# Diagnostic-Guided Dynamic Profile Optimization for LLM-based User Simulators in Sequential Recommendation

**Hongyang Liu[1], Zhu Sun[2*], Tianjun Wei[3], Yan Wang[1*], Jiajie Zhu[1], Xinghua Qu[4]**

[1]School of Computing, Macquarie University, Australia
[2]Information Systems Technology and Design, Singapore University of Technology and Design, Singapore
[3]College of Computing and Data Science, Nanyang Technological University, Singapore
[4]Bytedance Seed, Singapore
hongyang.liu2@hdr.mq.edu.au, zhu_sun@sutd.edu.sg, tjwei2-c@my.cityu.edu.hk
yan.wang@mq.edu.au, jiajie.zhu@mq.edu.au, xinghua.qu1@bytedance.com

## Abstract

Recent advances in large language models (LLMs) have enabled realistic user simulators for developing and evaluating recommender systems (RSs). However, existing LLM-based simulators for RSs face two major limitations: (1) static and single-step prompt-based inference that leads to inaccurate and incomplete user profile construction; (2) unrealistic and single-round recommendation-feedback interaction pattern that fails to capture real-world scenarios. To address these limitations, we propose DGDPO (**D**iagnostic-**G**uided **D**ynamic **P**rofile **O**ptimization), a novel framework that constructs user profile through a dynamic and iterative optimization process to enhance the simulation fidelity. Specifically, DGDPO incorporates two core modules within each optimization loop: firstly, a specialized LLM-based diagnostic module, calibrated through our novel training strategy, accurately identifies specific defects in the user profile. Subsequently, a generalized LLM-based treatment module analyzes the diagnosed defect and generates targeted suggestions to refine the profile. Furthermore, unlike existing LLM-based user simulators that are limited to single-round interactions, we are the first to integrate DGDPO with sequential recommenders, enabling a bidirectional evolution where user profiles and recommendation strategies adapt to each other over multi-round interactions. Extensive experiments conducted on three real-world datasets demonstrate the effectiveness of our proposed framework.

## Introduction

User simulators (Balog and Zhai 2024), which aim to mimic user behaviors and decision-making patterns, are crucial for advancing recommender systems (RSs) (Luo et al. 2022). They offer a controllable and cost-effective paradigm for developing and evaluating RSs, reducing the reliance on real-user experiments that are often expensive and ethically sensitive. While early user simulators ranged from rule-based (Rohde et al. 2018) to deep learning-based approaches (Shi et al. 2019b; Zhao et al. 2021), these traditional methods fail to explicitly represent users' complex semantic preferences and generate diverse user interaction behaviors. To address these issues, Large Language Models (LLMs) have recently emerged as a new paradigm. With their advanced cognitive and generative capabilities, LLMs are typically integrated into an agent framework (Park et al. 2023) that incorporates components such as Profile, Memory, and Action to emulate user interactions with RSs. For example, Agent4Rec (Zhang et al. 2024a) and RecAgent (Wang et al. 2025) prompt LLMs to generate user profile, enabling them to simulate diverse behaviors that mirror the complexity of real users.

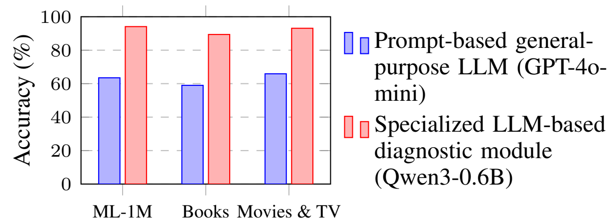


Figure 1: Accuracy comparison for profile defect identification. Our specialized LLM-based diagnostic module significantly outperforms the prompt-based general-purpose LLM.

However, existing LLM-based user simulators face two major limitations: **(1)** *the inaccuracy and incompleteness of profile construction due to the static and single-step prompt-based inference*. By generating the user profile only at initialization, user simulators produce a static representation that cannot adapt to the evolving user interests across multi-round interactions. Thus, any inaccuracy or incompleteness in this initial profile persists uncorrected, causing the simulated behavior to progressively diverge from real user actions throughout the simulation; **(2)** *the unrealistic and single-round recommendation-feedback interaction pattern*. Most existing simulators are typically integrated with static recommenders, e.g., MF (Koren, Bell, and Volinsky 2009), failing to simulate authentic, multi-round interactions where user profile and RS strategies should adapt and evolve based on the latest feedback.

Therefore, we propose a novel **D**iagnostic-**G**uided **D**ynamic **P**rofile **O**ptimization (DGDPO) framework for a more realistic simulation. However, developing this frame-

[*]Corresponding author

work faces two primary challenges. **(CH1)**: *How to build a reliable mechanism for dynamic profile construction?* While existing methods (Shu et al. 2024) like self-reflection (Sun et al. 2024) attempt iterative refinement, they often rely on general-purpose LLMs, thus may not be reliable due to potential hallucinations. As shown in Fig. 1, directly prompting general-purpose LLMs only achieves an average accuracy of 62.78% on profile defect identification. This highlights the necessity of a more targeted strategy that can accurately diagnose the specific type of defect, such as inaccurate and/or incomplete, and then apply a tailored refinement. To achieve this, we first design a specialized LLM-based diagnostic module through domain-adaptive pre-training and defect-specific fine-tuning, enabling reliable identification of profile defects. As shown in Fig. 1, the module achieves an average accuracy of 92.20%, significantly outperforming the general-purpose LLM. We then design a generalized LLM-based treatment module that uses this precise diagnosis to analyze the defect and generate targeted refinements for the profile. Instead of single-step prompting, our framework iteratively optimizes the profile by traversing the user's interaction history step-by-step, to dynamically enhance its accuracy and comprehensiveness. **(CH2)**: *How to achieve realistic multi-round interactions where user profiles and recommender strategies mutually influence and evolve over time?* Inspired by the multi-round interaction nature of real-world recommendation scenarios, we propose to incorporate the user simulator with sequential recommenders (SRs) (Sun et al. 2025). This establishes a genuine bidirectional evolution: the user profile is updated based on interactions, while the SRs adaptively adjust their recommendation strategies in response to the evolving user behavior.

Our main contributions are three-fold: **(1)** To address **CH1**, we propose a novel framework **DGDPO**, which integrates a specialized LLM-based diagnostic module and a generalized LLM-based treatment module. By iteratively optimizing the user profile through step-by-step traversal of user interactions, the framework dynamically refines the user profile, resulting in a more accurate and comprehensive representation. **(2)** To address **CH2**, we establish a realistic and multi-round interaction between the user simulator and sequential recommenders. This enables a bidirectional evolution of user profiles and recommendation strategies, offering a more credible evaluation environment. **(3)** We conduct extensive experiments on three real-world datasets, empirically validating the effectiveness of our DGDPO framework.

