# LLM-Powered User Simulator for Recommender System

**Zijian Zhang[1], Shuchang Liu[2], Ziru Liu[3], Rui Zhong[2], Qingpeng Cai[2] *, Xiangyu Zhao[3] *, Chunxu Zhang[1] *, Qidong Liu[3, 4], Peng Jiang[2]**

[1] Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University
[2] Kuaishou Technology
[3] City University of Hong Kong
[4] Xi'an Jiaotong University
{zhangzijian,zhangchunxu}@jlu.edu.cn, {liushuchang, zhongrui}@kuaishou.com, ziruliu2-c@my.cityu.edu.hk, cqpcurry@gmail.com, xianzhao@cityu.edu.hk, liuqidong@stu.xjtu.edu.cn, jp2006@139.com

## Abstract

User simulators can rapidly generate a large volume of timely user behavior data, providing a testing platform for reinforcement learning-based recommender systems, thus accelerating their iteration and optimization. However, prevalent user simulators generally suffer from significant limitations, including the opacity of user preference modeling and the incapability of evaluating simulation accuracy. In this paper, we introduce an LLM-powered user simulator to simulate user engagement with items in an explicit manner, thereby enhancing the efficiency and effectiveness of reinforcement learning-based recommender systems training. Specifically, we identify the explicit logic of user preferences, leverage LLMs to analyze item characteristics and distill user sentiments, and design a logical model to imitate real human engagement. By integrating a statistical model, we further enhance the reliability of the simulation, proposing an ensemble model that synergizes logical and statistical insights for user interaction simulations. Capitalizing on the extensive knowledge and semantic generation capabilities of LLMs, our user simulator faithfully emulates user behaviors and preferences, yielding high-fidelity training data that enrich the training of recommendation algorithms. We establish quantifying and qualifying experiments on five datasets to validate the simulator's effectiveness and stability across various recommendation scenarios.

**Code** — https://github.com/Applied-Machine-Learning-Lab/LLM_User_Simulator

## Introduction

Reinforcement Learning (RL)-based recommender systems have become increasingly important due to their capability to capture user preferences (Zhao et al. 2021a, 2023b; Liu et al. 2023b; Zhang et al. 2020) and long-term engagement (Liu et al. 2023a; Zhao et al. 2018a,b, 2019). They require interactive training where agent learns to make decisions by interacting with an environment. Online data reflects the real-time user feedback and behavioral patterns, which is critical for continuously refining recommender systems and solving real-world problems (Li et al. 2023; Han et al. 2024; Zhang et al. 2023, 2024c). However, due to the difficulty in obtaining online user interaction data, high collection costs, and user privacy concerns (Wang et al. 2023a), effectively simulating user interaction behavior has become an urgent problem to be solved. User simulators can quickly generate user behavior data, thus accelerating the evaluation process, and they guarantee user privacy without collection and use of real user data (Zhao et al. 2021b).

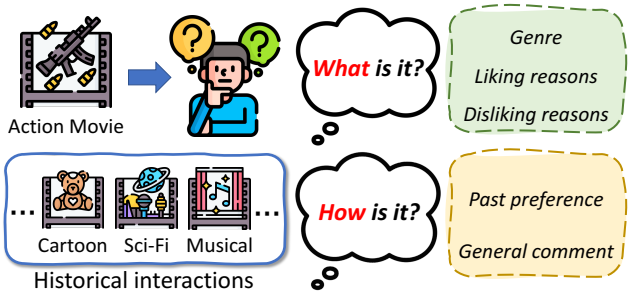


Figure 1: The user interaction logic for recommended items includes two problems: understanding what it is and how it is, given the historical interactions.

Despite the promising progress of user simulators for recommender systems (Wang et al. 2023a; Ie et al. 2019; Corecco et al. 2024; Zhang et al. 2024a; Huang et al. 2024), existing research has two principal deficiencies. Firstly, current simulators fail to explicitly model user preferences, which is a critical function for accurately predicting user choices. VirtualTaobao (Shi et al. 2019) leverages GAN to simulate user interaction distribution and KuaiSim (Zhao et al. 2023a) employs an offline trained transformer to emulate user's responses to recommendations. Secondly, there is an absence of an efficient evaluative framework to assess the fidelity of simulated interactions with real user behavior. Consequently, there is an urgent need for the advancement of user simulators that can operate with higher degrees of fidelity and transparency in replicating the complex dynamics of user-system interactions.

Recently, Large Language Models (LLMs) have demonstrated remarkable effectiveness in maintaining common knowledge and inferential capabilities across a spectrum of fields (Jia et al. 2023; Fu et al. 2023; Wang et al. 2024a,b; Liu et al. 2024a,b). This prowess positions LLMs as a

*Corresponding authors: Qingpeng Cai, Xiangyu Zhao, and Chunxu Zhang.

promising avenue for simulating user behavior in recommendation systems. SUBER (Corecco et al. 2024) harnesses the semantic comprehension capabilities of LLMs to directly infer user engagement with items. Agent4Rec (Zhang et al. 2024a) equips user simulation by integrating LLM agents. It supports a wide range of behaviors, including taste-driven actions, such as viewing and rating items, as well as emotion-driven actions, including exiting the system and feedback through comments. However, the application of LLMs in user simulation is not without drawbacks. Primarily, the computational and time cost of calling LLM for simulations is substantial, posing a significant barrier to the training of the recommendation system (Corecco et al. 2024; Huang et al. 2024). What's more, an overreliance on LLMs also introduces the risk of hallucination, where the model generates factually incorrect or misleading inferences (Ji et al. 2023; Yao et al. 2023). These challenges must be navigated carefully to harness LLMs in crafting user simulators that are both efficient and grounded in reality.

In this paper, we first figure out a fundamental logic governing user interactions with recommended items, as illustrated in Figure 1. Taking movie recommendations as an example, a user's engagement with a recommended action movie begins with recognizing **what** the movie is, *i.e.,* identifying its genre and characteristics that might lead to a liking or disliking sentiment. This recognition is followed by a deeper inquiry into **how** the item aligns with the user's tastes, *i.e.,* analyzing his past preference to this genre, and considering the general comments to this movie. Based on item characteristics and the user's preference information, our goal is to explicitly utilize the fundamental logic of user interactions and predict their behavior towards an item in a transparent and understandable manner.

