A LLM-based Controllable, Scalable, Human-Involved User Simulator Framework for Conversational Recommender Systems

Anonymous Author(s)

Abstract

Conversational Recommender System (CRS) leverages real-time feedback from users to dynamically model their preferences, thereby enhancing the system's ability to provide personalized recommendations and improving the overall user experience. CRS has demonstrated significant promise, prompting researchers to concentrate their efforts on developing user simulators that are both more realistic and trustworthy. The emergence of Large Language Models (LLMs) has marked the onset of a new epoch in computational capabilities, exhibiting human-level intelligence in various tasks. Research efforts have been made to utilize LLMs for building user simulators to evaluate the performance of CRS. Although these efforts showcase innovation, they are accompanied by certain limitations. In this work, we introduce a **C**ontrollable, **S**calable, and **H**uman-**I**nvolved (CSHI) simulator framework that manages the behavior of user simulators across various stages via a plugin manager. CSHI customizes the simulation of user behavior and interactions to provide a more lifelike and convincing user interaction experience. Through experiments and case studies in two conversational recommendation scenarios, we show that our framework can adapt to a variety of conversational recommendation settings and effectively simulate users' personalized preferences. Consequently, our simulator is able to generate feedback that closely mirrors that of real users. This facilitates a reliable assessment of existing CRS studies and promotes the creation of high-quality conversational recommendation datasets.

CCS Concepts

• **Information systems → Recommender systems**.

Keywords

User Simulator, Conversational Recommender Systems, Large language models

1 Introduction

Recommender systems have significantly enhanced performance and customer satisfaction in various industries. Traditional recommender systems[1–3], however, primarily analyze users' historical behaviors to model their long-term preferences. This limitation makes it challenging for these systems to address two important questions[4]: what is the user's current preference, and which specific aspects of an item does the user favor? Conversational Recommender System (CRS)[5], typically comprising a dialogue module and a recommendation module, can capture not only the user's long-term preferences but also their real-time preferences. This is achieved through multiple rounds of natural language interactions with the user, enabling the system to make personalized recommendations in real-time. Conversational recommendation has shown considerable promise in daily interactions, where people often seek advice through conversations, such as inquiring about worthwhile movies, dining establishments worth trying, and recently appealing music. Despite the considerable advancements in the field of CRS, The high costs involved in training and evaluating these systems through real human conversations often lead researchers to use user simulators instead. In attribute-based CRS[6–10], the user simulator's responses are based on fixed templates, neglecting the flow of the conversation. In contrast, natural language processing (NLP)-based CRS[11–15] takes into account the flow of the conversation, but evaluations are based on fixed conversations, which may neglect the interactivity of conversational recommendations.

Large Language Models (LLMs)[16] have the capability to leverage their extensive world knowledge and commonsense reasoning for text understanding and generation[17–20], in some aspects approaching human-level intelligence[21–23]. The use of LLMs to simulate user behavior has demonstrated encouraging outcomes in areas such as collaboration work[24, 25] and recommendation systems[26–28]. Recent research have utilized LLMs as user simulators for assessing the performance of CRS[26, 29], overcoming the template-based limitations of previous research and demonstrating promising results. However, the majority of current studies in constructing user simulators offer session-level guidance to LLMs on generating responses via a single prompt template, which we refer to as 'single-prompt'. The limitations present in these studies have been analyzed in previous research[30], which can be succinctly summarized as data leakage occurs in the user simulator's feedback and controlling the output of the user simulator through a single prompt template proves challenging.

In this work, we propose a LLM-based Controllable, Scalable and Human-Involved (CSHI) simulator framework that manages the behavior of user simulators across various stages via a plugin manager. The controllability of CSHI is demonstrated by its precise ability to manage the behavior of the user simulator through a plugin-based design. The scalability of CSHI is evident in its modular design, allowing for the expansion or reduction of plugins and stages to accommodate personalized requirements. Given the complexity and variety of real-world situations, and the distinct operational modes these scenarios demand, the stages of CSHI are designed to be flexible and extendable. The human-involved characteristic allows humans to edit the profiles of the CSHI-based user simulator agents, influencing their behavior and enabling direct participation in interactions with the CRS. This facilitates the guidance of conversation topics, such as simulating a user's desire to find a related movie through a memory of a specific movie poster. CSHI customizes the simulation of user behavior and interactions to provide a more lifelike and convincing user interaction experience.

In summary, our contributions are as follows:

- We propose a LLM-based Controllable, Scalable and Human-Involved (CSHI) simulator framework that addresses key issues present in existing user simulators, such as data leakage and the difficulties associated with controlling feedback

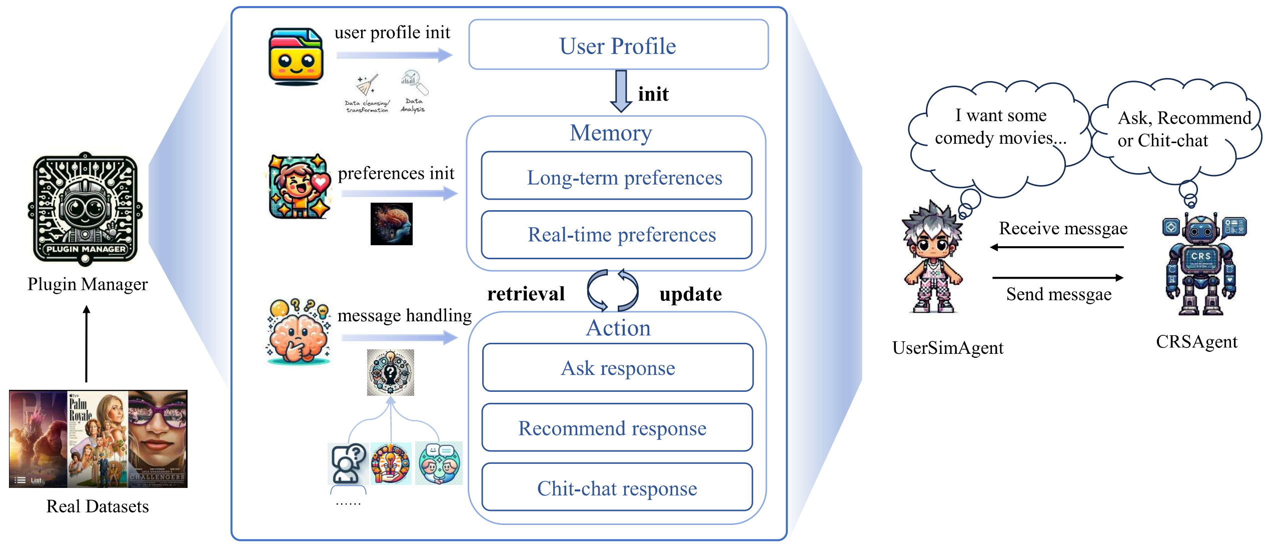


Figure 1: CSHI framework overview.

using a single prompt template. CSHI-based user simulator agents are more realistic and closely mirror real-world scenarios.