## Related Works

**Traditional User Simulators for RSs.** Early user simulators for RSs progressed from rule-based approaches (Rohde et al. 2018; Ie et al. 2019; Shi et al. 2019a) to advanced data-driven techniques (Huang et al. 2020), including methods based on reinforcement learning (RL) (Shi et al. 2019b; Zou et al. 2020), generative adversarial networks (GANs) (Chen et al. 2019; Shi et al. 2019b; Zhao et al. 2021), and transformers (Zhao et al. 2023; Afzali et al. 2023). However, they lack external context and knowledge, as well as the sophisticated cognitive reasoning mechanism that real users leverage during decision-making processes.

**LLM-based User Simulators for RSs.** The advent of LLMs has introduced novel paradigms for user simulation in RSs (Cai et al. 2025; Ma et al. 2025; Wei et al. 2025). LLM-based simulators have been widely adopted across various applications, including evaluating conversational systems (Friedman et al. 2023; Zhu, Huang, and Sang 2024, 2025) and generating high-quality interaction data for RL-based algorithms (Corecco et al. 2024; Ebrat, Paradalis, and Rueda 2024; Zhang et al. 2025). Current works on user profile construction can be categorized into two types. Some works (Zhang et al. 2024a; Wang et al. 2025) create user profile once at initialization, resulting in static representations that cannot capture evolving user preferences. Others employ self-reflection (Zhang et al. 2024b; Shu et al. 2024) for iterative refinement. However, they suffer from two limitations: (1) they adopt general-purpose LLMs which may generate unreliable profile due to potential hallucinations; (2) they rely on unrealistic single-round interactions, lacking dynamic adaptation to user feedback and failing to simulate the bidirectional evolution between user profile and recommendation strategies in realistic interactions.

**Sequential Recommenders.** SRs have evolved from early approaches combining Markov Chains with matrix factorization (Rendle, Freudenthaler, and Schmidt-Thieme 2010), to deep learning models. These include CNNs for local pattern extraction (Tang and Wang 2018), RNNs to capture temporal dependencies (Hidasi 2015; Zhu et al. 2017, 2019), attention mechanisms to focus on relevant historical items (Li et al. 2017; Kang and McAuley 2018; Liu et al. 2018; Sun et al. 2019), and GNNs to model high-order relationships within sequential data (Wu et al. 2022; Yu et al. 2024; Li et al. 2024; Liu, Xia, and Huang 2024). Despite their success, they have primarily been evaluated in a static mode, which may fail to fully reflect the dynamic complexity of real-world user interactions.

## The Proposed Method

**Framework Overview.** DGDPO consists of two core modules that iteratively optimize the user profile in a dynamic manner, as shown in Fig. 2. First, a specialized LLM-based diagnostic module identifies profile defects by detecting discrepancies between simulated and real user behavior. Then, a generalized LLM-based treatment module generates targeted refinements to optimize the profile based on the diagnosis. This optimization process is executed iteratively across a user's historical interaction sequence (i.e., step-by-step traversal of user interactions). For a more realistic simulation, DGDPO is further incorporated with SRs to establish multi-round interactions, enabling bidirectional evolution of user profiles and recommendation strategies.

### The DGDPO Framework (CH1)

Given a user's historical interaction sequence $\mathcal{D}_u = [i_1, i_2, ..., i_T]$ ordered by timestamp, we split it into two portions: $\mathcal{D}_u^{(ini)} = [i_1, i_2, ..., i_n]$, and $\mathcal{D}_u^{(opt)} = [i_{n+1}, ..., i_T]$. For effective and efficient user profile construction, we first create an initial user profile $\mathcal{P}_u^{(ini)}$ for each simulated user $u$. It is derived from $\mathcal{D}_u^{(ini)}$ via a single-step prompt-based

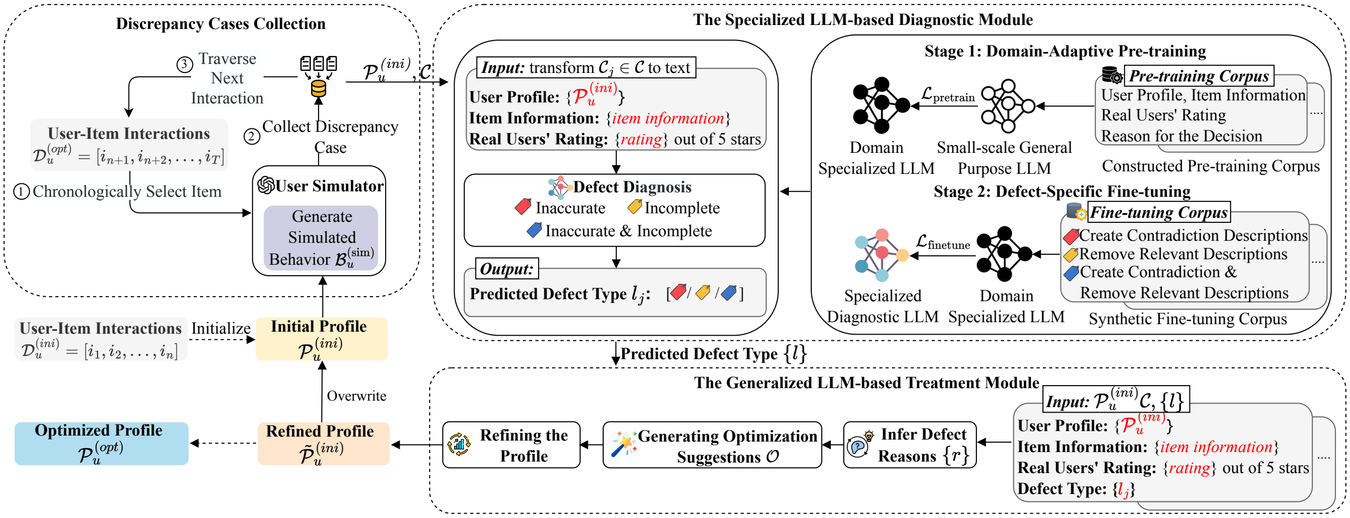


Figure 2: The overall framework of our proposed DGDPO.

inference method (Zhang et al. 2024a), serving as the foundation user profile. Then, we leverage $\mathcal{D}_u^{(opt)}$ to dynamically optimize $\mathcal{P}_u^{(ini)}$ via our proposed DGDPO, ultimately obtaining an optimized user profile $\mathcal{P}_u^{(opt)}$.