It is nontrivial to achieve this goal, and several main challenges must be conquered. First, **how to depict the user preference explicitly?** Translating user preferences into a clear and understandable model is inherently difficult. Existing deep collaborative filtering models with trainable embedding are hardly explainable, let alone indicating user preference. To address this, we propose leveraging LLM to analyze and interpret item characteristics from various angles, including genre, potential likes and dislikes, and user sentiments. By harnessing the rich analytical capabilities of LLM, we develop a logical model that captures the nuanced decision-making processes underlying user interactions.

Second, **how to alleviate computational cost and the hallucination issue when using LLM?** The computational demands of LLM and the risk of hallucination present substantial hurdles in system reliability (Ji et al. 2023). Instead of inferring the user interaction directly based on LLM (Zhang et al. 2024a; Corecco et al. 2024), we use LLM to distill the reasons a user might like or dislike an item, condensing and filtering this reasoning into concise keywords that minimize the impact of outliers. In addition, we complement this with a statistical model, *i.e.,* a trained sequential model, that provides a regularizing effect on the predictions. Our ensemble model integrates the strengths of both logical reasoning and statistical learning, enhancing the reliability and efficiency of user interaction simulations synergistically.

Third, **how to evaluate the user simulator?** Lacking a standardized metric for evaluation, we face the challenge of assessing the effectiveness of our user simulator. To tackle this, we establish a series of experiments that span a wide range of applications, *i.e.,* five public datasets encompassing POI, music, movie, game, and anime recommendation, to ensure that our tests are generalizable. By training reinforcement learning algorithms within these domains and comparing their performance, we aim to validate the simulator's ability to provide meaningful insights into user behavior within recommendation systems.

The main contributions of this paper are as follows:

- We identify the intrinsic logic governing user engagement with items in recommendation, and propose a logical model that explicitly infers user interaction.
- We construct an ensemble model consisting of rule-based logical and data-driven statistical models to imitate human interaction, maintaining consistency and reducing the likelihood of erroneous inferences.
- We conduct both qualifying and quantifying experiments on five benchmark datasets that span a variety of application fields to verify the efficacy of the proposed method.

# Methodology

## Preliminaries

***User Simulator.*** User simulator in recommendation aims to infer the engagement as a real human user with candidate item $i_c$ offered by recommender systems. It can be formulated as $P(y_c|h, i_c)$, where $y_c$ represents the interaction given by the user simulator, such as rating, buy, and dislike. In this paper, we consider like and dislike for clear descriptions. $h = \{(i_1, y_1), \ldots, (i_n, y_n)\}$ is the user's historical interactions, consisting of his past responses of $n$ items.

## Framework Overview

In this section, we implement the user engagement process with recommended items. The workflow of our system is depicted in Figure 2. The sequence of operations proceeds from left to right, beginning with an analysis of the potential item, followed by a comparison with the user's past interactions, culminating in a reasoned inference.

To express the logic of user interactions in a clear way, we use LLM to examine items and produce informative descriptions. We extract keywords that indicate why items might be liked or disliked based on both the item's features and user reviews. We also create an ensemble model consisting of three base models. This model suite evaluates the user's past preferences and the current item to produce an inference result. By combining these models, we aim to deliver a more precise and dependable assessment.

## Objective Item Description Collection

To discern **what** the item is, we initially leverage the items' factual descriptions. Our approach is to pinpoint the category $D_{cate}$ and delineate objective reasons for liking or disliking the item, *i.e.,* $D_{pos}$ and $D_{neg}$. This objective item

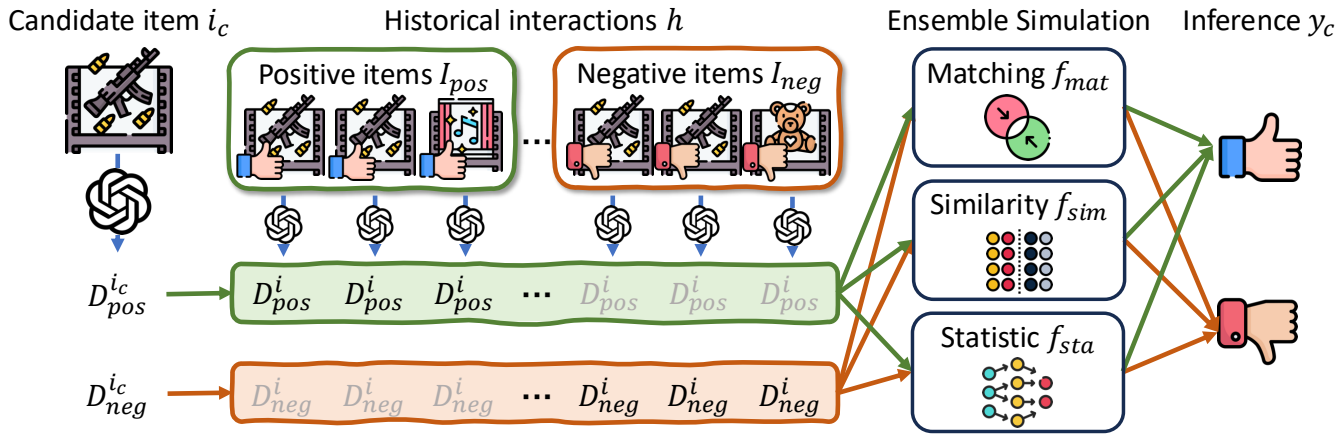


Figure 2: User interaction simulation pipeline. We use an LLM to generate preference keywords for items. For a candidate item $i_c$, we separate user historical interactions $h$ into positive and item sets $I_{pos}$ and $I_{neg}$, and apply three base models, *i.e.,* two logical models $f_{mat}$, $f_{sim}$, and one statistical model $f_{sta}$, to evaluate $i_c$'s pros and cons, *i.e.,* $D^{i_c}_{pos}$ and $D^{i_c}_{neg}$, against the historical item pros and cons. We ensemble the three base models' output as the final interaction inference $y_c$.

description analysis contributes to a comprehensive understanding of the item's attributes and potential user reactions.