- Through experiments and case studies in two conversational recommendation scenarios, we show that CSHI-based user simulator agent can adapt to a variety of conversational recommendation settings and effectively simulate users' personalized preferences.
- Our simulator is expected to be used to generate high-quality conversational recommendation datasets, thereby advancing research in this field. We have implemented a web-based demo that allows real users to interact with our simulator. It can be accessed at https://github.com/zlxxlz1026/CSHI.

2 PROPOSED FRAMEWORK

The overall framework of CSHI is illustrated in Figure 1. CSHI manages the behavior of user simulator agents across various stages via a plugin manager. Plugins are modular components that can be easily added or removed, offering a plug-and-play functionality that allows for swift customization and adaptation to meet specific needs. This enables the simulator to flexibly handle a variety of complex scenarios and provide a more realistic user interaction experience. Within CSHI, the tasks of each stage are accomplished through the division and collaboration of various plugins. The execution of tasks can either be independently carried out by a single plugin or achieved through the cooperative effort of multiple plugins, among which priorities can be set as needed. In our initial setup, the construction of CSHI-based user simulator agent is divided into three stages: (1) User Profile Init; (2) Preferences Init; (3) Message Handling. In the first stage, User Profile Init, CSHI generates user profiles that are further utilized in the second stage, Preferences Init, to establish the agent's long-term preferences. During the second tage, Preferences Init, CSHI constructs the agent's preference-related memory, which encompasses both long-term and real-time preferences. In the third stage, Message Handling, the agent retrieves relevant preferences from its memory in response to messages from the CRS and executes corresponding actions. If there are updates to the preferences during this stage, these are recorded back into the memory.

Compared to existing user simulators, CSHI-based user simulator agent introduces three notable improvements: (1) To prevent the problem of data leakage in the feedback from the user simulator, CSHI-based agent adopts a strategy that simulates real user memory. Specifically, before recommending the target item, the agent's memory does not include the specific name of the target item; (2) To generate more realistic user feedback, we draw on the distinction between search and recommendation, to further refine real-time preferences into two categories: known preferences and unknown preferences. Known preferences refer to those that users can explicitly express, such as wanting to watch a comedy movie from the 90s. Unknown preferences, on the other hand, refer to those that users are not yet aware of or cannot explicitly express. These preferences need to be guided and elicited through information provided by the CRS. For example, a user may initially be unaware of their preference for a particular director's work, but when the CRS introduces relevant movie or director information (such as presenting movies directed by Stephen Chow), the user may then display a clear preference for that director. The discovery and clarification of such preferences occur gradually through the information guidance provided by the recommendation system and the process of user interaction. (3) Sensitive information is anonymized. Sensitive information refers to domain-specific details closely related to the item, such as the duration of a movie or its release date. Users rarely express their preferences in terms such as "I want to find a movie that is 132 minutes long" or "I want to watch a movie that was released on June 1, 2012";

Next, we will provide a phased overview of the plugins contained within each stage of the CSHI-based user simulator agent.

2.1 User Profile Init

User profiles play a crucial role in ensuring the consistency of agent behavior. The CSHI offers two methods for generating profiles at this stage: (1) Manual configuration: To enhance agent personalization, CSHI supports manual editing of profiles by users to design agents with distinct personalities. For instance, traits such as "You are very patient and will patiently answer every question" or "You are impatient and prefer to conclude conversations quickly" can be added to construct agents with varied personalities. Although manual editing of profiles offers high flexibility and can endow agents with any characteristics, it may become time-consuming and laborious when the requirements are complex and the agents are diverse. (2) LLMs generation: At this stage, plugins can automate the generation of different information in profiles, for example, extracting a user's long-term taste preferences based on their interaction history, or adding basic information such as the user's interaction history, gender, age, etc. This method can efficiently generate profiles, making it particularly suitable for scenarios that require generic information.

2.1.1 *User Preferences Summary Plugin:*

This plugin, inspired by Agent4Rec[28], employs a LLM to summarize users' personalized preferences and rating patterns in the movie domain from their interaction history. The approach involves categorizing movies rated 3 and above as "liked" by the user, while those rated below 3 are considered "disliked". This process can be represented as follows:

$$T_u = plugin_1(R_u) \tag{1}$$

Where R_u represents user u ratings for a collection of movies, and the output of the plugin is the personalized preferences T_u of user.

2.1.2 *Basic Information Configuration Plugin:*

This plugin processes raw data to generate basic configuration information for users, such as age, gender, interaction history, etc. This process can be described as follows:

$$P_u = plugin_2(RD_u) \tag{2}$$

Where RD_u represents the raw data of user u in the dataset, and the output of the plugin is the configuration information P_u of the user.

2.2 Preferences init

In recommendation systems, user preferences are typically categorized into long-term preferences and real-time preferences. Long-term preferences are modeled based on the user's interaction history, reflecting the user's consistent preferences; whereas real-time preferences denote the user's short-term inclinations, which may align with or differ from their long-term preferences. For example, a user who generally favors science fiction movies might suddenly wish to watch a comedy when feeling down. CRS can capture these real-time feedbacks through interactions with users, facilitating more accurate personalized recommendations. Therefore, at this stage, CSHI generates memories for the user simulator agent that include both long-term and real-time preferences, based on the user profile and target items. Real-time preferences involve information related to target items, while long-term preferences, shaped by the user profile, reflect the user's personalized characteristics.

2.2.1 *Real-Time Preference Generation Plugin:*

This plugin is tasked with generating real-time preferences for the agent during the initialization phase of the user simulator. Its design comprises two main components: the segmentation of real-time preferences and the anonymization of sensitive information. In terms of preference segmentation, the plugin differentiates between "known preferences" and "unknown preferences." Anonymization of sensitive information refers to the handling of domain-specific information related to items, such as changing a movie's specific release date from "June 1, 2012," to "the 2010s" or altering the runtime from "144 minutes" to "about 2 hours." This process can be summarized as follows:

$$real\_time\_preference = Plugin_3(Info_i, k_1, k_2) \tag{3}$$

Where $Info_i$ represents the relevant information generated by the LLMs for target items, such as the movie's director, actors, genre, and language. The hyperparameters k_1 and k_2 are used to adjust the proportion of known preferences and unknown preferences within the agent, respectively. This plugin generates the user's real-time preferences and integrates them into the agent's memory.