This optimization process is triggered by collecting discrepancy cases where the *simulated behavior* $\mathcal{B}_u^{(\text{sim})}$ of user $u$ mismatches his *real behavior* $\mathcal{B}_u^{(\text{real})}$ observed in the datasets (i.e., the simulator declines to interact with an item that the real user has interacted with). Specifically, we present the first item $i_{n+1} \in \mathcal{D}_u^{(opt)}$ to the user simulator built upon $\mathcal{P}_u^{(ini)}$. Then, the user simulator determines whether to interact (i.e., click) with $i_{n+1}$ and produces an explanatory reason for the decision. We traverse items in $\mathcal{D}_u^{(opt)}$ to collect a batch ($N_1$) of such discrepancy cases, which then serve as the input for our optimization workflow.

**The Specialized LLM-based Diagnostic Module.** Given a discrepancy case, this module aims to accurately identify the defect of the user profile via a specialized LLM. As such, it overcomes the inherent unreliability of general-purpose LLMs for such specific diagnostic tasks, providing more accurate signals to guide the optimization process.

First, we categorize each profile defect into one of three types based on its root cause: (1) *Inaccurate*: the profile contains preference descriptions that contradict the real user's observed behavior. (2) *Incomplete*: the profile lacks the specific preference descriptions required to explain the user's interaction with a given item. (3) *Inaccurate & Incomplete*: the profile exhibits both of the aforementioned issues. Then, we formally define this diagnostic task as a function $f_{\text{diag}}$. Given a user profile $\mathcal{P}_u$ and the corresponding discrepancy case $\mathcal{C}_j$, it outputs the corresponding defect type $l = f_{\text{diag}}(\mathcal{P}_u, \mathcal{C}_j)$ from the three defined types.

The key challenge now is how to accurately predict $l$. A straightforward way is to directly prompt general-purpose LLMs. However, their performance may be suboptimal due to limited domain knowledge and inherent hallucination (as illustrated in Fig. 1). Therefore, we employ a delicately designed training strategy (i.e., domain-adaptive pre-training and defect-specific fine-tuning) to build a specialized LLM, transforming it into a specialized expert in profile defect identification by injecting extensive domain knowledge.

***Domain-Adaptive Pre-training.*** This stage is to inject the model with the underlying semantics and reasoning patterns in recommendation. By pre-training on a large corpus of simulated interactions, the model learns the complex relationships between user profiles, item attributes, and interaction decisions, providing a robust foundation for the downstream defect identification task.

To construct the pre-training corpus, for each user $u$, we leverage his initial sequence data $\mathcal{D}_u^{(ini)}$. For each item $i_t \in \mathcal{D}_u^{(ini)}$, we instruct the user simulator with profile $\mathcal{P}_u^{(ini)}$ to make a binary interaction decision for $i_t$ and generate a textual rationale. To ensure the data quality, we only consider the case where the user simulator's decision matches with the real user $u$'s behavior (i.e., choosing to interact with $i_t$) and user $u$ gave item $i_t$ a high rating ($\geq 3$ out of 5). Then, the user simulator's output, together with $\mathcal{P}_u^{(ini)}$ and relevant information (e.g., title, genres, and rating) of $i_t$, is used to construct a comprehensive corpus of structured texts.

Given the constructed corpus, we perform domain-adaptive pre-training on a small-scale general-purpose LLM using a language modeling objective, where the goal is to predict the next token in a sequence. This encourages the LLM to internalize the domain-specific logical flow of profile-based user simulation, defined as:

$$\mathcal{L}_{\text{pretrain}} = -\sum_{t=1}^{T_{Pre}} \log P(x_t | x_{<t}, \theta), \tag{1}$$

where $x = (x_1, \ldots, x_{T_{Pre}})$ represents a text sequence from the corpus with length $T_{Pre}$, and $\theta$ are the model parameters.

***Defect-Specific Fine-tuning.*** Next, we further fine-tune the LLM for the specific profile defect diagnosis task. A primary challenge lies in the absence of available, human-annotated data for defect identification. To overcome this, we develop a synthetic data generation pipeline that creates a high-quality, labeled corpus for fine-tuning purposes.

The pipeline starts with the high-quality, non-defective profile-item pairs that constitute our pre-training corpus. Building upon these non-defective pairs, we subsequently perform a series of targeted modifications to generate synthetic user profiles with different defect types: (1) *Inaccurate*: to create a contradiction, we first identify preference descriptions in the profile relevant to the target item (e.g., its genre). We then generate a new description with the opposite sentiment (e.g., transforming "enjoys comedy films" to "dislikes comedy films") and use it to replace the original statement. (2) *Incomplete*: to simulate a lack of information, we randomly remove a proportion of descriptions relevant to the target item from the profile, making the profile insufficient to explain the user's interaction. (3) *Inaccurate & Incomplete*: we first apply the "incomplete" modification, followed by the "inaccurate" modification. Using these synthetic profiles, we then format the data for instruction fine-tuning, where each sample has four parts: *System*, *Instruction*, *Input*, and *Output*.

To fine-tune the model, we employ a next token prediction loss objective with a modification: the loss is only calculated on the tokens in the *Output* field. The tokens from the other fields are masked from the loss calculation, forcing the model to focus exclusively on more accurate profile defect diagnosis. Thus, the fine-tuning objective is to minimize the loss only over the target response tokens:

$$\mathcal{L}_{\text{finetune}} = -\sum\nolimits_{t \in T_{\text{Output}}} \log P(x_t | x_{<t}, \theta), \tag{2}$$

where $T_{\text{Output}}$ is the set of token indices corresponding to the *Output* field, and $\theta$ are the model parameters.

In summary, our delicately designed training strategy transforms the general-purpose LLM into a highly accurate specialized diagnostic module. It provides a trustworthy and structured diagnostic signal that is essential for guiding the subsequent treatment module.

**The Generalized LLM-based Treatment Module.** This module is responsible for the complex reasoning and modification tasks. While the specialized diagnostic module excels at identifying what defects exist within a profile, understanding why these defects occur and determining how to address them effectively requires sophisticated common-sense reasoning and high-fidelity generative capabilities. Consequently, we employ a powerful general-purpose LLM, leveraging its superior analytical reasoning and text generation capabilities to carry out the following treatment process.

