Mirroring Users: Towards Building Preference-aligned User Simulator with User Feedback in Recommendation

Tianjun Wei[1] **Huizhong Guo**[2] **Yingpeng Du**[1*] **Zhu Sun**[3]
Huang Chen[1] **Dongxia Wang**[2] **Jie Zhang**[1]
[1]Nanyang Technological University, Singapore
[2]Zhejiang University, Hangzhou, China
[3]Singapore University of Technology and Design, Singapore
tjwei2-c@my.cityu.edu.hk, huiz_g@zju.edu.cn, dyp1993@pku.edu.cn,
zhu_sun@sutd.edu.sg, huang012@e.ntu.edu.sg,
dxwang@zju.edu.cn, zhangj@ntu.edu.sg

Abstract

User simulation is increasingly vital to develop and evaluate recommender systems (RSs). While Large Language Models (LLMs) offer promising avenues to simulate user behavior, they often struggle with the absence of specific task alignment required for RSs and the efficiency demands of large-scale simulation. A vast yet underutilized resource for enhancing this alignment is the extensive user feedback inherent in RSs, but leveraging it is challenging due to its ambiguity, noise and massive volume, which hinders efficient preference alignment. To overcome these hurdles, we introduce a novel data construction framework that leverages user feedback in RSs with advanced LLM capabilities to generate high-quality simulation data. Our framework unfolds in two key phases: (1) using LLMs to generate decision-making processes as explanatory rationales on simulation samples, thereby reducing ambiguity; and (2) data distillation based on uncertainty estimation and behavior sampling to efficiently filter the most informative, denoised samples. Accordingly, we fine-tune lightweight LLMs, as user simulators, using such high-quality dataset with corresponding decision-making processes. Extensive experiments confirm that our framework significantly boosts the alignment with human preferences and the in-domain reasoning capabilities of the fine-tuned LLMs, providing more insightful and interpretable signals for RS interaction. We believe our work, together with publicly available developed framework, high-quality mixed-domain dataset, and fine-tuned LLM checkpoints, will advance the RS community and offer valuable insights for broader human-centric AI research. Our code is available at https://github.com/Joinn99/UserMirrorer.

*Corresponding author.

1 Introduction

User-centric AI, which uses behavior analysis to understand human preferences, is fundamental to research in recommender systems (RSs), social networks, and search engines (Wang et al., 2025; Adomavicius and Tuzhilin, 2005; Gao et al., 2023; Wang et al., 2024b). As a crucial branch, RSs are vital in web services, mitigating information overload by continuously capturing user preferences from interaction histories (e.g., products, content) (Wang et al., 2023b; Wu et al., 2026). RS development, as a direct AI service, critically depends on user feedback. While online testing remains the gold standard for real user feedback capture (Deng et al., 2017), its practical limitations are significant: (1) Achieving statistical significance requires prolonged data collection (weeks/months), substantially delaying algorithm iteration (Gilotte et al., 2018); (2) Increasing privacy concerns challenge traditional user data-dependent evaluations (Stavinova et al., 2022).

These constraints call for virtual replicas of users that can faithfully reproduce real behavior without direct human involvement. The concept of digital twins, which creates virtual counterparts of real-world entities for simulation and optimization, provides a principled framework for this goal (Intizar Ali et al., 2021). Originally rooted in industrial systems such as smart manufacturing (Intizar Ali et al., 2021) and battery analytics (Zhu et al., 2026), digital twins have evolved toward cognitive modeling that replicates not only observable states but also the reasoning and decision-making of complex agents. In the context of RSs, such cognitive digital twins take the form of user simulators (Ie et al., 2019), mechanisms that generate synthetic yet faithful user behavior data to enable efficient, privacy-preserving evaluation and optimization.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 6046–6071

User simulators generate synthetic profiles to simulate user-system interactions, enabling the offline evaluation of RSs. Earlier methods primarily used rule-based (Wei et al., 2018) or reinforcement learning approaches (Ie et al., 2019; Zhao et al., 2023). However, these traditional simulators are limited: they focus only on intra-system preferences, neglecting essential external contextual knowledge and the reasoning underlying real user decisions. Recent advances in Large Language Models (LLMs) (Team et al., 2024), with their extensive world knowledge and advanced multi-step reasoning capabilities (Guo et al., 2025, 2026), offer a solution. They have enabled the development of simulators with human-like cognitive models in various fields, including paper review (Jin et al., 2024), social network (Gao et al., 2023), and even smart manufacturing (Intizar Ali et al., 2021). Research on LLM-based user simulators generally follows two paths: (1) using LLMs' generalization for feature extraction in traditional simulators (Corecco et al., 2024; Zhang et al., 2025; Liu et al., 2026), and (2) creating autonomous agents to simulate fine-grained individual user behaviors in recommendation scenarios (Zhang et al., 2024a,b; Wang et al., 2024a; Zhang et al., 2024c). Despite their promise, these LLM-based approaches have two major drawbacks: First, they over-rely on pre-trained LLM knowledge without fine-tuning on massive user feedback, resulting in poor task adaptation. Second, the need for a powerful LLM backend introduces prohibitive computational costs for large-scale user simulations in RSs due to the lack of task-specific knowledge.

In this work, we explore how to harness user feedback from RSs to fine-tune LLM-based user simulators, thereby improving their fidelity to authentic user behavior. In the context of RSs, raw user feedback exhibits two distinctive characteristics that present unique challenges on the development of LLM-based user simulators:

- User feedback typically captures only observed behaviors, omitting the underlying **decision-making processes**. This absence of context introduces **ambiguity** and hinders the detection of **noisy** data, limiting LLM from learning the human decision-making process.

- RSs generate **massive volumes of feedback with highly variable quality**, making it essential to filter and select **high-quality** samples to achieve effective and efficient LLM fine-tuning.

To address these challenges, we propose an end-to-end framework that fully exploits user feedback in RSs to align LLM simulators with genuine user preferences, named USERMIRRORER. Specifically, our framework first transforms feedback from diverse domains into a unified simulation scenario, capturing user memory, item exposures, and associated behaviors. Then, by prompting LLMs to generate explicit decision-making processes as explanatory rationales, we can reduce feedback ambiguity, denoise behavior signals, and quantify LLM performance across contexts. Finally, by selecting scenarios that expose the limitations of weaker LLMs, and matching decision processes with real user behaviors, we can curate a refined dataset comprising **challenging samples, high-fidelity decision processes generated by powerful LLMs, and denoised behavioral signals.**

Based on our developed framework, we investigate the following research questions: (RQ1) *Does the constructed dataset enable LLM simulator to make actions that align with human preferences with explanatory rationales?* (RQ2) *Does the fine-tuned user simulator more effectively mimic user behavior and provide enhanced support to the RSs?* To answer these questions, we construct a high-quality dataset from eight diverse domains and fine-tune LLMs to derive user simulators. Experimental results demonstrate that these LLM simulators achieve significant improvements in alignment with real user preferences, and able to provide effective and interpretable feedback to RSs.

We are committed to open-sourcing our framework[1], the constructed dataset[2], and the fine-tuned LLM-based user simulators[3], aiming to provide the academic and industrial communities with a tool that effectively enhances the realism and efficiency of user simulators. Our framework enables developers to build customized domain-specific user simulators, thereby advancing research and applications in RSs and beyond.

2 Building User Simulation Scene from User Feedback

2.1 Task formulation

In real-world internet applications, user behaviors are highly diverse, extending beyond simple clicks

[1] https://github.com/Joinn99/UserMirrorer

[2] https://huggingface.co/datasets/Joinn/UserMirrorer

[3] https://huggingface.co/Joinn/UserMirrorer-Llama-DPO

to include complex interactions such as ratings, writing reviews, and sharing content. Considering the core focus of RSs and the primary objective of user simulators, we concentrate on a prevalent and generalizable impression-aware scenario (Yang et al., 2012). In this scenario, a Recommender System (RS) presents a user with an exposure (a list of items). The user then acts on this exposure based on their preferences (e.g., clicking an item). While some existing works independently predict the user's interaction probability for each item (Zhang et al., 2025, 2024a), we adopt a list-wise approach (Zhao et al., 2023; Deng et al., 2025). This means we simulate user behavior at the level of the entire exposure, more closely mirroring how humans interact in reality. To emulate this behavior, the user simulator must rely on the user's current state. This state comprises their profile, historical interactions, and potential emotional factors that influence decisions (Zhang et al., 2025; Wang et al., 2025). We collectively term this comprehensive information the user simulator's **"memory"**. Based on the above definitions, the user simulation scene we focus on can be formally defined as follows:

Definition 2.1 (User Simulation Scene in RSs) *Let* $\mathbf{M}$ *denote the user's memory (e.g., profile and interaction history) and let* $\mathbf{L} = \{l_1, \ldots, l_k\}$ *be the exposure list presented to the user. We form a simulation scene* $\mathbf{X} = \mathrm{Prompt}(\mathbf{M}, \mathbf{L})$. *The action space is* $\mathbf{A} = \{a_1, \ldots, a_k\}$*, where action* a_i *corresponds to interacting with item* l_i. *A user simulator parameterized by* θ *defines a categorical distribution over* $\mathbf{A}$ *conditioned on* $\mathbf{X}$*:*

$$P_\theta(a_i \mid \mathbf{X}) \in [0,1], \qquad \sum\nolimits_{i=1}^{k} P_\theta(a_i \mid \mathbf{X}) = 1. \quad (1)$$

Equivalently, we write $P_\theta(\mathbf{A} \mid \mathbf{X}) = \big(P_\theta(a_1 \mid \mathbf{X}), \ldots, P_\theta(a_k \mid \mathbf{X})\big)$. *The goal of the simulator is to learn* $P_\theta(\cdot \mid \mathbf{X})$ *so as to match the distribution of real user behaviors.*

Existing LLM-based simulators typically focus on augmenting the user's contexts by utilizing the knowledge and reasoning capabilities of the LLM as well as the agent's memory mechanisms (Wang et al., 2024a; Zhang et al., 2024c,a,b). Comparatively, this research emphasizes the exploration of fine-tuning the LLM to **better match human preferences in certain user simulation scene**. The fine-tuned LLM can be used as a plugin to integrate with the existing LLM agent-based user simulator work to achieve better alignment with real user preference and higher efficiency.