2.3 Message handling

This stage is focused on facilitating the interactions between the user simulator and the CRS, with the aim of producing responses that closely resemble human feedback. Upon receiving a message from the CRS, the agent emulates human cognitive processes, first understanding the intent of the CRS, and then executing corresponding actions based on the intent. Within our framework, for common conversational scenarios, the user simulator principally performs three types of actions: ask response, recommend response, and chit-chat response.

- Ask response: When the agent is asked about its preferences, it provides suitable answers by retrieving information from its memory.
- Recommend response: Faced with item recommendations, the agent gives feedback based on its own preferences.
- Chit-chat response: The agent can generate natural and smooth replies based on the current conversation topic and skillfully guide the conversation back to the recommendation context when needed.

2.3.1 *Intent Understanding Plugin:*

Upon receiving a message from the CRS, it discerns the system's intent, identifying whether the message is asking about the user's preferences, making a recommendation, or simply engaging in chit-chat. If the message involves an ask, the plugin further identifies the specific content of the ask, such as which aspect of preferences is being asked about. This process can be illustrated as follows:

$$intent, rel_{attr} = Plugin_4(message_{last}) \tag{4}$$

Where $message_{last}$ represents the lastest message received from the CRS, this plugin inputs the recognized intent and related queries rel_{attr} into the various response plugins.

2.3.2 *Ask Response Plugin - Personalized Reply:*

When the intent of the CRS is identified as 'ask', this plugin is prioritized to initiate, simulating the human memory retrieval process. It first utilizes a LLM to search the user's profile to determine if there exists a set of items relevant to the question asked by the CRS. If relevant items are found, the plugin combines the user's long-term and real-time preferences to generate a personalized reply, thereby concluding the ask response process. Conversely, if no relevant items are retrieved, the process moves to a non-personalized reply plugin. This process can be illustrated as follows: Where $profile_u$ represents the user's profile, and $flag$ is a Boolean indicator. If $flag$ is true, it signifies that the long-term preferences are unrelated to the current question, resulting in $ask_{response}$ being null, necessitating a transition to the non-personalized reply plugin. Conversely, $ask_{response}$ constitutes the agent's feedback to the CRS's question.

2.3.3 *Ask Response Plugin - Non-Personalized Reply:*

This plugin responds directly to the CRS's question based solely on the agent's real-time preferences. This process can be represented as follows:

$$ask_{response} = Plugin_6(intent, real\_time\_preference) \tag{5}$$

2.3.4 *Recommend Response Plugin:*

When the CRS's intent is identified as 'recommend', this plugin initially assesses whether to accept the recommendation. If accepted, it combines previous conversational history to provide positive feedback. If not accepted, the plugin further analyzes the reasons provided by the recommendation system for the recommendation, checking for information related to unknown preferences, If such information is found, it will not only offer negative feedback but also update the agent's real-time preferences based on this information, feeding back this new preference to the CRS. This process can be represented as follows:

$$rec_{response}, new\_preference = Plugin_7(intent, real\_time\_preference) \tag{6}$$

Where $rec_{response}$ represents the agent's feedback to the CRS's recommendation. If the real-time preferences are updated, then $new\_preference$ denotes the updated real-time preferences.

2.3.5 *Chit-Chat Response Plugin:*

When the intent of the CRS is identified as 'chit-chat', this plugin generates smooth and natural replies based on the conversational history. However, considering that the fundamental goal of conversational recommendation is to make recommendations aligned with the user's preferences, this plugin also seamlessly includes its own preferences when generating replies, thereby guiding the direction of the conversation closer to the ultimate recommendation goal. This process can be illustrated as follows:

$$chit_{response} = Plugin_8(intent, real\_time\_preference) \tag{7}$$

Where $chit_{response}$ represents the agent's chit-chat feedback to the CRS.

3 EXPERIMENTS

The workflow of the user simulator is illustrated in Figure 2. We conducted experiments in two conversational recommendation scenarios, taking into account the characteristics of the dataset.

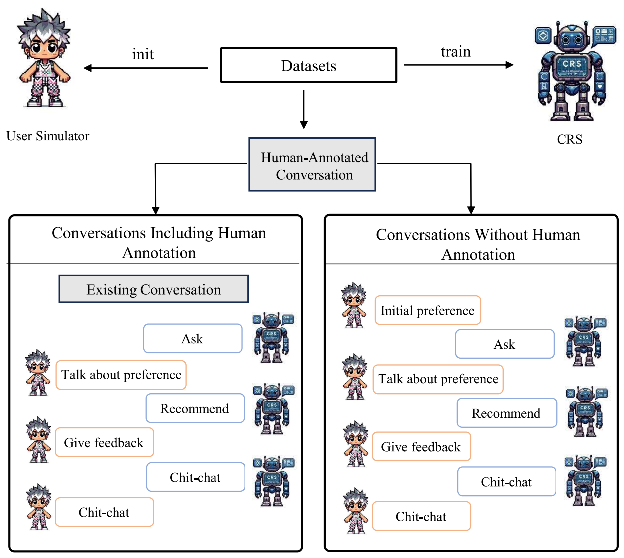


Figure 2: Workflow of the User Simulator.

The first scenario is built upon classic conversational recommendation datasets. These datasets typically contain human-annotated conversations and target item information. The user simulator utilizes the information related to the target item as real-time preferences, interacting with the CRS within predefined conversation contexts until the system successfully recommends an item or reaches the predetermined maximum number of interaction rounds. Here, the simulator's performance might be influenced by these annotated conversations. To address this, we introduced a scenario without annotated conversations, enabling the user simulator and the CRS to embark on entirely new conversations without any pre-established contexts, thus allowing for a more free-form interaction.

The second scenario is based on classic datasets from various fields, such as the MovieLens dataset in the movie recommendation domain. These datasets do not contain human-annotated conversations. The user simulator extracts long-term preferences based on the user's interaction history and treats the information related to the target item as real-time preferences. By integrating long-term and short-term preferences, profiles are generated to guide the user simulator in initiating new conversations with the CRS.