Specifically, we first prompt the LLM by providing it with the current user profile $\mathcal{P}_u^{(ini)}$, the target item $i$, and defect label $l$, to generate a comprehensive textual explanation (i.e., inferred reason $r$) towards the diagnosis made by the diagnosis module. Given the inferred reason, we further instruct the LLM to generate a set of concrete optimization suggestions (i.e., targeted modifications) denoted as $\mathcal{O}$. For instance, an *inaccurate* defect may result in a suggestion to correct contradictory preferences, while an *incomplete* defect would lead to a suggestion to add missing details. Finally, we prompt the LLM to apply the suggestions $\mathcal{O}$ to produce an updated user profile $\tilde{\mathcal{P}}_u^{(ini)}$. In this step, specific constraints are enforced to (1) maintain the profile's overall coherence and descriptive style, and (2) avoid mentioning specific item details, ensuring the resulting profile captures general user preferences.

**Algorithm 1:** DGDPO

**Input:** $\mathcal{D}_u^{(ini)}$, $\mathcal{D}_u^{(opt)}$, $N_1$
**Output:** $\mathcal{P}_u^{(opt)}$

$\mathcal{C} \leftarrow [\,]$; // Initialize
$\mathcal{P}_u^{(ini)} \leftarrow$ Init($\mathcal{D}_u^{(ini)}$); // Inference
**for** *item* $i$ *in* $\mathcal{D}_u^{(opt)}$ **do**
  $\mathcal{B}_u^{(\text{sim})} \leftarrow$ Simu($\mathcal{P}_u^{(ini)}, i$); // Get simulated behavior
  $\mathcal{B}_u^{(\text{real})} \leftarrow$ Real($\mathcal{D}_u^{(opt)}, i$); // Get real behavior
  **if** $\mathcal{B}_u^{(\text{sim})} \neq \mathcal{B}_u^{(\text{real})}$ **then**
    $\mathcal{C}.append(i, \mathcal{B}_u^{(\text{sim})}, \mathcal{B}_u^{(\text{real})})$;
    **if** $|\mathcal{C}| = N_1$ **then**
      // The Diagnostic Module
      $\{l\} \leftarrow f_{\text{diag}}(\mathcal{P}_u^{(ini)}, \mathcal{C})$;
      // The Treatment Module
      $\{r\} \leftarrow$ Infer($\mathcal{P}_u^{(ini)}, \mathcal{C}, \{l\}$)
      $\mathcal{O} \leftarrow$ GenOpt($\mathcal{P}_u^{(ini)}, \mathcal{C}, \{l\}, \{r\}$);
      $\tilde{\mathcal{P}}_u^{(ini)} \leftarrow$ Refine($\mathcal{P}_u^{(ini)}, \mathcal{O}$) $\mathcal{C} \leftarrow [\,]$;
      $\mathcal{P}_u^{(ini)} \leftarrow \tilde{\mathcal{P}}_u^{(ini)}$;
$\mathcal{P}_u^{(opt)} \leftarrow \tilde{\mathcal{P}}_u^{(ini)}$;
**return** $\mathcal{P}_u^{(opt)}$;

**Iterative Optimization.** Given the updated profile $\tilde{\mathcal{P}}_u^{(ini)}$, our framework continues to traverse the remaining interactions in $\mathcal{D}_u^{(opt)}$ to collect the next batch of discrepancy cases. Once a batch is accumulated, the complete workflow is executed: the diagnostic module provides a diagnosis for each case, which subsequently guides the treatment module to produce the next updated profile. This process repeats until all interactions in $\mathcal{D}_u^{(opt)}$ are processed, ultimately yielding the final optimized user profile $\mathcal{P}_u^{(opt)}$. The complete process is detailed in Algorithm 1.

## Bidirectional Evolution between User Profiles and Recommendation Strategies (CH2)

Existing LLM-based user simulators are primarily integrated with static recommenders in page-by-page scenarios (Zhang et al. 2024a), which essentially represent a one-time recommendation result split into multiple pages. This fails to capture realistic scenarios where users continuously interact with the RSs, leading to the dynamic evolution of both user profiles and recommendation strategies through user feedback and interactions. To enable realistic multi-round interactions, we propose to incorporate our optimized user simulator DGDPO with sequential recom-

| | ML-1M | Books | Movies & TV |
|---|---|---|---|
| #Users | 6,040 | 208,864 | 23,969 |
| #Items | 3,416 | 241,725 | 25,830 |
| #Interactions | 999,611 | 10,865,527 | 1,053,194 |
| Avg. Sequence Length | 198.41 | 49.16 | 48.59 |
| Sparsity | 95.16% | 99.98% | 99.83% |

Table 1: Dataset statistics after preprocessing.

menders (SRs). Unlike traditional static models, SRs excel at capturing temporal dynamics, which makes them ideal for modeling the bidirectional evolution of user preferences and recommendation strategies over time.

Specifically, in the multi-round interaction scenario, our DGDPO and SRs engage in a continuous interaction process that consists of four steps: (1) Based on the user's interaction history, the SRs generate a ranked list of candidate items $\mathcal{X}$ for the optimized simulator with profile $\mathcal{P}_u^{(opt)}$, where one item is the ground truth, and the others are randomly sampled negative items; (2) The user simulator decides whether to interact with the recommended items, where at most one item can be selected for interaction in each round; (3) When an item is selected, the simulator performs an update to $\mathcal{P}_u^{(opt)}$ to reflect the recent interaction pattern. In particular, we design two strategies for profile update: (i) *Update without Ground Truth (w/o GT)*: the profile is updated using the item that the simulator chooses to interact with. (ii) *Update with Ground Truth (w/ GT)*: the profile is updated using the ground truth positive item to maintain alignment with real user behavior. In both strategies, the profile update is performed via a carefully controlled prompt that instructs the LLM to make minimal, incremental adjustments while preserving the profile's overall coherence; and (4) Based on this newly observed interaction, the SRs update their internal hidden state for next recommendation. This loop continues until the simulator either chooses to exit or reaches the maximum rounds of interactions.

# Experiments and Results

We conduct extensive experiments to answer five research questions[1]. (**RQ1**) Can DGDPO improve the fidelity of user simulators? (**RQ2**) How do different profile update strategies and SRs impact the simulation fidelity in multi-round interaction scenarios? (**RQ3**) How do various components of DGDPO affect its performance? (**RQ4**) How do essential hyper-parameters affect DGDPO? (**RQ5**) How does DGDPO construct user profile with its optimization process?

