, which simulate the users and recommender system, respectively. We design different actions for the users and system, and let them freely evolve in the simulator. We observe many interesting user behaviors which are well aligned with the real human understandings. This paper makes a first step towards LLM-based simulation studies in the recommendation domain. We believe that there is much room left for improvement. To begin with, our simulator only has 10 users, which is still a toy model. In the future, one can introduce more users, and study how to design a tailored strategy to enhance the running efficiency. Then, one can also introduce more user behaviors other than interacting with the recommender system and social media, which may more comprehensively understand real user preferences.

## References

- [1] Brian Acosta, William Yang, and Michael Posa. Validating robotics simulators on real-world impacts. *IEEE Robotics and Automation Letters*, 7(3):6471–6478, 2022.
- [2] M Mehdi Afsar, Trafford Crump, and Behrouz Far. Reinforcement learning based recommender systems: A survey. *ACM Computing Surveys*, 55(7):1–38, 2022.
- [3] Kelsey R Allen, Tatiana Lopez Guevara, Yulia Rubanova, Kim Stachenfeld, Alvaro Sanchez-Gonzalez, Peter Battaglia, and Tobias Pfaff. Graph network simulators can learn discontinuous, rigid contact dynamics. In *Conference on Robot Learning*, pages 1157–1167. PMLR, 2023.
- [4] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.
- [5] Yukuo Cen, Jianwei Zhang, Xu Zou, Chang Zhou, Hongxia Yang, and Jie Tang. Controllable multi-interest framework for recommendation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2942–2951, 2020.
- [6] Jianxin Chang, Chen Gao, Yu Zheng, Yiqun Hui, Yanan Niu, Yang Song, Depeng Jin, and Yong Li. Sequential recommendation with graph neural networks. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*, pages 378–387, 2021.
- [7] Hongxu Chen, Yicong Li, Xiangguo Sun, Guandong Xu, and Hongzhi Yin. Temporal meta-path guided explainable recommendation. In *Proceedings of the 14th ACM international conference on web search and data mining*, pages 1056–1064, 2021.
- [8] Xinshi Chen, Shuang Li, Hui Li, Shaohua Jiang, Yuan Qi, and Le Song. Generative adversarial user model for reinforcement learning based recommendation system. In *International Conference on Machine Learning*, pages 1052–1061. PMLR, 2019.
- [9] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*, 2022.
- [10] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of*

---

<sup>4</sup><https://github.com/OpenBMB/AgentVerse>

- the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 639–648, 2020.
- [11] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural collaborative filtering. In *Proceedings of the Web Conference*, pages 173–182, 2017.
  - [12] Jin Huang, Harrie Oosterhuis, Maarten De Rijke, and Herke Van Hoof. Keeping dataset biases out of the simulation: A debiased simulator for reinforcement learning based recommender systems. In *Proceedings of the 14th ACM Conference on Recommender Systems*, pages 190–199, 2020.
  - [13] Eugene Ie, Chih-wei Hsu, Martin Mladenov, Vihan Jain, Sanmit Narvekar, Jing Wang, Rui Wu, and Craig Boutilier. Recsim: A configurable simulation platform for recommender systems. *arXiv preprint arXiv:1909.04847*, 2019.
  - [14] Xuanhat Lam, Thuc Vu, Trong Duc Le, and Anh Duc Duong. Addressing cold-start problem in recommendation systems. In *Proceedings of the 2nd international conference on Ubiquitous information management and communication*, pages 208–211, 2008.
  - [15] Xufang Luo, Zheng Liu, Shitao Xiao, Xing Xie, and Dongsheng Li. Mindsim: User simulator for news recommenders. In *Proceedings of the ACM Web Conference 2022*, pages 2067–2077, 2022.
  - [16] Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, et al. Isaac gym: High performance gpu-based physics simulation for robot learning. *arXiv preprint arXiv:2108.10470*, 2021.
  - [17] Masoud Mansoury, Himan Abdollahpouri, Mykola Pechenizkiy, Bamshad Mobasher, and Robin Burke. Feedback loop and bias amplification in recommender systems. In *Proceedings of the 29th ACM international conference on information & knowledge management*, pages 2145–2148, 2020.
  - [18] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022.
  - [19] Joon Sung Park, Joseph C O’Brien, Carrie J Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. Generative agents: Interactive simulacra of human behavior. *arXiv preprint arXiv:2304.03442*, 2023.
  - [20] Kanishka Rao, Chris Harris, Alex Irpan, Sergey Levine, Julian Ibarz, and Mohi Khansari. RL-cyclegan: Reinforcement learning aware simulation-to-real. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11157–11166, 2020.
  - [21] Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, et al. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*, 2022.
  - [22] Andrew I Schein, Alexandrin Popescul, Lyle H Ungar, and David M Pennock. Methods and metrics for cold-start recommendations. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 253–260, 2002.
  - [23] Jing-Cheng Shi, Yang Yu, Qing Da, Shi-Yong Chen, and An-Xiang Zeng. Virtual-taobao: Virtualizing real-world online retail environment for reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4902–4909, 2019.
  - [24] Rohan P Singh, Pierre Gergondet, and Fumio Kanehiro. mc-mujoco: Simulating articulated robots with fsm controllers in mujoco. In *2023 IEEE/SICE International Symposium on System Integration (SII)*, pages 1–5. IEEE, 2023.
  - [25] Harald Steck. Calibrated recommendations. In *Proceedings of the 12th ACM conference on recommender systems*, pages 154–162, 2018.
  - [26] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pages 1441–1450, 2019.

- [27] Juntao Tan, Shuyuan Xu, Yingqiang Ge, Yunqi Li, Xu Chen, and Yongfeng Zhang. Counterfactual explainable recommendation. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 1784–1793, 2021.
- [28] Qiaoyu Tan, Jianwei Zhang, Jiangchao Yao, Ninghao Liu, Jingren Zhou, Hongxia Yang, and Xia Hu. Sparse-interest network for sequential recommendation. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pages 598–606, 2021.
- [29] Ross Taylor, Marcin Kardas, Guillem Cucurull, Thomas Scialom, Anthony Hartshorn, Elvis Saravia, Andrew Poulton, Viktor Kerkez, and Robert Stojnic. Galactica: A large language model for science. *arXiv preprint arXiv:2211.09085*, 2022.
- [30] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.
- [31] Shiwen Wu, Fei Sun, Wentao Zhang, Xu Xie, and Bin Cui. Graph neural networks in recommender systems: a survey. *ACM Computing Surveys*, 55(5):1–37, 2022.
- [32] Dingqi Yang, Daqing Zhang, Zhiyong Yu, and Zhu Wang. A sentiment-enhanced personalized location recommendation system. In *Proceedings of the 24th ACM conference on hypertext and social media*, pages 119–128, 2013.
- [33] Mengyue Yang, Quanyu Dai, Zhenhua Dong, Xu Chen, Xiuqiang He, and Jun Wang. Top-n recommendation with counterfactual user preference simulation. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 2342–2351, 2021.
- [34] Yongfeng Zhang, Xu Chen, et al. Explainable recommendation: A survey and new perspectives. *Foundations and Trends® in Information Retrieval*, 14(1):1–101, 2020.
- [35] Yuke Zhu, Josiah Wong, Ajay Mandlekar, Roberto Martín-Martín, Abhishek Joshi, Soroush Nasiriany, and Yifeng Zhu. robosuite: A modular simulation framework and benchmark for robot learning. *arXiv preprint arXiv:2009.12293*, 2020.