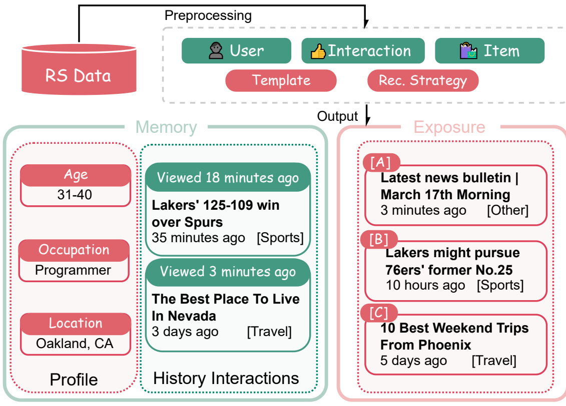


Figure 1: Example of the user simulation scene.

2.2 Creating user simulation scenes

To effectively leverage the massive user feedback data for training and aligning the LLM user simulator, we follow the ideas from existing works based on LLM agents (Corecco et al., 2024; Zhang et al., 2024a). We organize the raw feedback data into an user simulation scene that an LLM agent can understand. The core elements of a user simulation scene include two key components: **memory** and **exposure**, as shown in Figure 1.

Memory. Follow existing work (Corecco et al., 2024), we consider two main components in memory: user profile and interaction history. The user profile contains the user's static attributes, and interaction history record a series of past interactions with items. To accommodate the varying information granularity across different datasets, we develop a series of domain-specific templates. These templates systematically convert the available user attributes and interacted item features within each dataset into a standardized plain text format, for example, *"Age: 35-44, Occupation: Customer Service, User Interaction History: Crimson Tide (1995) Rating: 4/5 ..."* (Please refer to Figure 1). This unified textual representation then serves as the memory input for the user simulator, enabling it to accurately model user preferences. Detailed templates for all datasets can be found in the Appendix E.1.

Exposure. In most public RS datasets, typically only information about items with which the user actually interacted is available, lacking the complete exposure the user saw when making their decisions (Perez Maurera et al., 2025). Therefore, for datasets missing this information, we utilize existing recommendation strategies to construct an exposure list. To ensure this list more accurately reflects items a user might genuinely prefer, we build a hybrid exposure list based on the user's interaction behavior. This list is generated by uniformly sampling K items from three distinct sources: random sampling, collaborative filtering (Shen et al., 2021) and content-based nearest neighbor retrieval (Deshpande and Karypis, 2004) The specific implementation details and parameter configurations are described in Appendix B.1. Finally, we randomly sample N items in total from all Top-K item lists and randomly insert the ground truth item into an arbitrary position within this list, resulting in a final exposure list of length $N+1$. We set K=32 and N is uniformly sampled in the range of 2 to 12 across different users to ensure the generalization capability.

Upon completing the construction of the user memory and exposure, we establish a scene for the LLM user simulator. The user simulator then takes the corresponding behavior from the exposures. To aid our subsequent analysis of the LLM's behaviors, we adopt practices from existing ranking and search-related works (Sun et al., 2023; Chen et al., 2025) to use alphabetical identifiers such as *[A], [B], [C]*, etc., to label each option in the exposures. This approach encourages the LLM to output clear option identifiers rather than free-form text descriptions. In our developed framework USERMIRRORER, the scene construction process uses domain-specific templates to transform user feedback data from various domains into a unified format that LLMs can easily interpret. Specifically, we design the templates for datasets from eight different domains, including movies, books, news, etc., to convert the data to simulation scenes for further experiment and analysis. Detailed descriptions, statistics and template used for all datasets can be found in the Appendix A.1 and E.1.

3 USERMIRRORER: Building a Preference-aligned User Simulator

3.1 Can user feedback be helpful in user preference alignment?

To validate the effectiveness of leveraging user feedback data in aligning user simulators with real user preferences and to explore the factors that may influence this process, we conduct a straightforward preliminary experiment. Specifically, we randomly sample 1,024 user simulation scenes each for the training and test sets, and fix the exposure list length at 5. We employ different base LLMs

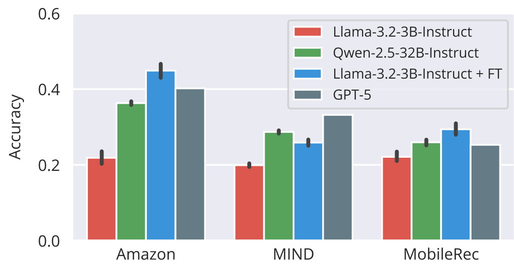


Figure 2: Accuracy of user simulation on representative domains with different base LLMs.

as user simulators, and fine-tune a *Llama-3.2-3B-Instruct* model for 1 epoch using the construct training set (See Appendix C.2 for fine-tuning configurations). We measure accuracy by matching LLM-predicted behaviors against actual user behaviors, and report representative results from three domains in Figure 2.

Stronger models perform better. Experimental results consistently show that LLMs with more advanced foundational capabilities achieve higher alignment with real user behaviors. Since these LLMs have not been fine-tuned on domain-specific user feedback in RSs, their more human-like behaviors likely stem from their robust context modeling and reasoning capabilities, enabling them to more effectively extract relevant information from the user's memory in relation to the current exposure. This indicates that the intrinsic capabilities of LLMs serve as a crucial foundation for their successful simulation of user preferences.

Fine-tuning using real user feedback works. The most significant finding is that directly utilizing user feedback data for fine-tuning is indeed effective. Figure 2 shows that after fine-tuned, the accuracy of the user behavior prediction shows a significant improvement. This demonstrates that the real user feedback data may contain latent and consistent preferences across different users.

However, it is also observed that on the MIND dataset, the stronger *Qwen-2.5-32B-Instruct* and *GPT-5* still achieves better alignment with real user preferences than the fine-tuned weaker LLM. This suggests that the current data alone might be insufficient for optimal preference alignment, and the performance still can be potentially improved.

3.2 Data distillation based on uncertainty decomposition

As demonstrated above, harnessing the synergy between powerful LLMs' intrinsic reasoning capabilities and the implicit preferences in user feedback can markedly align user simulator with user preference. However, the nature of user feedback in RSs may pose following challenges:

- **First**, user "memory" from profiles and past interactions lacks key context (e.g., environment, mental state), so the user intent remains unclear, bringing **ambiguity** to the data.
- **Second**, raw logs often include **noise** (e.g., accidental clicks) that brings negative effects.
- **Third**, behaviors stem from hidden decision steps, from impulse to deliberation, require strong LLM reasoning abilities to handle. Hence, it will be difficult to directly leverage user feedback to **transfer such capabilities** from a powerful LLM to a lightweight LLM, to meet the efficiency requirements of user simulation in RSs.

Given the limitations inherent in user feedback and the vast amount of feedback data in real-world RSs, it may not be optimal or **efficient** to fine-tune an LLM user simulator using all available raw data. Instead, selecting high-quality data to address these limitations could be a more effective approach. In this section, we seek to distill a curated subset of high-quality training samples, and propose a solution that kills two birds with one stone.

Motivation. In Hou et al. (2024), the author suggests that through generating clarifications, we can effectively distinguish and quantify two main sources of uncertainty during LLM generating process (Gal and Ghahramani, 2016): *aleatoric (data) uncertainty*, which arises from inherent data ambiguity, and *epistemic (model) uncertainty*, which stems from LLM's limitations in knowledge and capability. Building on this, Hou et al. (2024) suggest to generate multiple clarifications for inputs, thereby reducing the ambiguity of inputs with clarification and better estimate the model uncertainty.

Definition 3.1 (Uncertainty Decomposition) *Assume $P(\boldsymbol{C}|\boldsymbol{X})$ is distribution of the generated clarifications given a input $\mathbf{X}$. Then, the prediction is defined as the ensemble of predictions conditional on input with clarifications, i.e., $P(\boldsymbol{Y}|\boldsymbol{X}) = \mathbb{E}_{P(\boldsymbol{C}|\boldsymbol{X})}[P(\boldsymbol{Y}|\boldsymbol{X} \oplus \boldsymbol{C}, \boldsymbol{\theta})]$ (Model parameter $\boldsymbol{\theta}$ is a constant and will be omitted for brevity below.), and the decomposed uncertainty can be represented as:*

$$\underbrace{\mathcal{H}(P(\boldsymbol{Y}|\boldsymbol{X}))}_{①} = \underbrace{\mathcal{I}(\boldsymbol{Y};\boldsymbol{C}|\boldsymbol{X})}_{②} + \underbrace{\mathbb{E}_{P(\boldsymbol{C}|\boldsymbol{X})}\mathcal{H}(P(\boldsymbol{Y}|\boldsymbol{X} \oplus \boldsymbol{C}))}_{③}, \tag{2}$$

*where $P(\boldsymbol{Y}|\boldsymbol{X} \oplus \boldsymbol{C})$ denote the distribution of output with the clarification $\boldsymbol{C}$, $\mathcal{H}$ denotes the entropy function, $\mathcal{I}(\boldsymbol{Y};\boldsymbol{C}|\boldsymbol{X})$ denotes the mutual information between the output distribution $\boldsymbol{Y}$ and clarifications $\boldsymbol{C}$. Here, ② is the approximation of **aleatoric uncertainty**, ③ represents the **epistemic uncertainty**, and ① denotes the total uncertainty. In practice, the expectation terms are approximated via the ensemble of N clarifications sampled from the same input $\mathbf{X}$.*

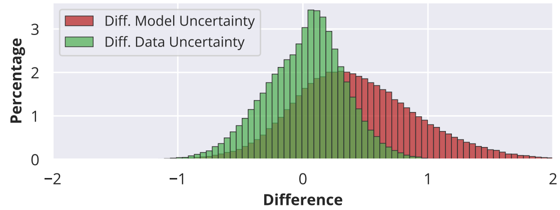


Figure 3: Results of uncertain decomposition on user simulation scenes.