## Experimental Setup

**Datasets.** We use three datasets: ML-1M (Harper and Konstan 2015), Amazon Books, and Amazon Movies & TV (Ni, Li, and McAuley 2019). We filter users and items with less than 5 interactions (ML-1M) or less than 20 (Amazon datasets). Interactions are sorted chronologically, with max sequence lengths set to 200 for ML-1M and 50 for Amazon datasets. For each user, the last 10 items are for testing, the 11th most recent item is for validation, and the rest for training (i.e., $\mathcal{D}_u$). $\mathcal{D}_u$ is further split into $\mathcal{D}_u^{(ini)}$ and $\mathcal{D}_u^{(opt)}$ by an $\alpha : (1-\alpha)$ ratio.

**Baselines.** We compare DGDPO with five baselines for user profile construction in two categories. In traditional methods, **KuaiSim** (Zhao et al. 2023) is a transformer-based model for user profile construction. In LLM-based methods, **Agent4Rec** (Zhang et al. 2024a) utilizes a single-step prompt-based inference. **LLM-US** (Zhang et al. 2025) generates explanations of user ratings as profile. **RAH** (Shu et al. 2024) and **AgentCF** (Zhang et al. 2024b) employ self-reflection and collaborative reflection, respectively. *For multi-round interactions, we integrate the user simulator with four SRs*: **FPMC** (Rendle, Freudenthaler, and Schmidt-Thieme 2010), **NARM** (Li et al. 2017), **SASRec** (Kang and McAuley 2018) and **LRURec** (Yue et al. 2024).

**Implementation Details.** Following state-of-the-arts (Shu et al. 2024; Zhang et al. 2024a), we randomly sample 1000 users from each dataset to serve as simulated users. Our diagnostic module is based on lightweight Qwen3-0.6B (Yang et al. 2025), while the treatment module employs gpt-4o-mini via its API with a temperature of 0. All LLM-based baselines also utilize GPT-4o-mini as their backbone. For DGDPO, we set $N_1 = 4$, $\alpha = 0.6$, and $\mathcal{X} = 20$ for all datasets. We build upon Agent4Rec (Zhang et al. 2024a) as the foundation user simulator and conduct ten-round interactions for each user. For KuaiSim and SRs methods, we tune them by Optuna (optuna.org) with 50 trials and train for up to 100 epochs with an early stopping mechanism (Sun et al. 2022). All experiments are conducted on a V100 GPU.

**Evaluation Metrics.** Following (Luo et al. 2022; Zhang et al. 2024a), we adopt Precision (Prec), Recall, Accuracy (Acc), and F1 Score (F1) as the evaluation metrics to measure the consistency between the simulated and real users.

## Results and Analysis

**Overall Comparison (RQ1).** Table 2 presents the performance comparison of all methods. The results show that DGDPO achieves the best performance across all datasets and metrics, with average relative improvements of 16.87%, 18.82%, and 14.63% over the runner-up methods across the three datasets, respectively. These substantial improvements demonstrate the effectiveness of our method. In addition, we observe that DGDPO provides a particularly substantial boost in Precision, with relative improvements of 23.60%, 28.78%, and 19.17% across the three datasets. This indicates that DGDPO can accurately identify and guide the optimization of defective user profiles, generating more precise and faithful user representations that prevent incorrect item selections and significantly enhance simulation precision.

**Analysis of Multi-round Interactions (RQ2).** Fig. 3 depicts the performance of DGDPO across ten interaction rounds on Books dataset under different profile update

[1]Our code and datasets are available at https://github.com/hyllll/DGDPO

| Datasets | | ML-1M | | | | Books | | | | Movies & TV | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Methods | | Prec | Recall | Acc | F1 | Prec | Recall | Accuracy | F1 | Prec | Recall | Acc | F1 |
| Traditional | KuaiSim | 0.1067 | 0.6304 | 0.6651 | 0.1825 | 0.1423 | 0.6107 | 0.6728 | 0.2308 | 0.0832 | 0.5917 | 0.6228 | 0.1459 |
| LLM-based | Agent4Rec | 0.1031 | 0.6250 | 0.6600 | 0.1770 | 0.1521 | 0.6431 | 0.6920 | 0.2460 | 0.0823 | 0.6013 | 0.6322 | 0.1448 |
| | LLM-US | 0.0978 | 0.6208 | 0.6702* | 0.1690 | 0.1457 | 0.6243 | 0.6832 | 0.2363 | 0.0765 | 0.5984 | 0.6281 | 0.1357 |
| | RAH | 0.1104 | 0.6413 | 0.6557 | 0.1884 | 0.1565 | 0.6348 | 0.6874 | 0.2511 | 0.0854 | 0.6078 | 0.6359 | 0.1498 |
| | AgentCF | 0.1157* | 0.6424* | 0.6652 | 0.1961* | 0.1598* | 0.6581* | 0.6953* | 0.2572* | 0.0871* | 0.6105* | 0.6394* | 0.1524* |
| | DGDPO | **0.1430** | **0.7365** | **0.7174** | **0.2395** | **0.2058** | **0.7556** | **0.7570** | **0.3235** | **0.1038** | **0.6913** | **0.6884** | **0.1805** |
| *Improve* | | 23.60% | 14.65% | 7.04% | 22.13% | 28.78% | 14.81% | 8.87% | 25.82% | 19.17% | 13.24% | 7.66% | 18.44% |

Table 2: Performance comparison on all datasets, where the best and runner-up results are highlighted in bold and marked by '*'; and '*Improve*' indicates the relative improvements comparing the best and runner-up results. Statistical significance of the improvement is determined by a paired t-test with $p$-value $< 0.01$.

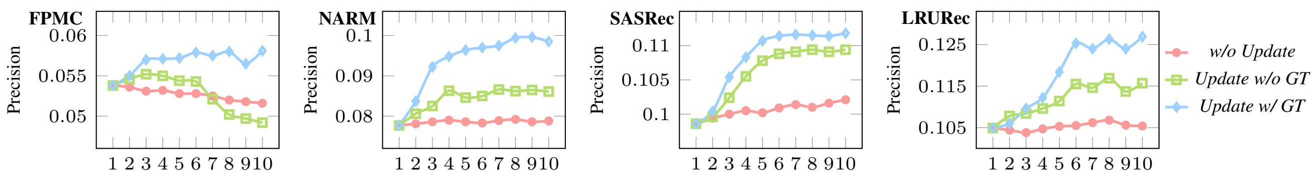


Figure 3: The impact of profile update strategies on DGDPO in multi-round interactions on Books dataset.

strategies[2]. The results unveils that dynamic profile update in multi-round interactions can further improve the simulation accuracy. **(1)** For '*w/o Update*', the performance generally exhibits minimal fluctuations across multi-round interactions and is the worst among all strategies. With profile unchanged, '*w/o Update*' fails to capture the evolution of user preference during multi-round interactions, thereby limiting the simulator's long-term fidelity. **(2)** '*Update w/o GT*' exhibits performance divergence when interacting with different SRs. Specifically, with FPMC, the performance shows a declining trend. In contrast, with advanced deep learning-based SRs, we observe steady performance improvements. This occurs because advanced SRs rank ground truth items higher, and LLM-based user simulators, sensitive to item positions (Hou et al. 2024), are more likely to interact with them when they are prominently placed, resulting in more accurate simulation. **(3)** '*Update w/ GT*' performs the best among all methods. This is attributed to an idealized update strategy that always uses the ground truth item to update the user profile, regardless of the simulator's actions. However, in practical applications, ground truth items are often unavailable, which limits the applicability of this strategy. Therefore, it underscores the importance of developing more advanced user simulators that closely replicate real user behaviors, enabling more effective and reliable updates to the user profile during long-term interactions.