For item categorization $D_{cate}$, we extract textual genre descriptions from the raw data, filtering out the most and least frequent categories. This process ensures that each item is assigned a final category that balances specificity and relevance, avoiding both over-generalization and excessive fragmentation of category representation.

To attain the potential reasons behind user preferences for items $D_{pos}$ and $D_{neg}$, we start with the fundamental details such as the item's name, category, and attributes. Employing an LLM, we generate objective descriptions that encompass the possible factors influencing a user's affinity or aversion towards the item. Take the movie recommendation as example again, we craft a prompt template $T_{obj}$ that integrates the available information. $T_{obj}$ is shown as follows, where NAME, ATTRIBUTES, and CATEGORIES will be filled with item information from raw data, and the response will fill the placeholder in red text:

**Objective Description Prompt Template $T_{obj}$**

Given a Movie named NAME and its characteristics, I need you to provide pros and cons, corresponding evidence and keywords from a customer perspective. Movie has the following attributes: ATTRIBUTES. Categories of the movie include CATEGORIES. Reasons, keywords, and evidence should be concise and reasonable. Keywords should appear positive or negative. Evidence should refer to the information given. Strictly follow the reply format, fill [], do not say anything else: Pros 1: [pros 1] Evidence: [evidence of pros 1] Keywords: [keywords of pros 1] Pros 2: [pros 2] Evidence: [evidence of pros 2] Keywords: [keywords of pros 2] Pros 3: [pros 3] Evidence: [evidence of pros 3] Keywords: [keywords of pros 3] Cons 1: [cons 1] Evidence: [evidence of cons 1] Keywords: [keywords of cons 1] Cons 2: [cons 2] Evidence: [evidence of cons 2] Keywords: [keywords of cons 2] Cons 3: [cons 3] Evidence: [evidence of cons 3] Keywords: [keywords of cons 3]

Given the LLM's response according to the form of $T_{obj}$, we extract the keywords that represent positive aspects as $D^{obj}_{pos}$ and those that signify negative aspects as $D^{obj}_{neg}$. Consequently, $D^{obj}_{pos}$ and $D^{obj}_{neg}$ emerge as curated lists of keywords, effectively encapsulating the essence of users' potential attitudes towards items. These keyword sets serve as concise, informative indicators, reflecting the likely inclinations of user preferences or aversions in a distilled form.

When harnessing the power of LLMs to generate reasons for potential user likes and dislikes, we integrate the Chain of Thoughts (CoT) (Wei et al. 2022; Chu et al. 2023) approach to bolster the precision and reliability of the LLM's output. In particular, we guide the LLM to first identify the concrete reasons behind the likes and dislikes using the provided keywords and then summarize the reasons into concise keywords. Finally, we require LLM to substantiate these reasons with evidence drawn from the information provided. This process helps circumvent the risk of hallucination inherent in LLMs, ensuring that the generated reasons are well-founded and relevant.

This prompt is engineered to inspire a detailed analysis from the LLM regarding the item's factual characteristics. It encourages the model to consider the full spectrum of attributes from the most appealing features to the possible shortcomings. By demanding a logical sequence of identification and justification, the LLM's output is not only informative but also transparent in its reasoning, aligning closely with how human users might evaluate the item.

## Subjective Item Description Collection

To comprehend **how** a user views an item, public opinions can greatly influence the decision-making process. For example, a considerable proportion of users tend to choose

mainstream options. Hence, we examine the comments to find keywords that indicate liking or disliking sentiments.

To accomplish this, we devise a prompt template $T_{sub}$ that encompasses the item's descriptions and the user's feedback. We ensure that the LLM is instructed to generate reasons for a positive rating when the user provides one, and vice versa for negative ratings. For instance, when presenting a prompt for a positive rating, we use "Pros"; we use "Cons" for a negative rating. The template $T_{sub}$ with positive rating is illustrated as follows, with placeholders for RATING, COMMENTS, NAME, ATTRIBUTES, and CATEGORIES to be filled with actual data. The LLM's response will answer the query indicated by the placeholder in red text.

**Subjective Description Prompt Template $T_{sub}$**

A customer rates RATING to the movie with comments: COMMENTS The information of this movie is: name: NAME, attributes: ATTRIBUTES, categories: CATEGORIES. I need you to provide Pros, corresponding evidence and keywords of the rating based on given information. Evidence should refer to the information given. Strictly follow the reply format, fill [], do not say anything else: Pros 1: [pros 1] Keywords: [keywords of pros 1] Evidence: [evidence of pros 1] Pros 2: [pros 2] Keywords: [keywords of pros 2] Evidence: [evidence of pros 2] Pros 3: [pros 3] Keywords: [keywords of pros 3] Evidence: [evidence of pros 3]

Similar to our approach with $T_{obj}$, we prompt the LLM to list the pros, their associated keywords, and supporting evidence, aiming for an output that is both reliable and well-substantiated. From the LLM's response, we extract the keywords that indicate positive aspects as $D^{sub}_{pos}$ and those indicating negative aspects as $D^{sub}_{neg}$.

Upon obtaining the lists of keywords that represent likes and dislikes from both the objective and subjective standpoints, we combine these lists to form comprehensive sets: $D_{pos} = D^{obj}_{pos} \cup D^{sub}_{pos}$ and $D_{neg} = D^{obj}_{neg} \cup D^{sub}_{neg}$. By merging these keyword lists, we develop a holistic view of the potential inclinations users may have towards items. This synthesis allows us to better anticipate and understand the diverse motivations that drive user preferences and behaviors in the context of recommendations.

We then apply a filtering process to refine these keywords, excluding those that are either too common or too rare. This ensures that the keywords we retain are both informative and concise, effectively capturing the essence of item factual information and real users' subjective opinions.

In summary, we have attained informative textual descriptions for each item, including category $D_{cate}$, a set of keywords reflecting reasons for liking $D_{pos}$, and disliking keywords $D_{neg}$. This comprehensive characterization equips our user simulation with a deeper understanding of the items' attributes and the potential reactions from users.