Decision-process Generation. Following this idea, we aim to decompose the uncertainty lied in user behavior simulation. When a human user provides feedback, there is always an underlying decision-making process. We believe that a complete decision-making process can play a "clarifying" role in user simulation. We adopt the widely researched consumer behavior model, the ***Engel Kollat Blackwell (EKB)*** model (Engel et al., 1978), with appropriate adaptations for user simulation scenes. This model was originally used to explain the decision-making processes of user purchasing, encompassing the complete journey from need recognition to post-purchase evaluation. To make it more suitable to user simulation in RSs, we have slightly adjust the steps of the model. The complete decision-making process includes the following stages:

- **Stimulus**: Identify the factors triggering the user's interaction, including external (time, location, social context) and internal (needs, emotional state) elements.
- **Knowledge**: Extract relevant factors and attributes from the exposures for further analysis.
- **Evaluation**: Evaluate potential behaviors using

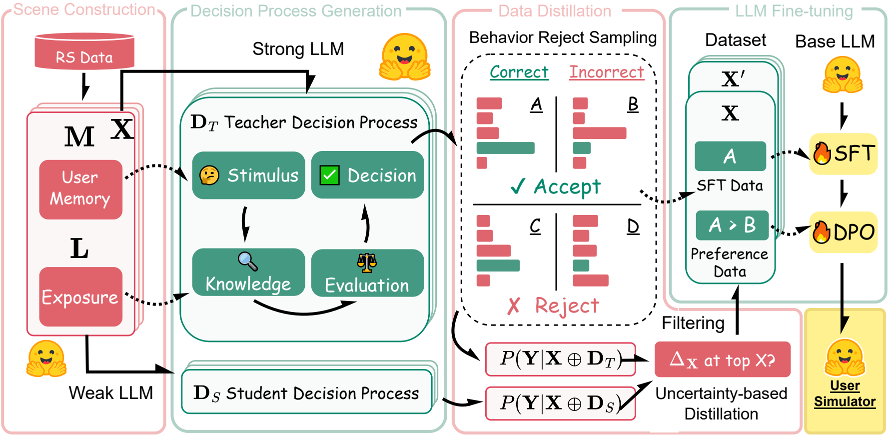


Figure 4: Illustration of our proposed USERMIRRORER framework.

different styles (Kahneman, 2011), from intuitive to logical.

After generating the above decision-making process, the LLM user simulator will predict the corresponding user behavior (See Appendix E.1 for full prompt and F for example case). This decision process furnishes the user simulation scenario with essential contextual information, thereby **reducing ambiguity** in raw user feedbacks.

Uncertainty-based Data Distillation. Moreover, when the decision process effectively leverages information from the user's memory, the simulator should choose its behavior with high confidence, reflecting the model's epistemic uncertainty. Therefore, by comparing the epistemic uncertainties of different LLMs on the same scene, we can **quantify their relative capabilities in specific user simulation scenes**, and select challenging cases where strong and weak LLMs exhibit larger difference in epistemic uncertainties. Suppose the decision-making process of scene $\boldsymbol{X}$ generated by LLMs A and B as $\boldsymbol{D}_A$ and $\boldsymbol{D}_B$ respectively, the difference of epistemic uncertainties is defined as:

$$\begin{aligned}\Delta_{EU}(\boldsymbol{X},(A,B)) =&\mathbb{E}_{P(\boldsymbol{D}_A|\boldsymbol{X})}\mathcal{H}(P(\boldsymbol{Y}|\boldsymbol{X}\oplus\boldsymbol{D}_A)) \\ &-\mathbb{E}_{P(\boldsymbol{D}_B|\boldsymbol{X})}\mathcal{H}(P(\boldsymbol{Y}|\boldsymbol{X}\oplus\boldsymbol{D}_B)).\end{aligned} \tag{3}$$

To validate effectiveness of the above method in uncertainty decomposition, we have conducted an experiments. We randomly sample 16k user simulation scenes from each dataset, and employ *Llama-3.2-3B-Instruct* and *Qwen2.5-32B-Instruct* as the weaker and stronger models, respectively, prompting each to generate $N = 10$ decision-making processes per scene. (See Appendix A.2 for the details of configurations of LLM inference). We then use the generated processes to guide *Llama-3.2-3B-Instruct* in predicting user behavior and compute the uncertainty difference based on the log-probabilities of its output options. Figure 3 shows that weaker LLMs tend to have higher epistemic uncertainty across most scenes. Scenes with large uncertainty gaps between stronger and weaker models are likely more complex, subtle, or involve nuanced decision-making—areas where weak LLMs often struggle. Therefore, we believe that focusing on these challenging cases will yield more effective datasets for improving the LLM user simulator.

Sampling Denoised Behaviors. Although the distillation process generates challenging simulation scenes, they may still contain noisy behaviors. Viewing each generated decision process as a rationale for its predicted action, we employ rejection sampling to remove potential noise. Specifically, for each scene, we compare the predictions regarding to N decision-making processes to the actual user behavior. We consider the scenes for which none of the decision processes yields a correct match as the potential noisy behaviors, and discard it. For the remaining scene, we mark the decision process according to the matching of behaviors as *accepted* and *rejected*, and formulate the samples with the highest confidence as the preference pair.

The full dataset construction pipeline, is illustrated in Figure 4, and the detailed steps are shown in Appendix B.2.

4 Experiments

In this section, we conduct comprehensive evaluations of our data construction pipeline and the fine-tuned LLM user simulator. Our analysis focuses on the two key research questions raised:

- **RQ1**: *Can our constructed dataset further align LLM user simulators with human preferences?*
- **RQ2**: *Can the fine-tuned user simulator provide effective and interpretable feedback for RSs?*

4.1 Dataset Construction and Fine-tuning LLMs

Datasets. For the final dataset construction, we reuse the scenes constructed from each domain in Section 3.2. Through empirical testing of different dataset sizes (presented in Section 4.2.1), we determine that 10,000 samples provide an optimal balance between data quality and computational efficiency. For evaluation, we randomly sample 1,280 and 5,120 scenes for MIND (with real exposures) and other domains (with synthetic exposures) respectively. We fix the length of exposure list in evaluation set to 5. All evaluation scenes are constructed with randomly sampled users and are excluded before the dataset construction phase to prevent possible data leakage (Sun et al., 2020). The final constructed dataset set composition and statistics are provided in Appendix A.2.

Fine-tuning LLMs. We primarily utilize *Llama-3.2-3B-Instruct*, the same model employed during the dataset construction phase, for fine-tuning. To validate the generalization capability of our constructed dataset, we also incorporate *Qwen2.5-3B-Instruct*. We adopt a two-stage fine-tuning process for these LLMs. Initially, Supervised Fine-tuning (SFT) is performed using the *accepted* responses. Subsequently, we conduct preference alignment through Direct Preference Optimization (Rafailov et al., 2023).[4] Given that predicted behaviors can be verified through simple rules, we also experiment with Group Relative Policy Optimization (GRPO) (Shao et al., 2024). Detailed information regarding the runtime environment and training configurations are listed in the Appendix C.2.

4.2 Is USERMIRRORER will aligned with user preference? (RQ1)

We begin by evaluating the accuracy of our fine-tuned LLM User Simulator in replicating real user behavior across multiple domains. For all inference process, we set the sampling parameters temperature to 1.0 and *top-p* to 0.9. To account for stochastic variation, we randomly sample 5 outputs for each scene and report the mean accuracy.

[4]Unless explicitly stated otherwise, all subsequent tests are run using *Llama-3.2-3B-Instruct* as the base model, and fine-tuned through the two-stage SFT+DPO process.

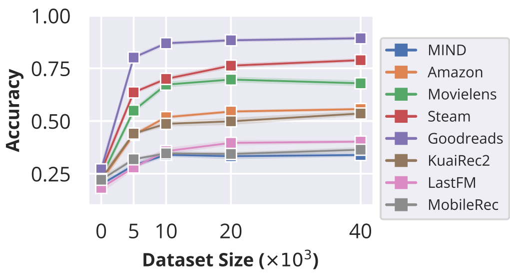


Figure 5: Effect of training dataset size on the performance of user simulator.

Table 1 presents a comprehensive comparison of various LLM served as user simulators. These fine-tuned LLM user simulators not only outperform the base LLMs and stronger teacher LLMs, but also exceed the performance of the most advanced proprietary LLMs in most of the cases, thereby demonstrating the efficacy of leveraging user feedback to better align with user preferences. Moreover, DPO delivers more pronounced improvements compared to GRPO, which we attribute to the relatively simple reward definition employed in GRPO, which could be further explored in the future to fully leverage the potential of reinforcement learning.

4.2.1 Detailed analysis

In this section, we present a detailed analysis on the effect of dataset construction settings on the performance of the fine-tuned user simulators.

Effect of training dataset size. To validate the size of datasets on the performance of fine-tuned LLM simulator, we construct a series of variants using our pipeline, with the varying number from 5,000 to 40,000. We increase the epoches when smaller datasets is used to mitigate the impact of inadequate training. As shown in Figure 5, accuracy rises sharply until approximately 10,000 samples, after which further increases yield only marginal improvements. It indicates that, beyond a certain dataset size, additional examples contribute limited gains in preference alignment and generalization, as existing data are already able to provide sufficient preference alignment information.

Effects of data selection strategies. To validate the effect of data selection strategy in constructing high-quality dataset, we compare a range of baselines: random sampling with or without decision processes, selection based on high, low, or differ-

	Type of Exposure	Real Exposure	Synthetic Exposure							
	Dataset	**MIND**	**Amazon**	**KuaiRec2**	**LastFM**	**Goodreads**	**Movielens**	**MobileRec**	**Steam**	**Overall**
Base Models	**Gemma-3-4B-it**	27.5	24.8	27.7	19.8	50.0	42.2	26.1	31.1	29.6
	Gemma-3-12B-it	29.1	34.4	26.8	26.8	56.5	55.0	23.1	33.8	36.0
	Qwen2.5-7B-Instruct	25.9	29.7	28.1	24.9	57.2	46.9	27.0	33.9	33.7
	Qwen2.5-14B-Instruct	27.9	31.5	32.3	24.5	63.1	55.3	26.0	29.2	35.6
	Qwen2.5-32B-Instruct (Teacher)	28.7	36.3	31.1	27.3	70.7	54.6	26.0	42.4	39.7
Fine-tuned Models	**Qwen2.5-3B-Instruct**	27.2	23.7	26.4	22.8	41.1	33.2	22.4	31.2	27.2
	+SFT	30.9	43.3	38.2	32.6	76.2	57.4	30.4	56.6	46.4
	+SFT+GRPO	32.8	49.9	44.9	36.0	85.3	64.6	32.5	67.7	53.1
	+SFT+DPO	33.5	**52.3**	46.2	35.3	86.3	66.2	32.5	**71.3**	54.7
	Llama-3.2-3B-Instruct	19.9	21.9	22.5	18.3	27.1	23.3	22.1	26.2	22.7
	+SFT	27.3	44.1	40.2	28.5	74.1	51.5	30.0	59.8	46.0
	+SFT+GRPO	32.2	48.4	46.6	**38.2**	82.7	65.3	32.6	68.0	52.7
	+SFT+DPO	**34.0**	51.8	**48.7**	35.7	**86.9**	**67.3**	**34.7**	70.0	**55.0**
Proprietary Models	**GPT-5-Nano (2025-08-07)**	28.4	35.4	31.9	27.8	65.6	57.8	24.5	33.3	36.8
	GPT-5-Mini (2025-08-07)	31.4	37.6	32.3	33.8	74.4	62.8	25.0	40.8	40.7
	GPT-5 (2025-08-07)	33.2	40.2	30.8	37.2	78.9	61.1	25.3	45.2	42.2
	GPT-5.1 (2025-11-13)	31.0	42.9	31.3	41.1	74.5	63.3	26.6	41.9	42.6
	Gemini-2.5-Flash (2025-06-17)	31.8	43.3	30.9	34.2	73.0	64.4	26.4	46.1	42.5
	Gemini-3.0-Pro-Preview (2025-11-18)	34.8	48.0	32.7	46.1	82.0	73.6	29.2	48.1	47.7

Table 1: Accuracy of user simulation on various domains of datasets (in percentage).