3.1 Conversational Recommendation Scenarios with Human Annotations

3.1.1 *Experimental setup.*

Our work builds upon iEvaLM[26]. For detailed information about our experimental settings, please refer to iEvaLM[1].

Dataset: We conduct experiments on two classic datasets in the conversational recommendation domain: ReDial[31] and OpenDialKG[32]. ReDial is a movie conversational recommendation dataset that contains 10,006 conversations. OpenDialKG is a multi-domain conversational recommendation dataset with 13,802 conversations, and only the movie domain was utilized in our experiments.

[1]https://github.com/RUCAIBox/iEvaLM-CRS/

Baselines: We conduct comparative experiments using three classical approaches in the field of conversational recommendations, as well as with a well-known large language model named ChatGPT:

- KBRD[11]: It utilizes an external Knowledge Graph (KG) to enhance the semantics of entities mentioned in the conversational history.
- BARCOR[13]: It proposes a unified CRS based on BART[33], which tackles two tasks using a single model.
- UniCRS[14]: It proposes a unified CRS model that leverages knowledge-enhanced prompt learning.
- ChatGPT [2]: We employ the publicly available model GPT-3.5-turbo-0613 provided by the OpenAI API.

Evaluation Metrics: Following existing work, we adopt Recall@k to evaluate the recommendation task. Similarly, we set $k = 1, 10, 50$, for both the ReDial and OpenDialKG datasets. Additionally, following existing work, we set the maximum number of interaction turns between the user simulator and the CRS at $t = 5$.

3.1.2 Experimental result. The results of the classic user simulator work, iEvaLM, have already been presented in the charts of reasearch work[30]. In this experiment, we solely provide the performance of various CRS methods when CSHI replaced iEvaLM in the construction of user simulator. Here, '-history' denotes that in our evaluation, conversations influenced by data leakage from conversational history, resulting in successful recommendations, are excluded from consideration. '-response' denotes that conversations affected by data leakage from the user simulator, which result in successful recommendations, are excluded from consideration. '-both' denotes that scenarios involving data leakage leading to successful recommendations in both aforementioned contexts are not considered in our evaluation. The evaluation results are presented in Table 1 and Figure 3. We can get the following observations:

- Data leakage from conversational history is an inherent problem of the dataset. However, we can look forward to the generation of more high-quality conversational recommendation data in the future through the interactions between user simulators and CRS.
- As shown in Table 1, CSHI-based agent significantly mitigates the issue of data leakage caused by user simulators. For instance, as illustrated in analysis work[30], when employing iEvaLM as the user simulator, various CRS methods experienced a decline in their recall@50 metric by 12.0%, 6.8%, 10.3%, and 17.9% on the ReDial dataset. In contrast, when utilizing CSHI to construct user simulator under the same experimental setup, there was no observable decrease in performance.
- As illustrated in Figure 3, in the '-both' scenario, CSHI-based agent demonstrates superior performance across multiple rounds of interactions (from the 2nd to the 5th round). This superior performance is attributable to the agent's ability to express its preferences in chit-chat conversational scenarios.

[2]https://chat.openai.com/

3.2 Conversational Recommendation Scenarios without Human Annotations

3.2.1 Experimental setup.
Dataset: We conducted experiments using the classic dataset in the movie recommendation domain, MovieLens. For each user, we selected their latest five interactions as the test set, while the remaining interaction history were utilized to generate user profiles.
Baselines: In our experiments, we observed an interesting phenomenon: in conversational recommendation scenarios with human annotations, the task of recommendation is often separate from conversation generation. Therefore, at times, even if the CRS generates target items that match user preferences within the conversation, the system might incorrectly deem the recommendation unsuccessful if the recommendation model does not successfully recommend the item. This is clearly unreasonable. Consequently, in conversational recommendation scenarios without human annotations, we adopted a generative CRS method based on LLMs[34].

The CRS agent comprises the Strategy Module, Memory Module, and Action Module: (1) The Strategy Module determines the actions to be taken based on the conversation history and historical decision-making behaviors stored in the Memory Module, as well as known user preferences. (2) The Memory Module stores the entire history of the conversation, historical decision-making behaviors, and information on user preferences for retrieval by the Strategy and Action Modules. (3) Actions comprise ask, recommend and chit-chat.
Evaluation Metrics: Given the challenge for generative CRS to recommend strictly based on item relevance, we utilize the recommendation accuracy at round t (success rate@t, SR@t) to measure the proportion of successful recommendations made by the CRS during the t-th interaction round in this scenario. We use the average number of turns (average turn, AT) to assess the average number of interactions between the user simulator and the CRS within a single conversation. Considering the context length limitations of generative CRS, we set the maximum number of recommended items generated by the CRS agent to 10. In the scenario without human annotations, we set the maximum number of interaction rounds between the user simulator and the conversational recommendation system to $T = 10$.

3.2.2 Experimental result. We conducted comparative and ablation experiments to evaluate the performance of CSHI, iEvaLM, and their variants. Here, "iEvaLM (+ui_info)" refers to the addition of target item information and user interaction history to the original prompt template. "CSHI (-filter)" indicates that the CSHI-based agent does not consider the anonymization of sensitive information in its processing. "-leakage" is used to denote the performance of the CRS without considering the impact of data leakage on successful conversations. The experimental results are presented in Table 2. Figure 4 displays the recommendation accuracy of the CRS under different user simulator configurations and across various rounds, disregarding successful conversations affected by data leakage. Through the analysis of Figure 4 and Table 2, we can draw the following conclusions:

- Compared to iEvaLM, when utilizing CSHI to construct user simulator, the CRS exhibits superior performance in

Table 1: Performance of existing CRS methods and ChatGPT under CSHI in various data leakage scenarios.