## Logical Model

Following the explicit user interaction logic, we design a logical model to simulate user engagement with the recommended candidate item. To decide whether to like or dislike a candidate item, users consider the characteristics of historical liking items and the potential reasons for liking candidate items and compare them with the characteristics of historical disliking items and the potential reasons for disliking candidate items. In this subsection, we devise two types of logical models leveraging the distilled information of items, *i.e.,* keywords matching model and similarity calculation model.

**Keywords Matching Model** Given user's historical interacted item list $h = \{(i_k, y_k)\}_{k=1}^{n}$ containing indicative features, we design a keywords matching model, which concentrates on the direct matching of textual keywords. Firstly, to retrieve user's preference explicitly, we extract historical interacted items with the same category of $i_c$, noted as $h_C := \{(i_k, y_k)\}_{k=1}^{C}$ with $C$ items. When there are no historical items with the same category, we set $h_C = h$. Then, we extract item set from $h_C$ with positive rating $y_k = 1$ as historical liking items $I_{pos} := \{i_k\}_{k=1}^{C_{pos}}$ and item set with negative rating $y_k = 0$ as historical disliking items $I_{neg} := \{i_k\}_{k=1}^{C_{neg}}$, where $C = C_{pos} + C_{neg}$.

Given the user historical preference $I_{pos}$ and $I_{neg}$, we design a keywords matching model $f_{mat}(I_{pos}, I_{neg}, i_c)$ to infer the user's inclination towards liking or disliking candidate item $i_c$. To implement this, we calculate the number of matched keywords of historical liking reasons, *i.e.,* $D_{pos}$ of items in $I_{pos}$, and potential liking reasons for the candidate item, *i.e.,* $D_{pos}$ of $i_c$. Similarly, we calculate the alignment of historical disliking reasons, *i.e.,* $D_{neg}$ of items in $I_{neg}$ and potential disliking reasons of candidate item, *i.e.,* $D_{neg}$ of $i_c$. Denote the keywords for positive and negative reasons of item $i$ with $D^i_{pos}$ and $D^i_{neg}$, respectively. The number of matched keywords for positive and negative attitudes $\alpha_{pos}$ and $\alpha_{neg}$ can be formulated as:

$$\alpha_{pos} = \sum_{i \in I_{pos}} |D^{i_c}_{pos} \cap D^i_{pos}| \tag{1}$$

$$\alpha_{neg} = \sum_{i \in I_{neg}} |D^{i_c}_{neg} \cap D^i_{neg}| \tag{2}$$

$\alpha_{pos}$ and $\alpha_{neg}$ capture the degree of overlap between the candidate item's potential reasons for liking/disliking and the user's historical preferences, quantifying the user's potential inclination towards the item. The keywords matching model can be represented by Eq. 3:

$$f_{mat}(I_{pos}, I_{neg}, i_c) = \begin{cases} 1 & if \quad \alpha_{pos} > \alpha_{neg}, \\ rand\{0,1\} & if \quad \alpha_{pos} = \alpha_{neg}, \\ 0 & if \quad \alpha_{pos} < \alpha_{neg}. \end{cases} \tag{3}$$

**Similarity Calculation Model** To enhance the underlying interaction logic simulation process beyond mere keyword matching, we introduce similarity calculation model $f_{sim}(I_{pos}, I_{neg}, I_c)$ that leverages embedding representations for a nuanced understanding of user preferences. Akin

| Dataset | #Users | #Items | #Instances | Sparsity |
|---|---|---|---|---|
| **Yelp** | 15,081 | 10,186 | 228,000 | 99.85% |
| **Amazon Music** | 125,627 | 65,019 | 162,261 | 99.99% |
| **Amazon Games** | 30,195 | 23,096 | 165,571 | 99.98% |
| **Amazon Movie** | 24,191 | 49,154 | 972,536 | 99.92% |
| **Anime** | 24,859 | 11,188 | 6,111,860 | 97.80% |

Table 1: Dataset statistics.

to the keywords matching model, this model focuses on the relationship among items within the same category $h_C$ and its respective positive and negative subsets, *i.e.,* $I_{pos}$ and $I_{neg}$. This categorical analysis allows us to discern patterns and preferences that are specific to the category, thereby enhancing the precision of our user inclination inference.

Initially, we employ a representative backbone model to transform keywords into embeddings in the semantic space. Specifically, we harness the capabilities of BERT (Devlin 2018) to produce embeddings that capture the semantic richness of each keyword set, calculating $E_{pos}$ as the average pooling of the embeddings of the elements in $D_{pos}$, and similarly, $E_{neg}$ as the average pooling of the embeddings in $D_{neg}$, as shown in Eqs. 4 and 5:

$$E_{pos} = AvePool(\{Bert(d)|d \in D_{pos}\}) \tag{4}$$

$$E_{neg} = AvePool(\{Bert(d)|d \in D_{neg}\}) \tag{5}$$

We then proceed to assess the closeness of the candidate item's pros and cons to those of historically liked or disliked items. We calculate similarities between the keywords embedding of the user's positively rated items, *i.e.,* $E_{pos}$ of items in $I_{pos}$, and embedding of pros from candidate item, *i.e.,* $E_{pos}$ of $i_c$. Meanwhile, we calculate similarities between the keywords embedding of the user's negatively rated items, *i.e.,* $E_{neg}$ of items in $I_{neg}$, and the cons from candidate item, *i.e.,* $E_{neg}$ of $i_c$. The similarity between keywords for positive and negative attitudes $\beta_{pos}$ and $\beta_{neg}$ can be mathematically presented as follows, where $\phi$ represents similarity metric, and we use cosine similarity.

$$\beta_{pos} = Max\{\phi(E^{i_c}_{pos}, E^{i}_{pos})|i \in I_{pos}\} \tag{6}$$

$$\beta_{neg} = Max\{\phi(E^{i_c}_{neg}, E^{i}_{neg})|i \in I_{neg}\} \tag{7}$$

This comparison with the cosine similarity metric provides a quantifiable and interpretable measure of how the candidate item aligns with the user's historical preferences. The similarity calculation model can be formulated as Eq. 8:

$$f_{sim}(I_{pos}, I_{neg}, i_c) = \begin{cases} 1 & if \quad \beta_{pos} > \beta_{neg}, \\ rand\{0,1\} & if \quad \beta_{pos} = \beta_{neg}, \\ 0 & if \quad \beta_{pos} < \beta_{neg}. \end{cases} \tag{8}$$

## Statistic Model

Beyond the two logical models, we augment our user simulator with a statistical model to enhance the reliability of the generated user inferences. To achieve this, we employ a deep model, SASRec (Kang and McAuley 2018), and train on the user's historical interaction data. Specifically, we pretrain the statistical model $f_{sta}(h, i_c)$ with the dataset. Subsequently, we load the pretrained model parameter to infer the engagement with the candidate item.