Dataset	MIND	Synthetic Exposure Overall
Llama-3.2-3B-Instruct	19.9$_{(0.6)}$	22.7$_{(0.5)}$
Random (w/o Decisions)	25.7$_{(0.8)}$	48.3$_{(0.6)}$
Random (w/ Decisions)	31.4$_{(1.4)}$	53.9$_{(0.4)}$
High Accuracy	32.4$_{(0.2)}$	53.3$_{(0.1)}$
Low Accuracy	30.9$_{(0.8)}$	54.5$_{(0.2)}$
Diff. Accuracy	30.1$_{(0.9)}$	53.9$_{(0.2)}$
IFD Score	29.9$_{(0.6)}$	52.7$_{(0.3)}$
Ours	**34.0**$_{(0.6)}$	**55.0**$_{(0.3)}$

*Standard deviations are shown in parentheses.
*Accuracy and standard deviation are shown in percentage.

Table 2: Effect of data selection strategies.

ential prediction accuracy of the predicted behaviors, and selecting using the Instruction-Following Difficulty (IFD) (Li et al., 2024) score. (Detailed setup of baselines is listed in Appendix C.4.) In our pipeline, as summarized in Table 2, our dataset construction pipeline outperforms these baselines across both real and synthetic exposures. This result confirms our pipeline, which combines uncertainty-based measures and behavior sampling based on real user feedback, yields robust, preference-aligned simulators than accuracy-based or instruction-following filtering metrics.

4.2.2 Analysis of decision making process

We conduct an in-depth analysis of how the decision-making process generated by our user simulator affects the accuracy of user behavior prediction. We require the model to summarize the key factors in each stage of decision-making process. We extract these factors from the generated results on the test set, then clean and categorize them, merging semantically similar terms. Ultimately, we derive factors from 3 stages: *Stimulus*, *Knowledge*, and *Evaluation*.

Our statistical analysis reveals that both the stimulus factors and the final evaluation style are correlated with prediction accuracy. As shown in Figure 6a, when users simulator explicitly express *"thematic preferences"*, the model achieves significantly higher prediction accuracy compared to cases where only *"boredom"* are expressed. Additionally, decision processes adopt a *"logical"* style tend to rely more on the LLM's reasoning ability, yielding higher accuracy than *"intuitive"* style. " We also find a positive correlation between the length of descriptions for stimulus and knowledge factors and the model's prediction accuracy, as shown in Figure 6b. In other words, the more factors a user considers and the more detailed the descriptions during the decision process, the better the alignment between the model's predictions and actual user behavior. This finding suggests that rich and diverse decision process information helps improve the performance of the user simulator.

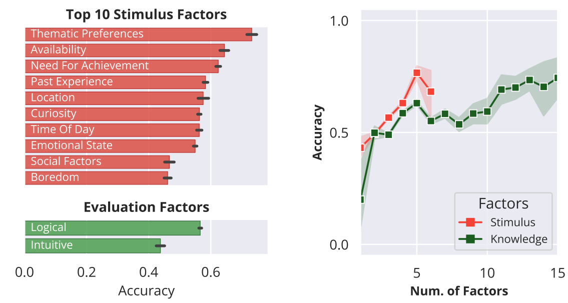
In summary, the generative decision-making process provides greater interpretability for user modeling and valuable intermediate variables for subsequent behavior analysis and causal inference.

4.3 Is USERMIRRORER helpful in providing feedback to RS? (RQ2)

To evaluate whether our user simulator can provide valuable feedback signals to RS, we design an interactive simulation framework to run the test. First, we train a RS on the original training dataset (labeled as **"Backbone"**), where all user interactions used in fine-tuning the simulator are excluded. Next, we generate a candidate recommendation list and expose it to the user simulator, collecting its simulated feedback. These synthetic interactions are then appended to the RS's training data,

Model		Movielens						Steam					
		R@5	**R@10**	**N@5**	**N@10**	**M@5**	**M@10**	**R@5**	**R@10**	**N@5**	**N@10**	**M@5**	**M@10**
LightGCN	**Backbone**	5.178	8.782	3.230	4.386	2.596	3.069	3.371	5.447	2.090	2.762	1.672	1.950
	Base LLM	5.596	9.465	3.513	4.751	2.834	3.337	3.524	5.800	2.225	3.030	1.803	2.132
	SFT+DPO	**6.089**	**9.996**	**3.845**	**5.098**	**3.112**	**3.623**	**3.707**	**5.922**	**2.389**	**3.108**	**1.958**	**2.256**
	Improve	17.6%	13.8%	19.0%	16.2%	19.8%	18.0%	10.0%	8.7%	14.3%	12.5%	17.1%	15.7%
DiffRec	**Backbone**	5.197	9.219	3.313	4.613	2.700	3.236	3.195	5.337	2.010	2.693	1.622	1.899
	Base LLM	6.051	9.674	4.026	5.198	3.366	3.851	3.619	**5.915**	2.470	3.209	2.095	2.398
	SFT+DPO	**6.449**	**10.338**	**4.544**	**5.793**	**3.924**	**4.434**	**3.619**	5.886	**2.515**	**3.243**	**2.155**	**2.452**
	Improve	24.1%	12.1%	37.1%	25.6%	45.3%	37.0%	13.3%	10.3%	25.2%	20.4%	32.9%	29.1%
SASRec	**Backbone**	7.303	12.785	4.279	6.038	3.297	4.015	5.191	8.949	3.268	4.475	2.640	3.134
	Base LLM	8.024	13.961	4.800	6.689	3.748	4.510	4.621	8.927	2.836	4.212	2.256	2.814
	SFT+DPO	**8.365**	**13.999**	**5.027**	**6.852**	**3.938**	**4.694**	**5.637**	**9.549**	**3.606**	**4.864**	**2.945**	**3.460**
	Improve	14.5%	9.5%	17.5%	13.5%	19.4%	16.9%	8.6%	6.7%	10.3%	8.7%	11.6%	10.4%
NARM	**Backbone**	6.791	12.747	3.973	5.879	3.059	3.834	5.272	9.074	3.352	4.573	2.725	3.223
	Base LLM	8.024	14.662	4.761	6.891	**3.700**	**4.569**	5.454	9.681	3.389	4.741	2.716	3.265
	SFT+DPO	**8.175**	**14.757**	**4.772**	**6.895**	3.672	4.545	**5.783**	**9.907**	**3.716**	**5.037**	**3.037**	**3.575**
	Improve	20.4%	15.8%	20.1%	17.3%	20.0%	18.5%	9.7%	9.2%	10.9%	10.1%	11.4%	10.9%

Table 3: Performance comparison of user simulators on recommendation scenarios. (In percentage)



(a) Content (b) Number

Figure 6: Effect of the generated decision-making process on the performance of user simulator.

and the model is retrained to incorporate the additional signals. We conduct this procedure across 10 representative RS algorithms (Due to space limitations, we present the results of LightGCN (He et al., 2020), DiffRec (Wang et al., 2023a), SASRec (Kang and McAuley, 2018), and NARM (Li et al., 2017), in Table 3. Additional model results are provided in Appendix C.3.) based on two datasets Movielens and Steam, comparing the performance of the fine-tuned simulator against the base LLM. We evaluate each scenario using *Recall (R@k)*, *Normalized Discounted Cumulative Gain (N@k)* and *Mean Reciprocal Rank (M@k)*, where k denotes denotes the number of items recommended to the user by the RS.

As shown in Table 3, augmenting RS training with feedback from our fine-tuned simulator yields consistent gains in all three metrics, demonstrating that preference-aligned simulation produces constructive signals that significantly enhance recommendation quality.

5 Conclusion and Future Work

In this work, we address the challenges of ambiguity, noise, and scale in leveraging user feedback for user simulation in RSs. Our proposed framework, USERMIRRORER, fully exploits user feedback to align LLM-based user simulators with authentic user preferences. It transforms raw feedback into unified simulation scenarios, generates explicit human-like decision-making processes, and distills high-quality examples for fine-tuning. Extensive experiments demonstrate that our fine-tuned simulators achieve superior alignment with real user behavior and provide more interpretable feedback to RSs, resulting in significant improvements in recommendation quality.

More broadly, the cognitive modeling perspective underlying our work aligns with the digital twins paradigm in AI-driven decision systems (Intizar Ali et al., 2021). Our approach builds virtual user replicas by extracting explicit reasoning from noisy behavior, similar to constructing digital counterparts in other domains using imperfect real-world signals. For example, in battery health management, physics-informed models combine domain knowledge and data-driven learning to predict degradation from noisy sensor data under uncertainty (Zhu et al., 2026). Both settings use knowledge transfer to bridge rich expert models and lightweight deployable replicas, and leverage uncertainty-aware selection to address observation noise (Kang et al., 2021; Li et al., 2025). These parallels suggest our methodology may also benefit other AI systems requiring reliable decision-making under noisy, incomplete observations.

Limitations

Our work has several limitations that point to promising directions for future research. First, our current framework does not incorporate multi-modal information (e.g., images, audio, or video), which can be crucial for user simulation in domains such as short video or music recommendation. Second, our approach focuses on modeling explicit user interactions and does not address the scenario where users are exposed to items but choose not to interact (i.e., user leaving or non-interaction feedback). While some prior work introduce modules to predict such user-leaving behaviors, the lack of corresponding public data makes it challenging to construct and evaluate these cases in our setting. As a result, our study centers on leveraging existing explicit feedback to enhance user simulation, and we leave the implicit non-interaction modeling for future work.

Ethical Considerations

User simulators trained on recommendation feedback may inherit the biases embedded in the underlying data. For instance, skewed preference distributions across demographic groups or content domains can be amplified by the simulator. To mitigate this risk, we deliberately incorporated feedback from as many diverse domains as possible into the training process, aiming to reduce systematic bias and improve generalizability.

Although the simulator leverages user-level profiles, we exclusively rely on publicly available datasets in which the original authors have already applied rigorous anonymization procedures. Consequently, no personally identifiable information is retained. Any resemblance of artificial generated profile descriptions and decision making process to real individuals is purely coincidental. All dataset reuse, modification, and related operations strictly complied with the copyright statements and usage policies issued by the respective data owners.