To further validate the effectiveness of our DGDPO in the multi-round interactions, we compare it against representative baselines. All simulators adopt the '*w/o Update*' strategy and interact with four different SRs. Fig. 4 presents the

[2] Hereafter, we only show the results on Books due to space limitation, and similar trends can be noted on the other two datasets.

| Variant | Prec | Recall | Acc | F1 |
|---|---|---|---|---|
| *w/o iteration* | 0.1702 | 0.7204 | 0.7262 | 0.2753 |
| *w/o specialized LLM* | 0.1711 | 0.7219 | 0.7348 | 0.2760 |
| *w/o fine-tuning* | 0.1803 | 0.7220 | 0.7404 | 0.2885 |
| *w/o pre-training* | 0.2027 | 0.7411 | 0.7419 | 0.3183 |
| DGDPO | **0.2058** | **0.7556** | **0.7570** | **0.3235** |

Table 3: Ablation study of DGDPO on Books dataset.

results. **First**, DGDPO consistently achieves the highest Precision across every interaction round, indicating that it better captures user preferences and yields more realistic simulation. **Second**, although AgentCF underperforms DGDPO, it significantly outperforms traditional KuaiSim, reinforcing the effectiveness of LLM-based user simulators.

**Ablation Study (RQ3).** Table 3 presents the results of ablation study. **First**, DGDPO consistently outperforms its '*w/o iteration*' variant, which conducts optimization only once without iteratively traversing interactions step-by-step for dynamic refinement. This showcases the importance of iterative optimization for progressively enhancing user profile quality and achieving superior fidelity. **Second**, we validate the effectiveness of our diagnostic module by comparing DGDPO with the '*w/o specialized LLM*' variant, where the specialized LLM is replaced with a general-purpose LLM. The results confirm that domain-specific specialized LLM enables more reliable and precise diagnostic guidance. **Third**, by removing either domain-adaptive pre-training from the specialized LLM ('*w/o pre-training*') or defect-specific fine-tuning ('*w/o fine-tuning*'), the performance significantly drops compared to DGDPO, validating

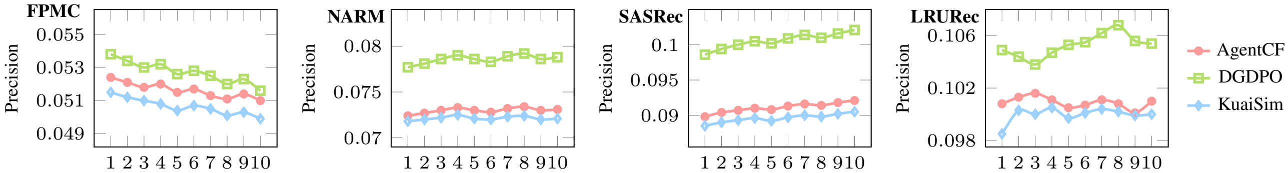


Figure 4: The performance of different user simulators in multi-round interactions on Books dataset.

| Split Ratio $\alpha$ | Prec | Recall | Acc | F1 |
|---|---|---|---|---|
| 0.5 | 0.2042 | 0.7522 | 0.7540 | 0.3212 |
| 0.6 | **0.2058** | **0.7556** | **0.7570** | **0.3235** |
| 0.7 | 0.1977 | 0.7253 | 0.7472 | 0.3107 |
| 0.8 | 0.1902 | 0.7110 | 0.7411 | 0.3001 |

Table 4: The impact of split ratio $\alpha$ on Books dataset.

| Specialized LLM | Prec | Recall | Acc | F1 |
|---|---|---|---|---|
| Qwen3-0.6B (Ours) | **0.2058** | **0.7556** | **0.7570** | **0.3235** |
| Qwen3-1.7B | 0.2007 | 0.7541 | 0.7522 | 0.3170 |

Table 5: The impact of model scale of the specialized LLM in the diagnostic module on Books dataset.

the usefulness of our designed training strategy. Notably, removing fine-tuning results in more substantial performance degradation, indicating a more critical role of defect-specific fine-tuning in profile defect identification. **Overall**, these findings confirm the effectiveness and necessity of all core components within our proposed DGDPO.

**Parameter Sensitivity Analysis (RQ4).** We analyze the impact of split ratio $\alpha$ on the performance of DGDPO. As shown in Table 4, the performance peaks when $\alpha = 0.6$ and declines as the ratio deviates from this value. **(1)** A high $\alpha$ value (e.g., 0.8) leaves limited data for the dynamic optimization, thereby restricting DGDPO's capacity to detect and correct profile defects, which ultimately degrades the performance. **(2)** A low $\alpha$ value (e.g., 0.5) results in profile initialization based on less data, increasing the risk of inaccuracy or incompleteness at the outset. Despite the availability of more data for subsequent optimization, the suboptimal starting point constrains final performance. This underscores the necessity of a balanced data allocation strategy to ensure both robust initialization and effective profile optimization.

We also examine the impact of the model scale of the specialized LLM in the diagnostic module. Table 5 compares the performance of our default Qwen3-0.6B model with a larger 1.7B variant. The results show that using the larger 1.7B model does not yield performance gains; instead, the Qwen3-0.6B model performs better across all metrics. A possible reason is that the profile defect identification is a specialized and fine-grained task. Thus, a smaller and more focused model like Qwen3-0.6B can be more effectively and efficiently fine-tuned using our designed training strategy. This finding indicates that for specialized components such

**Case study: Optimization Process in DGDPO**

**(1) Initial User Profile $\mathcal{P}_u^{(ini)}$**
Engaging narratives, Complex characters, Thought-provoking themes, Rich world-building, Emotional resonance, Unique writing styles.