This approach introduces the power of traditional statistical model, trained on user historical interaction data, to predict the user's inclination towards a candidate item. It enhances the reliability of the user simulator by integrating statistical learning with logical reasoning.

## Ensemble User Simulator

Given the established two logical models, *i.e.,* keywords matching model $f_{mat}$ and similarity calculation model $f_{sim}$, and statistic model $f_{sta}$, we construct an ensemble model to synergize the user interaction simulation performance.

By integrating the strengths of both logical reasoning and statistical learning, our ensemble model offers a comprehensive and robust framework for simulating user preferences and behaviors in recommendation scenarios.

Next we introduce the training pipeline with reinforcement learning-based recomender system in this subsection.

## MDP Formulation

In reinforcement learning-based recommender system training, the sequential item interactions can be formulated by a Markov Decision Process (MDP) (Puterman 1990). In this pipeline, recommender system serves as an agent, user interaction and preference are the environments, recommendation of item is action, user's response towards recommendation is the reward signal.

- **State** ($s \in \mathcal{S}$): the set of all possible states of the environment, including user profile, historical interactions $h$, and current context including $I_{pos}$ and $I_{neg}$.
- **Action** ($i_c \in \mathcal{A}$): the set of all possible actions the recommender system can take, where an action represents one recommended item $i_c$.
- **Transition Probabilities** ($\mathcal{P}(s'|s, i_c)$): the probabilities of transitioning to a new state $s'$ given the current state $s$ and action $i_c$ from recommender system.
- **Reward Function** ($\mathcal{R}(s, i_c, s')$): assigns a numerical reward to each transition from state $s$ to $s'$ by action $i_c$. We craft an ensemble model to serve as the user simulator, and the reward function is determined by the consensus reached among three constituent base models, which could be formulated as Eq. 9:

$$\mathcal{R}(s, i_c, s') = \begin{cases} 1 & if \quad f_{mat} + f_{sim} + f_{sta} \geq 2, \\ 0 & if \quad f_{mat} + f_{sim} + f_{sta} < 2. \end{cases} \tag{9}$$

- **Discount Factor** $\gamma$: A number between 0 and 1 used to discount future rewards.

# Experiments

## Experimental Setup

To verify the efficacy of the proposed ensemble user simulator, we incorporate datasets from diverse fields: **Yelp**[1] (the

[1]https://www.yelp.com/dataset/documentation/main

| Dataset | Metric | PPO | TRPO | A2C | DQN |
|---|---|---|---|---|---|
| **Yelp** | **A. Rwd↑** | 9.97 | 13.45 | 24.15 | **27.56** |
| | **T. Rwd↑** | 141.57 | 157.42 | 267.60 | **330.98** |
| | **Liking%↑** | 34.59 | 40.07 | 48.35 | **49.43** |
| **Amazon Music** | **A. Rwd↑** | 10.49 | 11.31 | 13.45 | **16.70** |
| | **T. Rwd↑** | 129.03 | 140.15 | 141.03 | **181.42** |
| | **Liking%↑** | 29.30 | 32.46 | 29.54 | **33.18** |
| **Amazon Games** | **A. Rwd↑** | 18.72 | 21.35 | **27.56** | 26.43 |
| | **T. Rwd↑** | 208.43 | 242.26 | **317.56** | 269.02 |
| | **Liking%↑** | 33.15 | 37.64 | **43.52** | 40.73 |
| **Amazon Movie** | **A. Rwd↑** | 29.42 | 27.47 | 31.72 | **38.60** |
| | **T. Rwd↑** | 310.69 | 301.40 | 354.34 | **416.18** |
| | **Liking%↑** | 38.59 | 36.70 | 42.37 | **44.50** |
| **Anime** | **A. Rwd↑** | 14.12 | 14.58 | **21.50** | 18.03 |
| | **T. Rwd↑** | 155.74 | 163.44 | **242.95** | 201.94 |
| | **Liking%↑** | 25.46 | 24.27 | **31.52** | 30.67 |

Table 2: Overall performance. A. Rwd and T. Rwd represent average and total rewards, respectively. Liking% is the liking items ratio in the top-10 recommendations.

state of Missouri), **Amazon**[2] (Digital Music, Video Games, and Movies), and **Anime**[3]. Dataset statistics are shown in Table **??**. We convert ratings into a binary format, where a rating is marked as '1' if it is 3 or higher and '0' for ratings below 3. We use ChatGLM-6B[4] as our LLM. We employ several representative reinforcement learning algorithms: **A2C** (Mnih et al. 2016), **DQN** (Mnih et al. 2013), **PPO** (Schulman et al. 2017), and **TRPO** (Schulman et al. 2015).

## Benchmark Results

We provide the results of reinforcement learning algorithms on our user simulator and report the average reward, total reward, and liking ratio. Experimental results are shown in Table **??**. The results indicate that DQN consistently outperforms other algorithms, a result that can be attributed to its superior capacity for handling tasks with discrete action spaces. DQN combines Q-learning with deep learning and excels at estimating the expected return for each action. The robust performance of DQN in the simulator is likely bolstered by its use of experience replay and target networks.

Besides, all algorithms exhibit good liking ratio of recommendation, which suggests that the user simulator provides a consistent environment where different algorithms can perform their interactions with the recommended items over a similar timescale. The user simulator can produce consistent and reliable interaction patterns across different algorithms, showcasing the simulator's reliability in mimicking real user behavior. It highlights the simulator's strength in replicating genuine user behaviors, which is crucial for accurately assessing algorithmic performance.