Acknowledgments

This research is supported by A*STAR under its MTC Programmatic (Award M23L9b0052); and by the Ministry of Education, Singapore, under its Academic Research Fund (AcRF) Tier 1 grant, and funded through the SMU-SUTD Internal Research Grant Call (SMU-SUTD 2023_02_01); and by the Ministry of Education, Singapore, under its Academic Research Fund Tier 2 (Award No. MOE-T2EP201230015).

References

G. Adomavicius and A. Tuzhilin. 2005. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6):734–749.

Thierry Bertin-Mahieux, Daniel P.W. Ellis, Brian Whitman, and Paul Lamere. 2011. The million song dataset. In *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR 2011)*.

Shijie Chen, Bernal Jimenez Gutierrez, and Yu Su. 2025. Attention in Large Language Models Yields Efficient Zero-Shot Re-Rankers. In *The Thirteenth International Conference on Learning Representations*.

Nathan Corecco, Giorgio Piatti, Luca A. Lanzendörfer, Flint Xiaofeng Fan, and Roger Wattenhofer. 2024. SUBER: An RL environment with simulated human behavior for recommender systems. In *Proceedings of the 27th European Conference on Artificial Intelligence (ECAI 2024)*.

Alex Deng, Jiannan Lu, and Jonthan Litz. 2017. Trustworthy Analysis of Online A/B Tests: Pitfalls, challenges and solutions. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, WSDM '17, pages 641–649, New York, NY, USA. Association for Computing Machinery.

Jiaxin Deng, Shiyao Wang, Kuo Cai, Lejian Ren, Qigen Hu, Weifeng Ding, Qiang Luo, and Guorui Zhou. 2025. OneRec: Unifying Retrieve and Rank with Generative Recommender and Iterative Preference Alignment. *Preprint*, arXiv:2502.18965.

Mukund Deshpande and George Karypis. 2004. Item-based top-N recommendation algorithms. *ACM Trans. Inf. Syst.*, 22(1):143–177.

J.F. Engel, R.D. Blackwell, and D.T. Kollat. 1978. *Consumer Behavior*. Business and Management: Marketing. Dryden Press.

Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. In *Proceedings of The 33rd International Conference on Machine Learning*, pages 1050–1059. PMLR.

Chen Gao, Xiaochong Lan, Zhihong Lu, Jinzhu Mao, Jinghua Piao, Huandong Wang, Depeng Jin, and Yong Li. 2023. S3: Social-network Simulation System with Large Language Model-Empowered Agents. *Preprint*, arXiv:2307.14984.

Chongming Gao, Shijun Li, Wenqiang Lei, Jiawei Chen, Biao Li, Peng Jiang, Xiangnan He, Jiaxin Mao, and Tat-Seng Chua. 2022. Kuairec: A fully-observed dataset and insights for evaluating recommender systems. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 540–550.

Alexandre Gilotte, Clément Calauzènes, Thomas Nedelec, Alexandre Abraham, and Simon Dollé. 2018. Offline A/B Testing for Recommender Systems. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, WSDM '18, pages 198–206, New York, NY, USA. Association for Computing Machinery.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, and 175 others. 2025. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638.

Huizhong Guo, Tianjun Wei, Dongxia Wang, Yingpeng Du, Ziyan Wang, Jie Zhang, and Zhu Sun. 2026. Think when needed: Model-aware reasoning routing for llm-based ranking. *Preprint*, arXiv:2601.18146.

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

Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yong-Dong Zhang, and Meng Wang. 2020. LightGCN: Simplifying and Powering Graph Convolution Network for Recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20, pages 639–648, New York, NY, USA. Association for Computing Machinery.

Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*, pages 173–182.

Bairu Hou, Yujian Liu, Kaizhi Qian, Jacob Andreas, Shiyu Chang, and Yang Zhang. 2024. Decomposing uncertainty for large language models through input clarification ensembling. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *ICML'24*, pages 19023–19042, Vienna, Austria. JMLR.org.

Eugene Ie, Chih-wei Hsu, Martin Mladenov, Vihan Jain, Sanmit Narvekar, Jing Wang, Rui Wu, and Craig Boutilier. 2019. RecSim: A Configurable Simulation Platform for Recommender Systems. *Preprint*, arXiv:1909.04847.

Muhammad Intizar Ali, Pankesh Patel, John G. Breslin, Ramy Harik, and Amit Sheth. 2021. Cognitive digital twins for smart manufacturing. *IEEE Intelligent Systems*, 36(2):96–100.

Yiqiao Jin, Qinlin Zhao, Yiyang Wang, Hao Chen, Kaijie Zhu, Yijia Xiao, and Jindong Wang. 2024. AgentReview: Exploring Peer Review Dynamics with LLM Agents. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1208–1226, Miami, Florida, USA. Association for Computational Linguistics.

Daniel Kahneman. 2011. *Thinking, Fast and Slow*. Thinking, Fast and Slow. Farrar, Straus and Giroux, New York, NY, US.

Dae Y. Kang, Pamela N. DeYoung, Justin Tantiongloc, Todd P. Coleman, and Robert L. Owens. 2021. Statistical uncertainty quantification to augment clinical decision support: a first implementation in sleep medicine. *npj Digital Medicine*, 4(1).

Wang-Cheng Kang and Julian McAuley. 2018. Self-Attentive Sequential Recommendation. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 197–206.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V. Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, Yuling Gu, Saumya Malik, Victoria Graf, Jena D. Hwang, Jiangjiang Yang, Ronan Le Bras, Oyvind Tafjord, Chris Wilhelm, Luca Soldaini, and 4 others. 2024. Tülu 3: Pushing Frontiers in Open Language Model Post-Training. *Preprint*, arXiv:2411.15124.

Jing Li, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Tao Lian, and Jun Ma. 2017. Neural attentive session-based recommendation. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, CIKM '17, page 1419–1428, New York, NY, USA. Association for Computing Machinery.

Ming Li, Yong Zhang, Shwai He, Zhitao Li, Hongyu Zhao, Jianzong Wang, Ning Cheng, and Tianyi Zhou. 2024. Superfiltering: Weak-to-Strong Data Filtering for Fast Instruction-Tuning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14255–14273, Bangkok, Thailand. Association for Computational Linguistics.

Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. 2023. Towards General Text Embeddings with Multi-stage Contrastive Learning. *Preprint*, arXiv:2308.03281.

Zenan Li, Fan Nie, Qiao Sun, Fang Da, and Hang Zhao. 2025. Uncertainty-aware decision transformer for stochastic driving environments. In *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pages 364–386. PMLR.

Dawen Liang, Rahul G Krishnan, Matthew D Hoffman, and Tony Jebara. 2018. Variational autoencoders for collaborative filtering. In *Proceedings of the 2018 world wide web conference*, pages 689–698.

Hongyang Liu, Zhu Sun, Tianjun Wei, Yan Wang, Jiajie Zhu, and Xinghua Qu. 2026. Diagnostic-guided dynamic profile optimization for llm-based user simulators in sequential recommendation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(18):15306–15314.

Kelong Mao, Jieming Zhu, Jinpeng Wang, Quanyu Dai, Zhenhua Dong, Xi Xiao, and Xiuqiang He. 2021. Simplex: A simple and strong baseline for collaborative filtering. In *Proceedings of the 30th ACM international conference on information & knowledge management*, pages 1243–1252.

Muhammad Hasan Maqbool, Umar Farooq, Adib Mosharrof, AB Siddique, and Hassan Foroosh. 2023. Mobilerec: A large scale dataset for mobile apps recommendation. In *Proceedings of the 46th international ACM SIGIR conference on research and development in information retrieval*, pages 3007–3016.

Jianmo Ni, Jiacheng Li, and Julian McAuley. 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)*, pages 188–197.

Mark O'Neill, Elham Vaziripour, Justin Wu, and Daniel Zappala. 2016. Condensing steam: Distilling the diversity of gamer behavior. In *Proceedings of the 2016 internet measurement conference*, pages 81–95.

Fernando Benjamin Perez Maurera, Maurizio Ferrari Dacrema, Pablo Castells, and Paolo Cremonesi. 2025. Impression-Aware Recommender Systems. *ACM Trans. Recomm. Syst.*

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2023. Direct Preference Optimization: Your Language Model is Secretly a Reward Model. In *Advances in Neural Information Processing Systems 36*, pages 53728–53741. Curran Associates, Inc.

Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, UAI '09, pages 452–461, Arlington, Virginia, USA. AUAI Press.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models. *Preprint*, arXiv:2402.03300.

Yifei Shen, Yongji Wu, Yao Zhang, Caihua Shan, Jun Zhang, B Khaled Letaief, and Dongsheng Li. 2021. How powerful is graph convolution for recommendation? In *Proceedings of the 30th ACM international conference on information & knowledge management*, pages 1619–1629.

Elizaveta Stavinova, Alexander Grigorievskiy, Anna Volodkevich, Petr Chunaev, Klavdiya Bochenina, and Dmitry Bugaychenko. 2022. Synthetic Data-Based Simulators for Recommender Systems: A Survey. *Preprint*, arXiv:2206.11338.

Weiwei Sun, Lingyong Yan, Xinyu Ma, Shuaiqiang Wang, Pengjie Ren, Zhumin Chen, Dawei Yin, and Zhaochun Ren. 2023. Is ChatGPT Good at Search? Investigating Large Language Models as Re-Ranking Agents. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14918–14937, Singapore. Association for Computational Linguistics.

Zhu Sun, Di Yu, Hui Fang, Jie Yang, Xinghua Qu, Jie Zhang, and Cong Geng. 2020. Are we evaluating rigorously? benchmarking recommendation for reproducible evaluation and fair comparison. In *Proceedings of the 14th ACM Conference on Recommender Systems*, RecSys '20, page 23–32, New York, NY, USA. Association for Computing Machinery.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, Johan Ferret, Peter Liu, Pouya Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, and 179 others. 2024. Gemma 2: Improving Open Language Models at a Practical Size. *Preprint*, arXiv:2408.00118.

Mengting Wan and Julian McAuley. 2018. Item recommendation on monotonic behavior chains. In *Proceedings of the 12th ACM Conference on Recommender Systems*, RecSys '18, page 86–94, New York, NY, USA. Association for Computing Machinery.

Lei Wang, Jingsen Zhang, Hao Yang, Zhi-Yuan Chen, Jiakai Tang, Zeyu Zhang, Xu Chen, Yankai Lin, Hao Sun, Ruihua Song, Xin Zhao, Jun Xu, Zhicheng Dou, Jun Wang, and Ji-Rong Wen. 2025. User Behavior Simulation with Large Language Model-based Agents. *ACM Trans. Inf. Syst.*, 43(2):55:1–55:37.