Model		KBRD			BARCOR			UniCRS			ChatGPT		
datasets		Recall@1	Recall@10	Recall@50	Recall@1	Recall@10	Recall@50	Recall@1	Recall@10	Recall@50	Recall@1	Recall@10	Recall@50
Redial	CSHI	0.027	0.170	0.445	0.033	0.196	0.476	0.064	0.266	0.564	0.209	0.445	0.668
	CSHI (-history)	0.009 (-66.7%)	0.116 (-31.8%)	0.387 (-13.0%)	0.034 (+0.03%)	0.185 (-5.6%)	0.446 (-6.3%)	0.024 (-62.5%)	0.206 (-22.6%)	0.529 (-6.2%)	0.183 (-12.4%)	0.417 (-6.3%)	0.654 (-2.1%)
	CSHI (-response)	0.027 (-0.0%)	0.170 (-0.0%)	0.445 (-0.0%)	0.033 (-0.0%)	0.196 (-0.0%)	0.476 (-0.0%)	0.064 (-0.0%)	0.264 (-0.8%)	0.564 (-0.0%)	0.209 (-0.0%)	0.445 (-0.0%)	0.668 (-0.0%)
	CSHI (-both)	**0.009 (-66.7%)**	**0.116 (-31.8%)**	**0.387 (-13.0%)**	**0.034 (+0.03%)**	**0.0185 (-5.6%)**	**0.446 (-6.3%)**	**0.024 (-62.5%)**	**0.204 (-23.3%)**	**0.529 (-6.2%)**	**0.183 (-12.4%)**	**0.417 (-6.3%)**	**0.654 (-2.1%)**
OpendialKG	CSHI	0.243	0.432	0.558	0.271	0.412	0.532	0.256	0.456	0.609	0.403	0.690	0.900
	CSHI (-history)	0.079 (-67.5%)	0.213 (-50.7%)	0.353 (-36.7%)	0.190 (-29.9%)	0.299 (-27.4%)	0.383 (-28.0%)	0.153 (-40.2%)	0.295 (-35.3%)	0.456 (-25.1%)	0.202 (-49.9%)	0.543 (-21.3%)	0.877 (-2.6%)
	CSHI (-response)	0.243 (-0.0%)	0.432 (-0.0%)	0.558 (-0.0%)	0.272 (+0.4%)	0.412 (-0.0%)	0.531 (-0.2%)	0.257 (+0.4%)	0.457 (+0.2%)	0.608 (-0.2%)	0.403 (-0.0%)	0.691 (+0.1%)	0.900 (-0.0%)
	CSHI (-both)	**0.079 (-67.5%)**	**0.213 (-50.7%)**	**0.353 (-36.7%)**	**0.191 (-29.5%)**	**0.298 (-27.7%)**	**0.380 (-28.6%)**	**0.153 (-40.2%)**	**0.295 (-35.3%)**	**0.455 (-25.3%)**	**0.203 (-49.6%)**	**0.544 (-21.2%)**	**0.877 (-2.6%)**

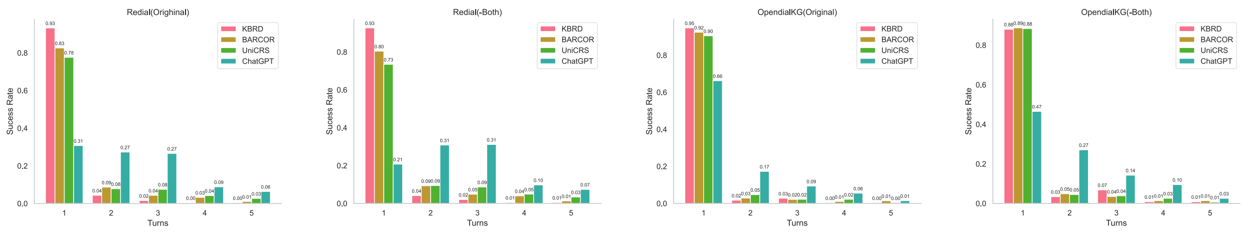


Figure 3: Percentage of successful recommendations by turn when constructing user simulators using CSHI.

Table 2: Performance of CRS under CSHI in conversational recommendation scenarios without Human Annotations.

	SR@3	SR@5	SR@10	AT
iEvaLM	0.434	0.680	0.742	5.384
iEvaLM (-leakage)	0.42 (-3.2%)	0.644 (-5.3%)	0.688 (-7.3%)	5.104
iEvaLM+ui_info	0.2	0.484	0.658	6.598
iEvaLM+ui_info (-leakage)	0.194 (-3.0%)	0.43 (-11.2%)	0.536 (-18.5%)	5.82
CSHI	0.634	0.776	0.906	4.434
CSHI (-leakage)	**0.63 (-0.6%)**	**0.772 (-0.5%)**	**0.902 (-0.4%)**	**4.422**
CSHI-filter	0.598	0.772	0.918	4.43
CSHI-filter (-leakage)	**0.592 (-1.0%)**	**0.766 (-0.8%)**	**0.908 (-1.1%)**	**4.386**

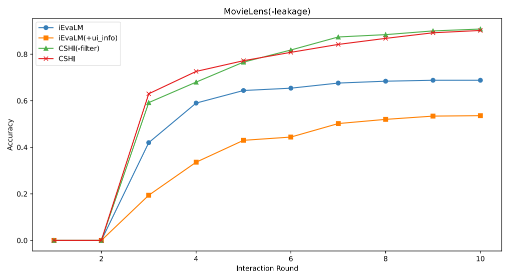


Figure 4: Recommendation accuracy across different rounds.

terms of both recommendation accuracy and average number of interaction turns. This advantage stems from the plugin-based working mechanism of CSHI, which renders the output of the user simulator agent more controllable. Meanwhile, single-prompt user simulators sometimes fail to generate feedback that meets expectations.

- Both CSHI and iEvaLM(+ui_info) incorporate the user's interaction history into their memory, yet iEvaLM(+ui_info) underperforms compared to iEvaLM. Upon case study, we hypothesize that this is due to the inclusion of the user's interaction history and target item information in the prompt template, which leads to an increase in template length and makes the output of the LLMs more challenging to control. A clear indication of this is that despite the prompt template explicitly stating not to mention the name of the target item, the issue of data leakage with iEvaLM(+ui_info) is more severe than with iEvaLM. This suggests that the addition of information may have a negative impact on the stability and controllability of the user simulator's output.
- Compared to CSHI, CSHI(-filter) demonstrates improved performance with an increase in the number of interaction turns. This phenomenon occurs because the CRS tends to request sensitive information such as release dates and durations to later stages. Consequently, if sensitive information is not anonymized, the CRS can leverage these detailed insights provided by the user simulator to make more accurate recommendations. However, this approach does not align with realistic conversational scenarios.

- As evident from Figure 4, the recommendation accuracy in the first two rounds is zero across different interactions between user simulators and the CRS. This indicates that the CRS tends to prefer detailed asks about user preferences in the initial stages to make rapid recommendations subsequently. In later interaction rounds, the recommendation accuracy of iEvaLM gradually stabilizes, suggesting that the feedback provided by iEvaLM offers limited assistance to the CRS as interactions progress. In contrast, CSHI-based agent exhibits a trend of consistent and steady improvement in recommendation accuracy with an increase in interaction rounds, indirectly highlighting CSHI's significant advantage in terms of controllability.