[2]https://nijianmo.github.io/amazon/index.html

[3]https://www.kaggle.com/datasets/CooperUnion/anime-recommendations-database

[4]https://huggingface.co/THUDM/chatglm-6b

## Case Study

In this subsection, we delve into case studies to further demonstrate the effectiveness of our user simulator. We first illustrate the recommendations by the DQN on Yelp in Table **??**. Due to space limitation, we showcase a selection of 3 historical items ($i^{t-3}$, $i^{t-2}$, and $i^{t-1}$) and 3 RL recommended items ($i_c^t$, $i_c^{t+1}$, and $i_c^{t+2}$) highlighting their notable pros and cons. We omit cons for positive historical items and pros for negative ones. The user simulator's detailed inference process on the RL recommendations is presented in Table **??**. For instance, for $i_c^t$, the matching cons with $i^{t-1}$ results in a $f_{mat}$ output of 0. Similarly, $f_{mat}$ for $i_c^{t+2}$ is 1, as its pros align with those of $i_{t-2}$. The $f_{mat}$ for $i_c^{t+1}$ being 1 is incidental, given the lack of matching pros or cons.

When faced with items featuring new genres, such as $i_c^{t+1}$, the logical model assesses both the matching of pros and cons and the overall similarity to the historical item set, which may somewhat reduce precision. Nonetheless, our statistical model $f_{sta}$ serves as a crucial fallback. Its collaborative filtering capabilities ensure that the inference remains accurate and well-informed.

In conclusion, our ensemble user simulator harnesses the advantages of both logical and statistical models to explicitly generate user interactions that reflect user preferences. The logical model ensures transparency and consistency in user engagement, while the statistical model captures the subtleties of user behavior, enhancing the simulator's fidelity and effectiveness in emulating real-world user interactions.

## User Simulator Comparison

To clearly position our user simulator with existing user simulators, we present the main features in Table **??**.

For RL-based recommender systems, traditional user simulators typically rely on statistical models to generate user inferences. our simulator provides a transparent and logical method for inferring user engagement, enhancing transparency and realism. The reliance on LLMs for inference will inevitably introduce implementation complexity and computational cost, particularly when dealing with large datasets or high-frequency user interactions. Moreover, the hallucination in LLMs can impede performance, which is an issue that still lacks an effective solution. Unlike other LLM-based simulators that face issues like computational cost and hallucination during training, our approach utilizes the LLM for offline analysis, avoiding direct calls during the training phase and thus mitigating related challenges.

According to the performance comparison in Table **??**, our simulator consistently outperforms the state-of-the-art simulators, which proves its precise approximation to user preference. It also achieves competitive efficiency against both LLM- and non-LLM-based simulators.

# Related Works

**Traditional User Simulator** To bridge the performance gap between offline metrics and online performance of recommender systems, RecoGym (Rohde et al. 2018) simulates user behavior in e-commerce and their responses

| | historical items $h$ | | | RL recommendations $i_c$ | | |
|---|---|---|---|---|---|---|
| | $i^{t-3}$ | $i^{t-2}$ | $i^{t-1}$ | $i_c^t$ | $i_c^{t+1}$ | $i_c^{t+2}$ |
| Name | City Diner | EI Maguey | Popeyes Kitchen | IHOP | Gooey Cakes | Crusoe's Original |
| Category | Restaurants | Restaurants | Restaurants | Restaurants | Bakeries | Restaurants |
| Pros | family gathering | child-friendly | - | casual | variety | child-friendly |
| Cons | - | - | loud, crowded | loud | no in-store dining | no reservations |
| Rating | 1 | 1 | 0 | - | - | - |

Table 3: RL algorithm recommendation on Yelp. Green grids denote the positive aspects of historical items, and orange ones represent the negative aspects, aligning with the framework in Figure 2.

| Recommendation | $f_{mat}$ | $f_{sim}$ | $f_{sta}$ | $R$ |
|---|---|---|---|---|
| $i_c^t$ | 0 | 0 | 1 | 0 |
| $i_c^{t+1}$ | 1 | 0 | 1 | 1 |
| $i_c^{t+2}$ | 1 | 1 | 1 | 1 |

Table 4: User simulator inference on recommendation.

| Simulators | Real dataset | Simulation engine | Evaluation |
|---|---|---|---|
| RecoGym (Rohde et al. 2018) | × | Statistical model | case study |
| RecSim (Ie et al. 2019) | × | Statistical model | case study |
| VirtualTaobao (Shi et al. 2019) | ✓ | GAN | online |
| Adversarial (Chen et al. 2019) | ✓ | GAN | offline |
| KuaiSim (Zhao et al. 2023a) | ✓ | Transformer | offline |
| SUBER (Corecco et al. 2024) | ✓ | LLM | offline & case study |
| Agent4Rec (Zhang et al. 2024a) | ✓ | LLM | offline & case study |
| Ours | ✓ | LLM-based logical & statistical model | offline & case study |

Table 5: User simulator qualified comparison.

| Metric | A. Rwd↑ | T. Rwd↑ | AUC↑ | Infer Time(s) |
|---|---|---|---|---|
| SUBER | 23.74 | 297.48 | 0.643 | 2.42 |
| KuaiSim | 25.35 | 316.46 | 0.658 | 0.53 |
| Ours | **27.56** | **330.98** | **0.674** | 0.76 |

Table 6: User simulator quantified comparison.

to recommendations. Both organic user interactions on e-commerce sites and bandit sessions are incorporated. RecSim (Ie et al. 2019) provides a customizable environment for testing user interaction hypotheses, allowing for tailored user preferences, latent states, dynamics, and choice models. Virtual-Taobao (Shi et al. 2019) uses GANs for realistic customer feature simulation and multi-agent adversarial imitation learning for generalized customer action generation. Similarly, Chen et al. (Chen et al. 2019) uses GANs to model online interaction rewards, introducing a model-based RL technique that enhances sample efficiency in policy learning. Kuaisim (Zhao et al. 2023a) integrates a transformer model for user responses and sets benchmarks at the request, session, and cross-session levels for comprehensive recommender system evaluation.