Wenjie Wang, Yiyan Xu, Fuli Feng, Xinyu Lin, Xiangnan He, and Tat-Seng Chua. 2023a. Diffusion Recommender Model. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '23, pages 832–841, New York, NY, USA. Association for Computing Machinery.

Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural graph collaborative filtering. In *Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval*, pages 165–174.

Yancheng Wang, Ziyan Jiang, Zheng Chen, Fan Yang, Yingxue Zhou, Eunah Cho, Xing Fan, Yanbin Lu, Xiaojiang Huang, and Yingzhen Yang. 2024a. RecMind: Large Language Model Powered Agent For Recommendation. In *Findings of the Association* for Computational Linguistics: NAACL 2024*, pages 4351–4364, Mexico City, Mexico. Association for Computational Linguistics.

Yifan Wang, Weizhi Ma, Min Zhang, Yiqun Liu, and Shaoping Ma. 2023b. A Survey on the Fairness of Recommender Systems. *ACM Trans. Inf. Syst.*, 41(3):52:1–52:43.

Zhenduo Wang, Zhichao Xu, Vivek Srikumar, and Qingyao Ai. 2024b. An In-depth Investigation of User Response Simulation for Conversational Search. In *Proceedings of the ACM Web Conference 2024*, WWW '24, pages 1407–1418, New York, NY, USA. Association for Computing Machinery.

Wei Wei, Quoc Le, Andrew Dai, and Jia Li. 2018. AirDialogue: An Environment for Goal-Oriented Dialogue Research. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3844–3854, Brussels, Belgium. Association for Computational Linguistics.

Fangzhao Wu, Ying Qiao, Jiun-Hung Chen, Chuhan Wu, Tao Qi, Jianxun Lian, Danyang Liu, Xing Xie, Jianfeng Gao, Winnie Wu, and Ming Zhou. 2020. MIND: A Large-scale Dataset for News Recommendation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3597–3606, Online. Association for Computational Linguistics.

Haotian Wu, Yingpeng Du, Tianjun Wei, Puay Siew Tan, Jie Zhang, Ong Yew Soon, and Zhu Sun. 2026. Efficient large language models for recommendation: A survey. *TechRxiv*, 2026(0126).

Jiancan Wu, Xiang Wang, Fuli Feng, Xiangnan He, Liang Chen, Jianxun Lian, and Xing Xie. 2021. Self-supervised Graph Learning for Recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21, pages 726–735, New York, NY, USA. Association for Computing Machinery.

Xiwang Yang, Harald Steck, Yang Guo, and Yong Liu. 2012. On top-k recommendation using social networks. In *Proceedings of the Sixth ACM Conference on Recommender Systems*, RecSys '12, pages 67–74, New York, NY, USA. Association for Computing Machinery.

An Zhang, Yuxin Chen, Leheng Sheng, Xiang Wang, and Tat-Seng Chua. 2024a. On Generative Agents in Recommendation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '24, pages 1807–1817, New York, NY, USA. Association for Computing Machinery.

Erhan Zhang, Xingzhu Wang, Peiyuan Gong, Yankai Lin, and Jiaxin Mao. 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*, SIGIR '24, pages 2687–2692, New York, NY, USA. Association for Computing Machinery.

Junjie Zhang, Yupeng Hou, Ruobing Xie, Wenqi Sun, Julian McAuley, Wayne Xin Zhao, Leyu Lin, and Ji-Rong Wen. 2024c. AgentCF: Collaborative Learning with Autonomous Language Agents for Recommender Systems. In *Proceedings of the ACM Web Conference 2024*, WWW '24, pages 3679–3689, New York, NY, USA. Association for Computing Machinery.

Zijian Zhang, Shuchang Liu, Ziru Liu, Rui Zhong, Qingpeng Cai, Xiangyu Zhao, Chunxu Zhang, Qidong Liu, and Peng Jiang. 2025. LLM-Powered User Simulator for Recommender System. In *Proceedings of the Thirty-Four International Joint Conference on Artificial Intelligence*, AAAI '25.

Kesen Zhao, Shuchang Liu, Qingpeng Cai, Xiangyu Zhao, Ziru Liu, Dong Zheng, Peng Jiang, and Kun Gai. 2023. KuaiSim: A comprehensive simulator for recommender systems. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, pages 44880–44897, Red Hook, NY, USA. Curran Associates Inc.

Wayne Xin Zhao, Shanlei Mu, Yupeng Hou, Zihan Lin, Yushuo Chen, Xingyu Pan, Kaiyuan Li, Yujie Lu, Hui Wang, Changxin Tian, Yingqian Min, Zhichao Feng, Xinyan Fan, Xu Chen, Pengfei Wang, Wendi Ji, Yaliang Li, Xiaoling Wang, and Ji-Rong Wen. 2021a. RecBole: Towards a Unified, Comprehensive and Efficient Framework for Recommendation Algorithms. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, CIKM '21, pages 4653–4664, New York, NY, USA. Association for Computing Machinery.

Wayne Xin Zhao, Shanlei Mu, Yupeng Hou, Zihan Lin, Yushuo Chen, Xingyu Pan, Kaiyuan Li, Yujie Lu, Hui Wang, Changxin Tian, Yingqian Min, Zhichao Feng, Xinyan Fan, Xu Chen, Pengfei Wang, Wendi Ji, Yaliang Li, Xiaoling Wang, and Ji-Rong Wen. 2021b. Recbole: Towards a unified, comprehensive and efficient framework for recommendation algorithms. In *CIKM*, pages 4653–4664. ACM.

Tianwen Zhu, Guangyu Wu, Zhiwei Cao, Ruihang Wang, Jimin Jia, Yong Luo, and Yonggang Wen. 2026. Physics-informed multi-task learning for battery state of health prediction with uncertainty quantification. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(2):1659–1667.

A Dataset

A.1 Statistics of Source User Feedback Datasets in RS

In this work, we collect multiple open-source user feedback datasets from various domains to construct USERMIRRORER. Our dataset selection is based on two key criteria: (1) coverage across multiple recommendation domains to enhance the generalization ability of LLMs across different recommendation scenarios, and (2) the inclusion of human-readable item descriptions to align with human cognition. The detailed descriptions of the selected datasets are as follows:

- **Microsoft News Dataset (MIND)** (Wu et al., 2020) is a large-scale news recommendation dataset from *Microsoft News*[5], containing user interactions (impressions, clicks) and metadata (titles, abstracts and categories). It is widely used for studying personalized news recommendations.

- **MovieLens-1M (MovieLens)** (Harper and Konstan, 2015) is a popular dataset for collaborative filtering, featuring around 1 million explicit movie ratings (1–5 scale) from the *MovieLens*[6] platform. It is commonly used for evaluating rating prediction algorithms and user preference modeling.

- **KuaiRec2** (Gao et al., 2022) is a large-scale dataset from *Kuaishou*[7], a short-video platform, containing user interactions such as watch behavior, likes, and shares. This data follows CC-BY-SA 4.0 license. It is valuable for studying sequential recommendations and user engagement in short-video scenarios.

- **Goodreads** (Wan and McAuley, 2018) is a book recommendation dataset from *Goodreads*[8], featuring user ratings, reviews, and reading behaviors. It is useful for analyzing long-term preferences, content-based recommendations, and social influence. The data follows the Apache 2.0 License.

- **MobileRec** (Maqbool et al., 2023) is a mobile app recommendation dataset that records user interactions, including installs, usage duration, and engagement. It supports research on context-aware recommendations and user experience optimization in mobile environments.

- **Steam** (O'Neill et al., 2016) is a game recommendation dataset from the *Steam*[9] platform, containing user purchase history, playtime, and reviews. It is useful for studying implicit feedback and analyzing the impact of game popularity on user engagement.

- **LastFM** (Bertin-Mahieux et al., 2011) is a music recommendation dataset from *Last.fm*[10], capturing user listening history, artist preferences, and tagging behaviors. It supports research on collaborative filtering, content-based recommendations, and dynamic user preference modeling.

- **Amazon** (Ni et al., 2019) is a large-scale e-commerce dataset covering user-item interactions across multiple domains from the e-commerce platform *Amazon*[11]. It is widely used for evaluating product recommendation models. Key domain-specific subsets include:
  - **Fashion**: User purchase histories and reviews for clothing, accessories, and footwear.
  - **Beauty**: Interactions with beauty and personal care products.
  - **Grocery**: User transactions and ratings in the grocery sector.
  - **Office**: Interactions related to office supplies, electronics, and similar items.

  These subsets support research on cross-domain recommendations, personalized ranking, and user behavior modeling in e-commerce.

A.2 Details of the Construction of High-quality User Simulation Dataset

We introduced a framework USERMIRRORER for data distillation in which we sample user simulation scenes and generate decision-making processes to construct a high-quality dataset. For our experiments, we randomly sampled 16,384 user scenes for each domain (except for Amazon-Beauty which has less than 16,384). We then employed two large language models with contrasting capacities: a stronger LLM *Qwen-2.5-32B-Instruct* and a lightweight LLM *Llama-3.2-3B-Instruct*. All inferences were conducted via the *vLLM* framework,

[5] https://microsoftnews.msn.com/
[6] https://movielens.org/
[7] https://www.kuaishou.com/
[8] https://www.goodreads.com/
[9] https://store.steampowered.com/
[10] https://www.last.fm/
[11] https://www.amazon.com/

Dataset	Domain	User Feature	Item Feature	Behavior Feature	Additional Information			Dataset Details		
					Exposure	Review	Social	#User	#Item	#Interaction
MIND	News		✓	✓	✓			307,353	130,379	552,043
MovieLens	Movie	✓	✓	✓				6,040	3,883	1,000,209
KuaiRec2	Short video	✓	✓	✓			✓	7,176	10,728	12,530,806
Goodreads	Book		✓	✓		✓		342,415	89,411	6,995,891
MobileRec	APPs		✓	✓				15,141	10,173	100,000
Steam	Game	✓	✓	✓		✓		250,793	32,132	2,091,415
LastFM	Music		✓	✓			✓	1,892	17,632	70,297
Amazon Fashion	E-commerce		✓	✓		✓		14,791	97,287	56,086
Amazon Beauty	E-commerce		✓	✓		✓		1,620	37,580	14,984
Amazon Grocery	E-commerce		✓	✓		✓		24,076	63,122	378,775
Amazon Office	E-commerce		✓	✓		✓		11,453	64,726	162,297

Table 4: Descriptions of Datasets.

producing $N = 10$ candidate outputs with the sampling parameter temperature set to 1.0 and top-p set to 0.9. After filtering out any samples that exhibited generation errors, we record the valid sample counts and total token consumption, as summarize in the Table 5 below. The resulting distilled dataset comprises 10,000 simulated decision-making scenes for training and 6,400 scenes for testing. Table 6 lists the statistics of the training set and test set.