**(2) Identify User Profile Defect $l$**
**Input:**
**- Simulated User Profile:** Engaging narratives, Complex characters, Thought-provoking themes, Rich world-building, Emotional resonance, Unique writing styles.
**- Item Information:** < Title: Munich, 1938: Appeasement and World War II> <Genre: Politics & Social Sciences, Politics & Government> <History ratings: 4.27>
**- Real User's rating:** 5 out of 5 stars
**Output:** *Incomplete*

**(3) Infer Reasons of Defect $r$**
**Reason:** The simulated user profile *does not mention any preferences related to the genre of Politics & Social Sciences*, which is relevant to the item "Munich, 1938: Appeasement and World War II." Despite the real user rating of 5/5, the profile lacks key information that would indicate an interest in this genre.

**(4) Generating Optimization Suggestions $\mathcal{O}$**
**Suggestion:** *Introduce a preference for "Politics & Social Sciences"* to align with the user's high rating of "Munich, 1938: Appeasement and World War II," indicating an interest in this genre.

**(5) Updated User Profile $\tilde{\mathcal{P}}_u^{(ini)}$**
Engaging narratives, Complex characters, Thought-provoking themes, Rich world-building, Emotional resonance, Unique writing styles, *Politics & Social Sciences*.

Figure 5: The case study on Books dataset.

as our diagnostic module, model scaling should be guided by task requirements rather than model size alone.

**Case Study (RQ5).** Fig. 5 illustrates the DGDPO optimization process with a case study from the Books dataset. A discrepancy (a highly-rated book being ignored by the simulator) triggers the diagnostic module to identify the profile as "*Incomplete*". The treatment module then infers the missing preference for "Politics & Social Sciences" and updates the profile accordingly.

## Conclusion

In this paper, we propose DGDPO, a novel framework that constructs user profile through a dynamic optimization process to enhance simulation fidelity. DGDPO consists of two core modules: a specialized LLM-based diagnostic module calibrated through our novel training strategy first accurately identifies profile defects, which then guides a generalized LLM-based treatment module to generate targeted refinements. Moreover, we propose to incorporate DGDPO with sequential recommenders to enable realistic multi-round interactions, establishing a bidirectional evolution process between user profile and recommendation strategies. Experiments on three real-world datasets demonstrate the superiority of our DGDPO against the state-of-the-art methods.

## Acknowledgments

This research is supported by the Australian Research Council (ARC) Discovery Project DP230100676 and partially supported by the Ministry of Education, Singapore, under its MOE AcRF Tier 1, SUTD Kickstarter Initiative (SKI 2021_06_12).

## References

Afzali, J.; Drzewiecki, A. M.; Balog, K.; and Zhang, S. 2023. UserSimCRS: a user simulation toolkit for evaluating conversational recommender systems. In *Proceedings of the 16th ACM International Conference on Web Search and Data Mining (WSDM)*, 1160–1163.

Balog, K.; and Zhai, C. 2024. User Simulation for Evaluating Information Access Systems. *Foundations and Trends in Information Retrieval*, 18(1-2): 1–261.

Cai, S.; Zhang, J.; Bao, K.; Gao, C.; Wang, Q.; Feng, F.; and He, X. 2025. Agentic feedback loop modeling improves recommendation and user simulation. In *Proceedings of the 48th International ACM SIGIR conference on Research and Development in Information Retrieval (SIGIR)*, 2235–2244.

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 *Proceedings of the 36th International Conference on Machine Learning (ICML)*, 1052–1061.

Corecco, N.; Piatti, G.; Lanzendörfer, L. A.; Fan, F. X.; and Wattenhofer, R. 2024. An LLM-based Recommender System Environment. *arXiv preprint arXiv:2406.01631*.

Ebrat, D.; Paradalis, E.; and Rueda, L. 2024. Lusifer: LLM-based user simulated feedback environment for online recommender systems. *arXiv preprint arXiv:2405.13362*.

Friedman, L.; Ahuja, S.; Allen, D.; Tan, Z.; Sidahmed, H.; Long, C.; Xie, J.; Schubiner, G.; Patel, A.; Lara, H.; et al. 2023. Leveraging large language models in conversational recommender systems. *arXiv preprint arXiv:2305.07961*.

Harper, F. M.; and Konstan, J. A. 2015. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (TIIS)*, 5(4): 1–19.

Hidasi, B. 2015. Session-based Recommendations with Recurrent Neural Networks. *arXiv preprint arXiv:1511.06939*.

Hou, Y.; Zhang, J.; Lin, Z.; Lu, H.; Xie, R.; McAuley, J.; and Zhao, W. X. 2024. Large language models are zero-shot rankers for recommender systems. In *Proceedings of the 46th European Conference on Information Retrieval (ECIR)*, 364–381. Springer.

Huang, J.; Oosterhuis, H.; De Rijke, M.; and Van Hoof, H. 2020. Keeping dataset biases out of the simulation: A debiased simulator for reinforcement learning based recommender systems. In *Proceedings of the 14th ACM conference on recommender systems (RecSys)*, 190–199.

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*.

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

Koren, Y.; Bell, R.; and Volinsky, C. 2009. Matrix factorization techniques for recommender systems. *Computer*, 42(8): 30–37.

Li, J.; Ren, P.; Chen, Z.; Ren, Z.; Lian, T.; and Ma, J. 2017. Neural attentive session-based recommendation. In *Proceedings of the 26th ACM on Conference on Information and Knowledge Management (CIKM)*, 1419–1428.

Li, X.; Chen, H.; Yu, J.; Zhao, M.; Xu, T.; Zhang, W.; and Yu, M. 2024. Global Heterogeneous Graph and Target Interest Denoising for Multi-behavior Sequential Recommendation. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining (WSDM)*, 387–395.

Liu, Q.; Zeng, Y.; Mokhosi, R.; and Zhang, H. 2018. STAMP: short-term attention/memory priority model for session-based recommendation. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining (KDD)*, 1831–1839.

Liu, Y.; Xia, L.; and Huang, C. 2024. Selfgnn: Self-supervised graph neural networks for sequential recommendation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, 1609–1618.

Luo, X.; Liu, Z.; Xiao, S.; Xie, X.; and Li, D. 2022. Mindsim: user simulator for news recommenders. In *Proceedings of the 31st ACM Web Conference (TheWebConf)*, 2067–2077.

Ma, C.; Xu, Z.; Ren, Y.; Hettiachchi, D.; and Chan, J. 2025. PUB: An LLM-Enhanced Personality-Driven User Behaviour Simulator for Recommender System Evaluation. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, 2690–2694.