**LLM-based User Simulator** Given the successful precedents in related areas (Bubeck et al. 2023; Bommasani et al. 2021), there emerge LLM-based user inference simulations leveraging LLM's outstanding semantic understanding and inferring capability. With an empirical study on ChatGPT's performance in conversational recommendation using benchmark datasets, Wang et al. (Wang et al. 2023b) advocate to consider two types of interaction: attribute-based question answering and free-form chit-chat. Aiming at simulating search users, USimAgent (Zhang et al. 2024b) devises LLM agent to fabricate complete search sessions, including querying, clicking, and stopping behaviors, based on specific search tasks. SUBER (Corecco et al. 2024) utilizes LLMs as synthetic users within a gym environment, marking progress towards more realistic training environments for recommender systems. Agent4Rec (Zhang et al. 2024a) focuses on developing a simulator that accurately reflects user preferences and social traits. It leverages LLMs to create agents, each initialized with a unique user profile that includes tastes and social traits, ensuring agents' behaviors mirror those of real human users.

## Conclusion

This paper presents a novel user simulator for RL-based recommender systems. To address the prevalent issues of opacity and simulation evaluation in current systems in existing user simulators, we introduce an LLM-powered user simulator designed to explicitly model user preferences and engagement. Our method identifies explicit logic of user preferences, utilizing LLMs to analyze item characteristics and distill user sentiments. We propose a novel ensemble model that integrates both logical and statistical insights, enhancing the reliability and fidelity of user interaction simulations. We conduct comprehensive experiments across five datasets, demonstrating the simulator's effectiveness and stability in various recommendation scenarios.

Currently, this simulator only infers binary interactions of 'like' or 'dislike'. Future work will focus on integrating additional interaction signals, such as duration, rating, and retention, to enrich the application of the simulator.

## Acknowledgments

This research was partially supported by Collaborative Research Fund (No.C1043-24GF), Research Impact Fund (No.R1015-23), APRC - CityU New Research Initiatives (No.9610565, Start-up Grant for New Faculty of CityU), CityU - HKIDS Early Career Research Grant (No.9360163), Hong Kong ITC Innovation and Technology Fund Midstream Research Programme for Universities Project (No.ITS/034/22MS), Hong Kong Environmental and Conservation Fund (No. 88/2022), and SIRG - CityU Strategic Interdisciplinary Research Grant (No.7020046), the Fundamental Research Funds for the Central Universities, JLU, and Kuaishou.

## References

Bommasani, R.; Hudson, D. A.; Adeli, E.; Altman, R.; Arora, S.; von Arx, S.; Bernstein, M. S.; Bohg, J.; Bosselut, A.; Brunskill, E.; et al. 2021. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*.

Bubeck, S.; Chandrasekaran, V.; Eldan, R.; Gehrke, J.; Horvitz, E.; Kamar, E.; Lee, P.; Lee, Y. T.; Li, Y.; Lundberg, S.; et al. 2023. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*.

Chen, X.; Li, S.; Li, H.; Jiang, S.; Qi, Y.; and Song, L. 2019. Generative adversarial user model for reinforcement learning based recommendation system. In *International Conference on Machine Learning*, 1052–1061. PMLR.

Chu, Z.; Chen, J.; Chen, Q.; Yu, W.; He, T.; Wang, H.; Peng, W.; Liu, M.; Qin, B.; and Liu, T. 2023. A survey of chain of thought reasoning: Advances, frontiers and future. *arXiv preprint arXiv:2309.15402*.

Corecco, N.; Piatti, G.; Lanzendörfer, L. A.; Fan, F. X.; and Wattenhofer, R. 2024. SUBER: An RL Environment with Simulated Human Behavior for Recommender Systems.

Devlin, J. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Fu, Z.; Li, X.; Wu, C.; Wang, Y.; Dong, K.; Zhao, X.; Zhao, M.; Guo, H.; and Tang, R. 2023. A unified framework for multi-domain ctr prediction via large language models. *ACM Transactions on Information Systems*.

Han, X.; Zhu, C.; Hu, X.; Qin, C.; Zhao, X.; and Zhu, H. 2024. Adapting Job Recommendations to User Preference Drift with Behavioral-Semantic Fusion Learning. In Baeza-Yates, R.; and Bonchi, F., eds., *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2024, Barcelona, Spain, August 25-29, 2024*, 1004–1015. ACM.

Huang, F.; Yang, Z.; Jiang, J.; Bei, Y.; Zhang, Y.; and Chen, H. 2024. Large Language Model Interaction Simulator for Cold-Start Item Recommendation. *arXiv preprint arXiv:2402.09176*.

Ie, E.; Hsu, C.-w.; Mladenov, M.; Jain, V.; Narvekar, S.; Wang, J.; Wu, R.; and Boutilier, C. 2019. Recsim: A configurable simulation platform for recommender systems. *arXiv preprint arXiv:1909.04847*.

Ji, Z.; Yu, T.; Xu, Y.; Lee, N.; Ishii, E.; and Fung, P. 2023. Towards mitigating LLM hallucination via self reflection. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, 1827–1843.

Jia, P.; Liu, Y.; Zhao, X.; Li, X.; Hao, C.; Wang, S.; and Yin, D. 2023. MILL: Mutual Verification with Large Language Models for Zero-Shot Query Expansion. *arXiv preprint arXiv:2310.19056*.

Kang, W.-C.; and McAuley, J. 2018. Self-attentive sequential recommendation. In *2018 IEEE international conference on data mining (ICDM)*, 197–206. IEEE.

Li, M.; Zhang, Z.; Zhao, X.; Wang, W.; Zhao, M.; Wu, R.; and Guo, R. 2023. Automlp: Automated mlp for sequential recommendations. In *Proceedings of the ACM Web Conference 2023*, 1190–1198.

Liu, L.; Cai, L.; Zhang, C.; Zhao, X.; Gao, J.; Wang, W.; Lv, Y.; Fan, W.; Wang, Y.; He, M.; et al. 2023a. Linrec: Linear attention mechanism for long-term sequential recommender systems. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 289–299.

Liu, Q.; Wu, X.; Zhao, X.; Zhu, Y.; Zhang, Z.; Tian, F.; and Zheng, Y. 2024a. Large language model distilling medication recommendation model. *arXiv preprint arXiv:2402.02803*.