All data is completely artificially generated. Any similarity in profile descriptions to actual persons, living or dead, is purely coincidental. Our data follows CC-BY-SA 4.0 license. All reuse, modification, and related operations on these datasets strictly adhered to the copyright statements of the data owners.

B Details of Simulation Scene and Behavior Construction

B.1 Construction of Exposure Lists

To simulate the decision-making process in which users are exposed to multiple options before making a final selection, we reconstruct an exposure for each recorded user interaction. The reconstruction of exposures is achieved by using a combination of context-aware, collaborative, and random strategies, thereby providing LLMs with a more human-aligned decision context.

Context-aware Sampling Strategy. This strategy constructs exposure lists by retrieving items that are semantically similar to those the user has previously interacted with. Specifically, we utilize *GTE-Qwen2-1.5B-Instruct* model (Li et al., 2023) to encode the textual content of all items into dense embeddings. Then, for a given target item, we compute cosine similarities in the embedding space and retrieve the top-K most similar items to simulate items likely to co-occur in the user's exposure context.

Collaborative Sampling Strategy. To reflect real-world recommendation scenarios, we employ collaborative filtering signals to model system-driven exposure. We first pre-train a recommendation model, *GF-CF* (Shen et al., 2021), to learn latent interaction patterns from the training data. Given a user, we use this model to rank all items and select the top-K items as candidates that the system would most likely expose.

Random Sampling Strategy. To ensure diversity and account for noise or exploration in RSs, we also include a strategy that randomly samples items the user has not previously interacted with. This allows the exposure list to include both relevant and irrelevant items, better approximating the uncertain and imperfect nature of real-world exposures.

Once the three candidate lists are generated using the above strategies, we combine them into a unified exposures via a randomized iterative procedure (as shown in Algorithm 1). At each iteration, we randomly select one of the three candidate lists with equal probability and extract its top-ranked item. If the item is not yet present in the final list, it is appended. This process repeats until the maximum exposure length N is reached. To ensure that the ground-truth item (i.e., the item the user actually interacted with) appears in the list, we insert it at a randomly chosen position. This strategy ensures that the exposure list reflects a mixture of relevance (from context and collaborative cues), diversity (via random sampling), and decision-critical signal (via inclusion of the ground-truth item), thereby supporting more cognitively plausible user simulations.

In this work, except for the MIND dataset which contains the real exposures, we construct the synthetic exposure lists for other datasets following the above Algorithm 1.

Dataset	# Scene	Qwen-2.5-32B-Instruct				Llama-3.2-3B-Instruct			
		Input		Output		Input		Output	
		Average	Total (M)	Average	Total (M)	Average	Total (M)	Average	Total (M)
KuaiRec2	16,384	1,847	30.3	237	38.9	1,902	31.2	245	40.1
LastFM	16,384	1,111	18.2	231	37.9	1,126	18.5	243	39.8
MobileRec	16,384	1,455	23.8	244	40.0	1,435	23.5	267	43.7
Amazon-Beauty	10,124	1,344	13.6	247	25.0	1,345	13.6	273	27.7
Amazon-Fashion	16,384	1,247	20.4	235	38.5	1,252	20.5	262	42.9
Amazon-Office	16,384	1,623	26.6	254	41.6	1,595	26.1	273	44.7
Amazon-Grocery	16,384	1,572	25.8	238	39.0	1,553	25.5	254	41.6
Movielens	16,384	1,538	25.2	252	41.2	1,498	24.5	266	43.5
Steam	16,383	2,315	37.9	262	43.0	2,268	37.2	290	47.5
Goodreads	16,384	2,694	44.1	267	43.7	2,609	42.7	290	47.5
MIND	16,335	1,129	18.4	178	29.1	1,142	18.7	279	45.6
Overall	**173,914**	**1,625**	**284.4**	**241**	**417.9**	**1,611**	**281.9**	**267**	**464.7**

Table 5: Token usage during the dataset construction phase.

Dataset	Goodreads	KuaiRec2	Steam	Movielens	LastFM	MobileRec	Amazon				MIND	Overall
							Grocery	Beauty	Fashion	Office		
# Training	1,184	1,093	1,055	1,022	737	657	985	570	910	823	964	**10,000**
# Test	640	640	640	640	640	640	320	320	320	320	1,280	**6,400**

Table 6: Statistics of constructed datasets.

B.2 Dataset construction pipeline in USERMIRRORER

We construct a data selection mechanism to extract high-quality samples from massive raw user feedback data that are more valuable for aligning LLM user simulators with the following steps:

1. **Simulation Scene Construction**: Randomly sampling from raw user feedback to construct a batch of user simulation scenes.

2. **Decision Process Generation**: Using LLM A and B with different capabilities to generate N decision processes with predicted behaviors for each sample.

3. **Uncertainty-based Scene Distillation**: Calculate $\Delta_{EU}(\boldsymbol{X}, (A, B))$ via Equation 3, sort scenes by descending Δ_{EU} values, and sequentially retain those with the highest values.

4. **Behavior Reject Sampling**: We discard the scene where all behaviors predicted by stronger LLM are unmatched with the behavior in raw data. For the remaining scene, we mark the decision process according to the matching of behaviors as *accepted* and *rejected*, and formulate the samples with highest confidence as the preference pair.

C Experimental Configurations

C.1 Experiment Environment

All experiments were conducted on a unified setup featuring an *AMD EPYC 7713* 64-core processor and four *NVIDIA A100* 40GB GPUs. LLM fine-tuning (SFT and DPO stages) was performed using *torchtune*[12], and GRPO was implemented with *verl*[13]. For LLM inference, we employed *vllm*[14], and RecBole (Zhao et al., 2021a) was used for training and evaluating the recommendation system models.

C.2 LLM Fine-tuning Configurations

To fine-tune our LLMs on the constructed user feedback dataset, we use the hyperparameters in Table 7. Unspecified settings follow the defaults of the *torchtune* and *verl* frameworks. We consider two training setups:

- **Single-Stage SFT:** For models without the decision-making process, we apply supervised fine-tuning only, treating the task as single-token classification.

- **Two-Stage Fine-Tuning:** For models incorporating decision-making, we perform a standard two-stage pipeline: supervised fine-tuning (SFT) followed by decision-focused

[12]https://github.com/pytorch/torchtune
[13]https://github.com/volcengine/verl
[14]https://github.com/vllm-project/vllm

Algorithm 1: Exposure List Construction

Input: User u, item set $\mathcal{L}$, ground-truth item l^*, maximum exposure length N
Output: Final exposure list $\mathbf{L}$
Generate three candidate lists:
  $\mathbf{L}_{\text{content}} \leftarrow$ content-based top-K ranking of $\mathcal{L}$ for u
  $\mathbf{L}_{\text{collab}} \leftarrow$ collaborative filtering top-K ranking of $\mathcal{L}$ for u
  $\mathbf{L}_{\text{rand}} \leftarrow$ random K permutation of $\mathcal{L}$
Initialize exposure list: $\mathbf{L} \leftarrow \emptyset$
Candidate pool:
$\mathcal{C} \leftarrow \{\mathbf{L}_{\text{content}}, \mathbf{L}_{\text{collab}}, \mathbf{L}_{\text{rand}}\}$
while $|\mathbf{L}| < N$ **do**
  Randomly select a list $\mathbf{L}_i \in \mathcal{C}$ with uniform probability
  Pop the top-ranked item l from $\mathbf{L}_i$
  if $l \notin \mathbf{L}$ **then**
    Append l to $\mathbf{L}$
Randomly select a position $p \in [0, N]$
Insert l^* at position p in $\mathbf{L}$
return L

fine-tuning (e.g., DPO) using our preference data.

We also employ the following two strategies in all training process:

- **Grid Search:** We systematic explore the learning rates and number of epoches to identify the optimal settings of fine-tuning.
- **Mixture of General Purpose Data:** During SFT, we mix in examples from the *Tulu-v3-mixture* (Lambert et al., 2024) dataset to prevent catastrophic forgetting, maintaining a 10:1 ratio of task-specific to general data.

C.3 Configurations of Experiments on Interactive Simulation with RS

We conducted a series of experiments to evaluate whether the simulated user behaviors generated by USERMIRRORER can effectively benefit recommendation models. The experimental procedure consists of three main steps:

Step 1: Training Backbone Recommendation Models. To prevent potential information leakage, we first removed all user data used to train USERMIRRORER from the dataset (Movielens and Steam). The remaining data was then split into training ($\mathcal{D}_{train}$), validation ($\mathcal{D}_{valid}$), and test sets ($\mathcal{D}_{test}$) with a ratio of 8:1:1. We selected eight recommendation models based on diverse technical paradigms and trained them using standard procedures.

- **BPR** (Rendle et al., 2009): Bayesian Personalized Ranking is a pairwise matrix factorization model that learns personalized rankings by optimizing the relative order of observed and unobserved items.
- **LightGCN** (He et al., 2020): Light Graph Convolution Network simplifies traditional GCNs by discarding feature transformations and nonlinearities, focusing solely on neighbor aggregation to improve efficiency and accuracy.
- **MultiDAE** (Liang et al., 2018): The Multinomial Denoising Autoencoder learns robust user representations by reconstructing interaction vectors from corrupted input, effectively modeling non-linear user preferences.
- **DiffRec** (Wang et al., 2023a): Diffusion recommendation model formulates recommendation as a denoising diffusion process, introducing noise to user signals and learning to recover them through a multi-step generative model with structural uncertainty.
- **SGL** (Wu et al., 2021): Self-supervised Graph Learning augments graph-based collaborative filtering with contrastive learning, enforcing consistency between perturbed views of the user-item interaction graph.
- **NeuMF** (He et al., 2017): Neural Matrix Factorization combines matrix factorization with a multi-layer perceptron to capture both linear and non-linear user-item interactions in a unified architecture.
- **NGCF** (Wang et al., 2019): Neural Graph Collaborative Filtering explicitly incorporates high-order connectivity in graph message passing, enhancing user and item embeddings with both structural and semantic signals.
- **SimpleX** (Mao et al., 2021): SimpleX adopts a margin-based contrastive objective and soft negative sampling to simplify self-supervised learning, achieving strong performance without complex augmentations.