Ni, J.; Li, J.; and McAuley, J. 2019. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, 188–197.

Park, J. S.; O'Brien, J. C.; Cai, C. J.; Morris, M. R.; Liang, P.; and Bernstein, M. S. 2023. Generative agents: Interactive simulacra of human behavior. *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology (UIST)*.

Rendle, S.; Freudenthaler, C.; and Schmidt-Thieme, L. 2010. Factorizing personalized markov chains for next-basket recommendation. In *Proceedings of the 19th international conference on World wide web (WWW)*, 811–820.

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*.

Shi, B.; Ozsoy, M. G.; Hurley, N.; Smyth, B.; Tragos, E. Z.; Geraci, J.; and Lawlor, A. 2019a. Pyrecgym: A reinforcement learning gym for recommender systems. In *Proceedings of the 13th ACM Conference on Recommender Systems (RecSys)*, 491–495.

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

Shu, Y.; Zhang, H.; Gu, H.; Zhang, P.; Lu, T.; Li, D.; and Gu, N. 2024. RAH! RecSys–Assistant–Human: A Human-Centered Recommendation Framework With LLM Agents. *IEEE Transactions on Computational Social Systems (TCSS)*.

Sun, F.; Liu, J.; Wu, J.; Pei, C.; Lin, X.; Ou, W.; and Jiang, P. 2019. BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM international conference on information and knowledge management (CIKM)*, 1441–1450.

Sun, Y.; Yang, X.; Sun, Z.; Wang, Y.; Wang, B.; and Qu, X. 2025. LLM4RSR: Large Language Models as Data Correctors for Robust Sequential Recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 39, 12604–12612.

Sun, Z.; Fang, H.; Yang, J.; Qu, X.; Liu, H.; Yu, D.; Ong, Y.-S.; and Zhang, J. 2022. Daisyrec 2.0: Benchmarking recommendation for rigorous evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 45(7): 8206–8226.

Sun, Z.; Liu, H.; Qu, X.; Feng, K.; Wang, Y.; and Ong, Y. S. 2024. Large language models for intent-driven session recommendations. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, 324–334.

Tang, J.; and Wang, K. 2018. Personalized top-n sequential recommendation via convolutional sequence embedding. In *Proceedings of the eleventh ACM international conference on web search and data mining (WSDM)*, 565–573.

Wang, L.; Zhang, J.; Yang, H.; Chen, Z.-Y.; Tang, J.; Zhang, Z.; Chen, X.; Lin, Y.; Sun, H.; Song, R.; et al. 2025. User behavior simulation with large language model-based agents. *ACM Transactions on Information Systems (TOIS)*, 43(2): 1–37.

Wei, T.; Guo, H.; Du, Y.; Sun, Z.; Huang, C.; Wang, D.; and Zhang, J. 2025. Mirroring Users: Towards Building Preference-aligned User Simulator with User Feedback in Recommendation. *arXiv preprint arXiv:2508.18142*.

Wu, B.; He, X.; Zhang, Q.; Wang, M.; and Ye, Y. 2022. GCRec: Graph-augmented capsule network for next-item recommendation. *IEEE Transactions on Neural Networks and Learning Systems (TNNLS)*, 34(12): 10164–10177.

Yang, A.; Li, A.; Yang, B.; Zhang, B.; Hui, B.; Zheng, B.; Yu, B.; Gao, C.; Huang, C.; Lv, C.; et al. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Yu, M.; Zhu, K.; Zhao, M.; Yu, J.; Xu, T.; Jin, D.; Li, X.; and Yu, R. 2024. Learning Neighbor User Intention on User–Item Interaction Graphs for Better Sequential Recommendation. *ACM Transactions on the Web (TWEB)*, 18(2): 1–28.

Yue, Z.; Wang, Y.; He, Z.; Zeng, H.; McAuley, J.; and Wang, D. 2024. Linear recurrent units for sequential recommendation. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining (WSDM)*, 930–938.

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 (SIGIR)*, 1807–1817.

Zhang, J.; Hou, Y.; Xie, R.; Sun, W.; McAuley, J.; Zhao, W. X.; Lin, L.; and Wen, J.-R. 2024b. Agentcf: Collaborative learning with autonomous language agents for recommender systems. In *Proceedings of the 33rd ACM on Web Conference (TheWebConf)*, 3679–3689.

Zhang, Z.; Liu, S.; Liu, Z.; Zhong, R.; Cai, Q.; Zhao, X.; Zhang, C.; Liu, Q.; and Jiang, P. 2025. Llm-powered user simulator for recommender system. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 39, 13339–13347.

Zhao, K.; Liu, S.; Cai, Q.; Zhao, X.; Liu, Z.; Zheng, D.; Jiang, P.; and Gai, K. 2023. KuaiSim: A comprehensive simulator for recommender systems. *Proceedings of the 37th Conference on Neural Information Processing Systems (NeurIPS)*, 36: 44880–44897.

Zhao, X.; Xia, L.; Zou, L.; Liu, H.; Yin, D.; and Tang, J. 2021. Usersim: User simulation via supervised generativeadversarial network. In *Proceedings of the 30th ACM Web Conference (TheWebConf)*, 3582–3589.

Zhu, L.; Huang, X.; and Sang, J. 2024. How Reliable is Your Simulator? Analysis on the Limitations of Current LLM-based User Simulators for Conversational Recommendation. In *Companion Proceedings of the ACM on Web Conference 2024 (WWW)*, 1726–1732.

Zhu, L.; Huang, X.; and Sang, J. 2025. A llm-based controllable, scalable, human-involved user simulator framework for conversational recommender systems. In *Proceedings of the ACM on Web Conference 2025 (WWW)*, 4653–4661.

Zhu, Y.; Gong, Y.; Liu, Q.; Ma, Y.; Ou, W.; Zhu, J.; Wang, B.; Guan, Z.; and Cai, D. 2019. Query-based interactive recommendation by meta-path and adapted attention-gru. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management (CIKM)*, 2585–2593.

Zhu, Y.; Li, H.; Liao, Y.; Wang, B.; Guan, Z.; Liu, H.; and Cai, D. 2017. What to do next: Modeling user behaviors by time-LSTM. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI)*, volume 17, 3602–3608. Melbourne, VIC.

Zou, L.; Xia, L.; Du, P.; Zhang, Z.; Bai, T.; Liu, W.; Nie, J.-Y.; and Yin, D. 2020. Pseudo Dyna-Q: A reinforcement learning framework for interactive recommendation. In *Proceedings of the 13th International Conference on Web Search and Data Mining (WSDM)*, 816–824.