Liu, Q.; Zhao, X.; Wang, Y.; Wang, Y.; Zhang, Z.; Sun, Y.; Li, X.; Wang, M.; Jia, P.; Chen, C.; Huang, W.; and Tian, F. 2024b. Large Language Model Enhanced Recommender Systems: Taxonomy, Trend, Application and Future. arXiv:2412.13432.

Liu, Z.; Tian, J.; Cai, Q.; Zhao, X.; Gao, J.; Liu, S.; Chen, D.; He, T.; Zheng, D.; Jiang, P.; et al. 2023b. Multi-task recommendations with reinforcement learning. In *Proceedings of the ACM Web Conference 2023*, 1273–1282.

Mnih, V.; Badia, A. P.; Mirza, M.; Graves, A.; Lillicrap, T.; Harley, T.; Silver, D.; and Kavukcuoglu, K. 2016. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, 1928–1937. PMLR.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Graves, A.; Antonoglou, I.; Wierstra, D.; and Riedmiller, M. 2013. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*.

Puterman, M. L. 1990. Markov decision processes. *Handbooks in operations research and management science*, 2: 331–434.

Rohde, D.; Bonner, S.; Dunlop, T.; Vasile, F.; and Karatzoglou, A. 2018. Recogym: A reinforcement learning environment for the problem of product recommendation in online advertising. *arXiv preprint arXiv:1808.00720*.

Schulman, J.; Levine, S.; Abbeel, P.; Jordan, M.; and Moritz, P. 2015. Trust region policy optimization. In *International conference on machine learning*, 1889–1897. PMLR.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Shi, J.-C.; Yu, Y.; Da, Q.; Chen, S.-Y.; and Zeng, A.-X. 2019. Virtual-taobao: Virtualizing real-world online retail environment for reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 4902–4909.

Wang, H.; Liu, X.; Fan, W.; Zhao, X.; Kini, V.; Yadav, D.; Wang, F.; Wen, Z.; Tang, J.; and Liu, H. 2024a. Rethinking large language model architectures for sequential recommendations. *arXiv preprint arXiv:2402.09543*.

Wang, K.; Zou, Z.; Zhao, M.; Deng, Q.; Shang, Y.; Liang, Y.; Wu, R.; Shen, X.; Lyu, T.; and Fan, C. 2023a. RL4RS: A real-world dataset for reinforcement learning based recommender system. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2935–2944.

Wang, M.; Zhao, Y.; Liu, J.; Chen, J.; Zhuang, C.; Gu, J.; Guo, R.; and Zhao, X. 2024b. Large Multimodal Model Compression via Iterative Efficient Pruning and Distillation. In *Companion Proceedings of the ACM on Web Conference 2024*, 235–244.

Wang, X.; Tang, X.; Zhao, W. X.; Wang, J.; and Wen, J.-R. 2023b. Rethinking the evaluation for conversational recommendation in the era of large language models. *arXiv preprint arXiv:2305.13112*.

Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Xia, F.; Chi, E.; Le, Q. V.; Zhou, D.; et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35: 24824–24837.

Yao, J.-Y.; Ning, K.-P.; Liu, Z.-H.; Ning, M.-N.; and Yuan, L. 2023. Llm lies: Hallucinations are not bugs, but features as adversarial examples. *arXiv preprint arXiv:2310.01469*.

Zhang, A.; Chen, Y.; Sheng, L.; Wang, X.; and Chua, T.-S. 2024a. On generative agents in recommendation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1807–1817.

Zhang, E.; Wang, X.; Gong, P.; Lin, Y.; and Mao, J. 2024b. Usimagent: Large language models for simulating search users. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2687–2692.

Zhang, W.; Zhao, X.; Zhao, L.; Yin, D.; Yang, G. H.; and Beutel, A. 2020. Deep reinforcement learning for information retrieval: Fundamentals and advances. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2468–2471.

Zhang, Y.; Shen, G.; Han, X.; Wang, W.; and Kong, X. 2023. Spatio-Temporal Digraph Convolutional Network-Based Taxi Pickup Location Recommendation. *IEEE Trans. Ind. Informatics*, 19(1): 394–403.

Zhang, Z.; Liu, S.; Yu, J.; Cai, Q.; Zhao, X.; Zhang, C.; Liu, Z.; Liu, Q.; Zhao, H.; Hu, L.; et al. 2024c. M3oe: Multi-domain multi-task mixture-of experts recommendation framework. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 893–902.

Zhao, K.; Liu, S.; Cai, Q.; Zhao, X.; Liu, Z.; Zheng, D.; Jiang, P.; and Gai, K. 2023a. KuaiSim: A comprehensive simulator for recommender systems. *Advances in Neural Information Processing Systems*, 36: 44880–44897.

Zhao, K.; Zou, L.; Zhao, X.; Wang, M.; and Yin, D. 2023b. User retention-oriented recommendation with decision transformer. In *Proceedings of the ACM Web Conference 2023*, 1141–1149.

Zhao, X.; Gu, C.; Zhang, H.; Yang, X.; Liu, X.; Tang, J.; and Liu, H. 2021a. Dear: Deep reinforcement learning for online advertising impression in recommender systems. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, 750–758.

Zhao, X.; Xia, L.; Tang, J.; and Yin, D. 2019. Deep reinforcement learning for search, recommendation, and online advertising: a survey. *ACM sigweb newsletter*, 2019(Spring): 1–15.

Zhao, X.; Xia, L.; Zhang, L.; Ding, Z.; Yin, D.; and Tang, J. 2018a. Deep reinforcement learning for page-wise recommendations. In *Proceedings of the 12th ACM conference on recommender systems*, 95–103.

Zhao, X.; Xia, L.; Zou, L.; Liu, H.; Yin, D.; and Tang, J. 2021b. Usersim: User simulation via supervised generativeadversarial network. In *Proceedings of the Web Conference 2021*, 3582–3589.

Zhao, X.; Zhang, L.; Ding, Z.; Xia, L.; Tang, J.; and Yin, D. 2018b. Recommendations with negative feedback via pairwise deep reinforcement learning. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 1040–1048.