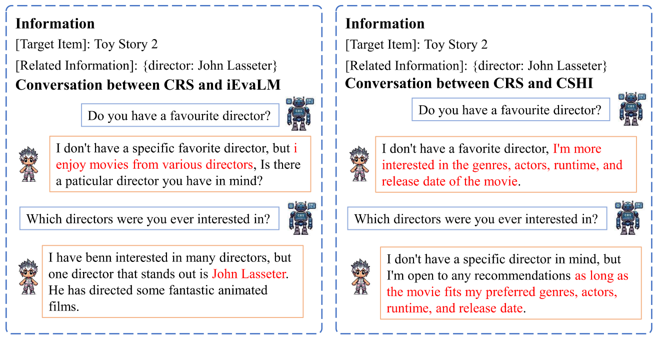
From the experimental results in two conversational recommendation scenarios, we observed that CSHI, compared to single-prompt user simulators, can generate more controllable feedback, thereby enhancing the performance of the CRS.

4 Case Study

Some advantages of CSHI are not readily apparent through quantitative experiments alone. In this section, we will showcase these benefits and discuss them in detail from the perspectives of controllability, scalability, and human involvement. The following cases are based on real interactions.

4.1 Controllability

The controllability of CSHI is primarily manifested in two aspects: the initialization of user preferences and the generation of feedback.



Information
[Target Item]: Toy Story 2
[Related Information]: {director: John Lasseter}
Conversation between CRS and iEvaLM

Do you have a favourite director?

I don't have a specific favorite director, but i enjoy movies from various directors, Is there a paticular director you have in mind?

Which directors were you ever interested in?

I have benn interested in many directors, but one director that stands out is John Lasseter. He has directed some fantastic animated films.

(a) Feedback from iEvaLM

Information
[Target Item]: Toy Story 2
[Related Information]: {director: John Lasseter}
Conversation between CRS and CSHI

Do you have a favourite director?

I don't have a favorite director, I'm more interested in the genres, actors, runtime, and release date of the movie.

Which directors were you ever interested in?

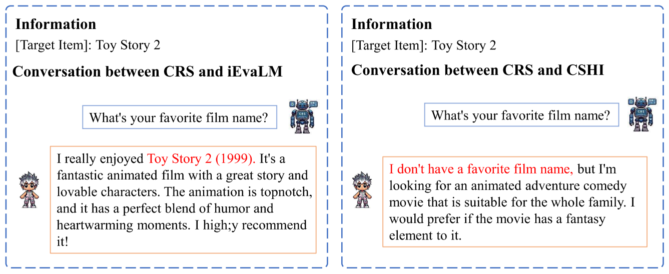
I don't have a specific director in mind, but I'm open to any recommendations as long as the movie fits my preferred genres, actors, runtime, and release date.

(b) Feedback from CSHI

Figure 5: Feedback from Different Simulators.

4.1.1 Feedback Generation.

From Figure 5, it can be observed that when iEvaLM is asked about preferences related to directors, based on its real-time preferences, it might provide information relevant to the question or might not, with no control over which specific information should be recognized as known preferences and supplied to the CRS. In contrast, CSHI is acutely aware of its real-time preferences. Therefore, when faced with asks about director preferences, CSHI will explicitly indicate a lack of preferences related to directors, emphasizing its focus on aspects such as movie genres and actors instead.



Information
[Target Item]: Toy Story 2
Conversation between CRS and iEvaLM

What's your favorite film name?

I really enjoyed Toy Story 2 (1999). It's a fantastic animated film with a great story and lovable characters. The animation is topnotch, and it has a perfect blend of humor and heartwarming moments. I high;y recommend it!

(a) Data leakage from iEvaLM

Information
[Target Item]: Toy Story 2
Conversation between CRS and CSHI

What's your favorite film name?

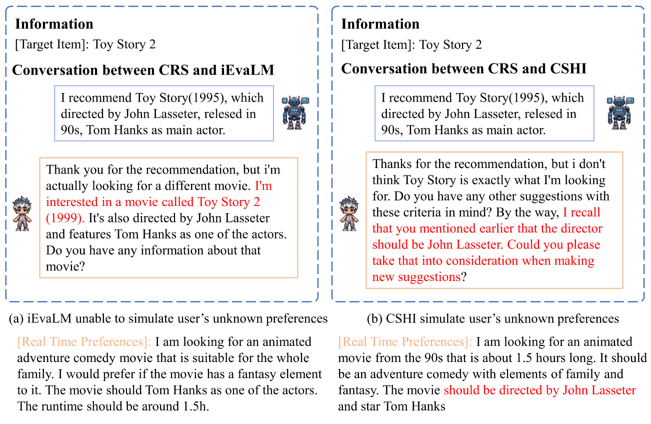
I don't have a favorite film name, but I'm looking for an animated adventure comedy movie that is suitable for the whole family. I would prefer if the movie has a fantasy element to it.

(b) No data leakage in CSHI

Figure 6: Data leakage from user simulators.

4.1.2 Data Leakage.

From Figure 6, it can be observed that even though iEvaLM's prompt template explicitly instructs to avoid disclosing the name of the target items, inadvertent data leakage can still occur due to the inclusion of the target item's name in the template itself. In contrast, as CSHI's memory does not contain any names of target items, it effectively prevents the leakage of target item names.



Information
[Target Item]: Toy Story 2
Conversation between CRS and iEvaLM

I recommend Toy Story(1995), which directed by John Lasseter, relesed in 90s, Tom Hanks as main actor.

Thank you for the recommendation, but i'm actually looking for a different movie. I'm interested in a movie called Toy Story 2 (1999). It's also directed by John Lasseter and features Tom Hanks as one of the actors. Do you have any information about that movie?

(a) iEvaLM unable to simulate user's unknown preferences

[Real Time Preferences]: I am looking for an animated adventure comedy movie that is suitable for the whole family. I would prefer if the movie has a fantasy element to it. The movie should Tom Hanks as one of the actors. The runtime should be around 1.5h.

(a) Real-time preferences before update

Information
[Target Item]: Toy Story 2
Conversation between CRS and CSHI

I recommend Toy Story(1995), which directed by John Lasseter, relesed in 90s, Tom Hanks as main actor.

Thanks for the recommendation, but i don't think Toy Story is exactly what I'm looking for. Do you have any other suggestions with these criteria in mind? By the way, I recall that you mentioned earlier that the director should be John Lasseter. Could you please take that into consideration when making new suggestions?

(b) CSHI simulate user's unknown preferences

[Real Time Preferences]: I am looking for an animated movie from the 90s that is about 1.5 hours long. It should be an adventure comedy with elements of family and fantasy. The movie should be directed by John Lasseter and star Tom Hanks

(b) Real-time preferences after update

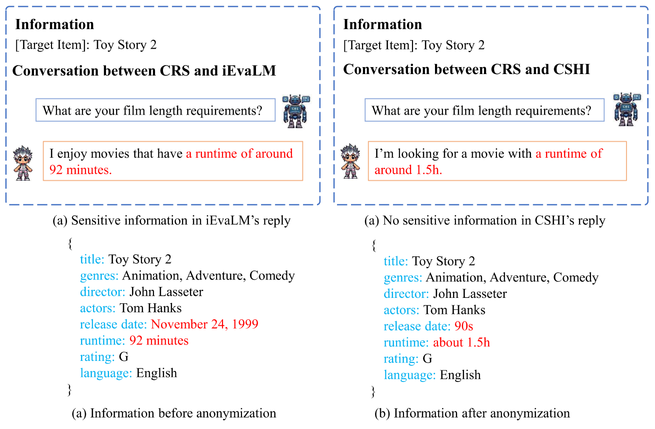
Figure 7: Unknown preferences from user Simulators.

4.1.3 Unknown Preferences.

As demonstrated in the Figure 7, single-prompt user simulators struggle to effectively simulate users' latent preferences. The prevalent approach is to provide information about the target item while rejecting the recommendation list. In contrast, when the CRS provides information related to unknown preferences, CSHI responds positively and updates its real-time preferences.

4.2 Scalability

The scalability of CSHI is demonstrated by its ability to meet specific needs through the design of dedicated plugins, such as information anonymization and generating personalized responses by incorporating user interaction history.



Information
[Target Item]: Toy Story 2
Conversation between CRS and iEvaLM

What are your film length requirements?

I enjoy movies that have a runtime of around 92 minutes.

(a) Sensitive information in iEvaLM's reply

Information
[Target Item]: Toy Story 2
Conversation between CRS and CSHI

What are your film length requirements?

I'm looking for a movie with a runtime of around 1.5h.

(a) No sensitive information in CSHI's reply

```
{
  title: Toy Story 2
  genres: Animation, Adventure, Comedy
  director: John Lasseter
  actors: Tom Hanks
  release date: November 24, 1999
  runtime: 92 minutes
  rating: G
  language: English
}
```

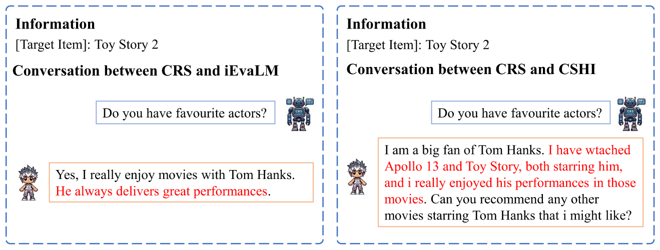
(a) Information before anonymization

```
{
  title: Toy Story 2
  genres: Animation, Adventure, Comedy
  director: John Lasseter
  actors: Tom Hanks
  release date: 90s
  runtime: about 1.5h
  rating: G
  language: English
}
```

(b) Information after anonymization

Figure 8: Comparison before and after information anonymization.

4.2.1 Anonymization of Sensitive Information. From Figure 8, it is evident that without the anonymization of sensitive information, the user simulator would output domain-specific information related to items, such as a movie's release date. However, in real-life scenarios, it is uncommon for users to express their preferences in such specific terms as "I want to watch a movie released on June 1, 2012."



Information
[Target Item]: Toy Story 2
Conversation between CRS and iEvaLM

Do you have favourite actors?

Yes, I really enjoy movies with Tom Hanks. He always delivers great performances.

Information
[Target Item]: Toy Story 2
Conversation between CRS and CSHI

Do you have favourite actors?

I am a big fan of Tom Hanks. I have wtached Apollo 13 and Toy Story, both starring him, and i really enjoyed his performances in those movies. Can you recommend any other movies starring Tom Hanks that i might like?

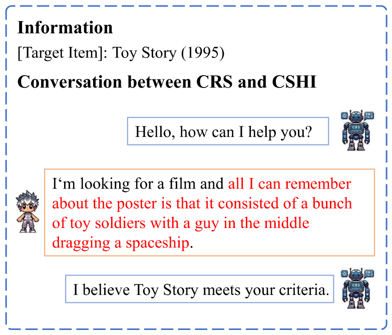
(a) iEvaLM can only give reply based on basic information (b) CSHI can generate personalized reply

Figure 9: User simulator's personalized reply.

4.2.2 Personalized Reply. In real-world scenarios, users often favor certain items because they have interacted with similar items previously. Hence, we implemented a plugin that can generate personalized responses by incorporating users' interaction histories. From Figure 9, it is observable that since different users have distinct interaction histories, the feedback generated, even for the same theme (such as directors), is personalized. iEvaLM struggles to achieve this level of personalization.

4.3 Human Involvement

In CSHI, humans can edit the configuration profile of the user simulator agent, thereby influencing the agent's behavior, and can directly engage in interactions with the CRS to guide the conversation topic. This is exemplified by a user wanting to find a related movie through a memory associated with a movie poster.



Information
[Target Item]: Toy Story (1995)
Conversation between CRS and CSHI

Hello, how can I help you?

I'm looking for a film and all I can remember about the poster is that it consisted of a bunch of toy soldiers with a guy in the middle dragging a spaceship.

I believe Toy Story meets your criteria.

Figure 10: Human involvement in the conversation.

As Figure 10 illustrated, human participants can act as the user's memory by incorporating relevant information about a poster into the user simulator's profile, or they can directly participate in the conversation by providing partial poster information during the interaction, guiding the CRS towards successful recommendations.

5 CONCLUSION

In this paper, wo introduced a LLM-based Controllable, Scalable, Human-Involved user simulator framework that manages the behavior of user simulators across various stages via a plugin manager. CSHI customizes the simulation of user behavior and interactions to provide a more lifelike and convincing user interaction experience. Through experiments and case studies in two conversational recommendation scenarios, we show that our framework can adapt to a variety of conversational recommendation settings and effectively simulate users' personalized preferences. CSHI is expected to be used to generate high-quality conversational recommendation datasets, thereby advancing research in this field.

Currently, we have only attempted to manually input visual features, such as movie posters, into agents. Given the scalability of the CSHI, it can conveniently utilize information from various modalities, including visual, to construct user simulation agents. This approach allows for a closer approximation to the behaviors of real users.

References

[1] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*, pages 173–182, 2017.
[2] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.
[3] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. Wide & deep learning for recommender systems. In *Proceedings of the 1st workshop on deep learning for recommender systems*, pages 7–10, 2016.
[4] Chongming Gao, Wenqiang Lei, Xiangnan He, Maarten de Rijke, and Tat-Seng Chua. Advances and challenges in conversational recommender systems: A survey. *AI open*, 2:100–126, 2021.
[5] Yueming Sun and Yi Zhang. Conversational recommender system. In *The 41st international acm sigir conference on research & development in information retrieval*, pages 235–244, 2018.
[6] Wenqiang Lei, Xiangnan He, Yisong Miao, Qingyun Wu, Richang Hong, Min-Yen Kan, and Tat-Seng Chua. Estimation-action-reflection: Towards deep interaction between conversational and recommender systems. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, pages 304–312, 2020.
[7] Wenqiang Lei, Gangyi Zhang, Xiangnan He, Yisong Miao, Xiang Wang, Liang Chen, and Tat-Seng Chua. Interactive path reasoning on graph for conversational

recommendation. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2073–2083, 2020.

[8] Yang Deng, Yaliang Li, Fei Sun, Bolin Ding, and Wai Lam. Unified conversational recommendation policy learning via graph-based reinforcement learning. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1431–1441, 2021.

[9] Kerui Xu, Jingxuan Yang, Jun Xu, Sheng Gao, Jun Guo, and Ji-Rong Wen. Adapting user preference to online feedback in multi-round conversational recommendation. In *Proceedings of the 14th ACM international conference on web search and data mining*, pages 364–372, 2021.

[10] Mengyuan Zhao, Xiaowen Huang, Lixi Zhu, Jitao Sang, and Jian Yu. Knowledge graph-enhanced sampling for conversational recommendation system. *IEEE Transactions on Knowledge and Data Engineering*, 2022.

[11] Qibin Chen, Junyang Lin, Yichang Zhang, Ming Ding, Yukuo Cen, Hongxia Yang, and Jie Tang. Towards knowledge-based recommender dialog system. *arXiv preprint arXiv:1908.05391*, 2019.

[12] Kun Zhou, Wayne Xin Zhao, Shuqing Bian, Yuanhang Zhou, Ji-Rong Wen, and Jingsong Yu. Improving conversational recommender systems via knowledge graph based semantic fusion. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 1006–1014, 2020.

[13] Ting-Chun Wang, Shang-Yu Su, and Yun-Nung Chen. Barcor: Towards a unified framework for conversational recommendation systems. *arXiv preprint arXiv:2203.14257*, 2022.

[14] Xiaolei Wang, Kun Zhou, Ji-Rong Wen, and Wayne Xin Zhao. Towards unified conversational recommender systems via knowledge-enhanced prompt learning. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1929–1937, 2022.

[15] Zujie Liang, Huang Hu, Can Xu, Jian Miao, Yingying He, Yining Chen, Xiubo Geng, Fan Liang, and Daxin Jiang. Learning neural templates for recommender dialogue system. *arXiv preprint arXiv:2109.12302*, 2021.

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

[17] Sunhao Dai, Ninglu Shao, Haiyuan Zhao, Weijie Yu, Zihua Si, Chen Xu, Zhongxiang Sun, Xiao Zhang, and Jun Xu. Uncovering chatgpt's capabilities in recommender systems. In *Proceedings of the 17th ACM Conference on Recommender Systems*, pages 1126–1132, 2023.

[18] Yunfan Gao, Tao Sheng, Youlin Xiang, Yun Xiong, Haofen Wang, and Jiawei Zhang. Chat-rec: Towards interactive and explainable llms-augmented recommender system. *arXiv preprint arXiv:2303.14524*, 2023.

[19] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

[20] Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems*, 36, 2024.

[21] Abulhair Saparov, Richard Yuanzhe Pang, Vishakh Padmakumar, Nitish Joshi, Mehran Kazemi, Najoung Kim, and He He. Testing the general deductive reasoning capacity of large language models using ood examples. *Advances in Neural Information Processing Systems*, 36, 2024.

[22] Zhiheng Xi, Wenxiang Chen, Xin Guo, Wei He, Yiwen Ding, Boyang Hong, Ming Zhang, Junzhe Wang, Senjie Jin, Enyu Zhou, et al. The rise and potential of large language model based agents: A survey. *arXiv preprint arXiv:2309.07864*, 2023.

[23] Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, et al. A survey on large language model based autonomous agents. corr abs/2308.11432 (2023), 2023.

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

[25] Xinyi Mou, Zhongyu Wei, and Xuanjing Huang. Unveiling the truth and facilitating change: Towards agent-based large-scale social movement simulation. *arXiv preprint arXiv:2402.16333*, 2024.

[26] Xiaolei Wang, Xinyu Tang, Wayne Xin Zhao, Jingyuan Wang, and Ji-Rong Wen. Rethinking the evaluation for conversational recommendation in the era of large language models. *arXiv preprint arXiv:2305.13112*, 2023.

[27] Lei Wang, Jingsen Zhang, Xu Chen, Yankai Lin, Ruihua Song, Wayne Xin Zhao, and Ji-Rong Wen. Recagent: A novel simulation paradigm for recommender systems. *arXiv preprint arXiv:2306.02552*, 2023.

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

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

[30] Lixi Zhu, Xiaowen Huang, and Jitao Sang. How reliable is your simulator? analysis on the limitations of current llm-based user simulators for conversational recommendation. *arXiv preprint arXiv:2403.16416*, 2024.

[31] Raymond Li, Samira Ebrahimi Kahou, Hannes Schulz, Vincent Michalski, Laurent Charlin, and Chris Pal. Towards deep conversational recommendations. *Advances in neural information processing systems*, 31, 2018.

[32] Seungwhan Moon, Pararth Shah, Anuj Kumar, and Rajen Subba. Opendialkg: Explainable conversational reasoning with attention-based walks over knowledge graphs. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pages 845–854, 2019.

[33] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*, 2019.

[34] Lei Li, Yongfeng Zhang, Dugang Liu, and Li Chen. Large language models for generative recommendation: A survey and visionary discussions. *arXiv preprint arXiv:2309.01157*, 2023.