Hyperparameter	Epoches	Learning Rate	Batch Size	Optimizer	Weight Decay	Decay Schedule	Warm-up Steps	Other Hyperparameter
SFT	[1,2,3]	[2e-4, 1e-4, 5e-5, 2e-5]	16	AdamW	0.01	Cosine Decay	10	
DPO	[1,2,3]	[2e-4, 1e-4, 5e-5, 2e-5]	16	AdamW	0.05	Cosine Decay	10	beta: 0.1
GRPO	[1,2,3]	1e-5	64	AdamW	0.01	Cosine Decay	10	N: 8; KL Coeff: 1e-3

Table 7: Configurations of LLM fine-tuning stages.

- **SASRec** (Kang and McAuley, 2018): Self-Attention based Sequential Recommendation employs a self-attention mechanism to capture sequential patterns in user behavior, effectively modeling temporal dynamics.

- **NARM** (Li et al., 2017): Neural Attentive Session-based Recommendation models user behavior with an attention mechanism, effectively capturing temporal dynamics and user preferences.

All model implementations and training configurations followed the publicly available library, RecBole (Zhao et al., 2021b). The specific hyperparameter settings are detailed below:

Model	Hyperparameters
BPR	Embedding dimension = 64; Learning rate = 0.001
LightGCN	Embedding dimension = 64; #Layers = 2; Learning rate = 0.001; L_2 regularization = 0.001
MultiDAE	Latent dimension = 64; Hidden layer size = [600]; Learning rate = 0.001; Dropout rate = 0.5
DiffRec	Noise schedule = linear; Noise scale = 0.001; Noise range = [0.0005, 0.005]; DNN layer size = [300]; Learning rate = 0.0001; L_2 regularization = 10^{-5}; Diffusion steps = 5; $w_{\min}$ = 0.1
SGL	Graph type = ED; #Layers = 2; τ = 0.5; Embedding dim = 64; Edge dropout = 0.1; Learning rate = 0.001; $L_2 = 10^{-5}$; SSL weight = 0.05
NeuMF	MF dim = 64; MLP dim = 64; MLP layers = [64, 32, 16]; Dropout rate = 0.1; Learning rate = 0.001;
NGCF	Embedding dim = 64; Message dropout = 0.1; L_2 weight = 10^{-5}; #Layers = 3
SimpleX	Embedding dim = 64; Margin = 0.9; Neg weight = 10; L_2 weight = 10^{-5}; Gamma = 0.5
SASRec	Embedding dim = 64; Inner size = 256; Num layers = 2; Learning rate = 0.001; Dropout rate = 0.5
NARM	Embedding dim = 64; Hidden size = 128; Num layers = 1; Learning rate = 0.001; Dropout rate = 0.5

Table 8: Hyperparameter configurations for backbone recommendation models.

Step 2: Simulating User Behaviors. We utilized each trained backbone recommendation model to generate a candidate exposure list of length 10 for each user. Based on this list, we simulated user behaviors using two models: a base LLM (LLM Base) and a fine-tuned user simulator (SFT+DPO). Each model selects ab item from the exposure list to represent the simulated user–item interaction ($\mathcal{D}_{sim}$). In this experiment, we use *Llama-3.2-3B-Instruct* as the base LLM.

Step 3: Retraining with Simulated Interactions. Step 3: Retraining with Simulated Interactions. The simulated user–item interactions obtained in Step 2 are incorporated into the original training set to create an augmented dataset, $\mathcal{D}^{aug}_{train} = \mathcal{D}_{train} \cup \mathcal{D}_{sim}$. We then retrain the backbone recommendation models using the same procedure outlined in Step 1. By comparing model performance before and after the augmentation, we assess the effectiveness of the simulated user behaviors in improving recommendation quality.

C.4 Detailed Setup of Different Data Selection Strategies

In Section 4.2.1, we compare the proposed uncertainty decomposition based data distillation method with several baselines. Here we list the detailed setup of each baseline. To ensure a fair comparison, all strategies are adopted on the same candidate sample set we processed in Section 3.2, with only the following changes:

- **Random (w/o Decisions)**: We use random selection to replace the difference of epistemic uncertainties $\Delta(\mathbf{X}, (A, B))$ in equation 3, and use the direct behavior prediction as the same as Section 3.1.

- **Random (w/ Decisions)**: We use random selection to replace the difference of epistemic uncertainties $\Delta(\mathbf{X}, (A, B))$ in equation 3.

- **High Accuracy**: We select the samples where the powerful LLM *Qwen-2.5-32B-Instruct* achieves the highest average accuracy with the 10 sampling results.

- **Low Accuracy**: We select the samples where the powerful LLM *Qwen-2.5-32B-Instruct* achieves the lowest average accuracy with the 10 sampling results.

- **Diff. Accuracy**: We select the samples where the powerful LLM *Qwen-2.5-*

32B-Instruct and weak LLM *Llama-3.2-3B-Instruct* achieve the maximum difference in the average accuracy with the 10 sampling results.

- **IFD Score (Li et al., 2024)**: We use the Instruction-Following Difficulty (IFD) score, which measure the difference of the perplexity of the generated decision-making processes with and without the prompt. We use *Llama-3.2-3B-Instruct* to calculate the perplexity.

D Additional Experiment Results

D.1 Interactive Simulation with RS

In addition to the results of the four primary models presented in Table 3, the performance of the remaining six models (BPR, MultiDAE, SGL, NeuMF, NGCF and SimpleX) is reported in Table 9. These additional results serve to further validate the consistency and robustness of our findings across a broader range of model configurations. By including these supplementary models in the appendix, we aim to provide a more comprehensive evaluation and ensure that our conclusions are not limited to a specific subset of experimental settings.

D.2 Factors

To further understand the reasoning patterns exhibited by the fine-tuned LLM during user simulation, we conducted a statistical analysis of the factors referenced by the model when processing recommendation tasks. As illustrated in Figure 7a, stimulus-related factors are dominated by Curiosity *(21.6%)* and Emotional State *(20.4%)*, suggesting that the model frequently simulates users as being driven by intrinsic motivations and affective conditions. Other notable factors include Past Experience *(12.3%)*, Need for Achievement *(10.7%)*, and Time of Day *(9.4%)*, indicating the model's sensitivity to both psychological and contextual signals. The remaining factors, such as Boredom *(5.6%)*, Availability *(3.3%)*, Social Factors *(2.7%)*, Thematic Preferences *(2.5%)*, and Location *(2.3%)*, contribute to a more nuanced understanding of user context, while Others account for 9.2%.

Regarding knowledge-based reasoning, the LLM most frequently relies on Rating Score Distribution *(11.1%)*, Personal Relevance *(9.2%)*, and Emotional Appeal *(8.8%)*. These factors indicate a hybrid reasoning pattern, where both objective signals and affective resonance shape user preferences. Additional sources such as Review Content Sentiment *(6.7%)*, Prior User Experience *(5.1%)*, and Product Quality *(5.1%)* further inform the model's decisions. Less emphasized yet still relevant are factors like Visual Presentation *(4.5%)*, Source Information Clarity *(4.2%)*, Perceived Novelty *(3.8%)*, and Brand Reputation *(3.5%)*. A significant portion *(37.9%)* is grouped under Others, suggesting a broad and diverse interpretation of user-relevant knowledge by the model.

In terms of evaluation strategy, the LLM overwhelmingly adopts a Logical reasoning approach *(89.4%)*, with relatively fewer instances of Intuitive decisions *(10.3%)* and minimal reliance on Others *(0.3%)*. This highlights the model's preference for rational and structured decision-making when simulating user behavior, aligning with deliberative human reasoning processes.

D.3 Feature Ablation Study

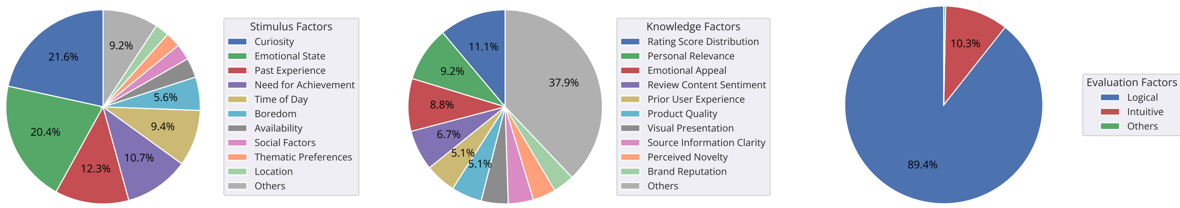
Table 10 and Table 11 present feature ablation results on two distinct domains: the Steam dataset (video games) and the Goodreads dataset (books). Each table reports model accuracy when either only a single feature is retained or when a single feature is removed, highlighting the independent predictive power and marginal contribution of features, respectively.

For the Steam dataset, *Title* and *Tags* stand out as the most informative features both individually and collectively, with Title removal causing the largest accuracy drop. This reflects the crucial role of semantic identifiers and categorical tags in characterizing games, which aligns well with the strong textual and categorical priors that large language models (LLMs) possess about gaming content. Conversely, features such as *History* and *Developer* contribute less individually and appear somewhat redundant when other features are present, possibly due to overlapping information captured by *Tags* and *Player Reviews*.

In contrast, for the Goodreads dataset, features tied to content metadata and contextual information—namely *Title*, Published time, and *Author*—exhibit the highest individual predictive power. Their removal significantly impacts performance, underscoring the importance of temporal and authorial context in book recommendation tasks. Features like *History*, *Description*, and *Tags* show relatively weaker standalone influence and limited marginal contribution, suggesting that, in

Model		Movielens						Steam					
		R@5	R@10	N@5	N@10	M@5	M@10	R@5	R@10	N@5	N@10	M@5	M@10
BPR	Backbone	4.970	8.403	3.138	4.239	2.541	2.990	3.268	5.308	2.004	2.660	1.593	1.861
	Base LLM	5.235	8.725	3.249	4.371	2.602	3.062	3.254	5.316	2.067	2.729	1.680	1.950
	SFT+DPO	**5.520**	**9.256**	**3.531**	**4.727**	**2.883**	**3.370**	**3.290**	**5.381**	**2.171**	**2.844**	**1.805**	**2.081**
	Improve	11.1%	10.2%	12.5%	11.5%	13.5%	12.7%	0.7%	1.4%	8.3%	6.9%	13.3%	11.8%
MultiDAE	Backbone	5.046	8.953	3.314	4.562	2.750	3.255	3.173	5.535	1.973	2.733	1.582	1.894
	Base LLM	5.672	9.503	3.643	4.867	2.982	3.478	3.544	5.664	2.245	3.020	1.759	2.075
	SFT+DPO	**6.392**	**10.015**	**4.244**	**5.410**	**3.543**	**4.021**	**3.678**	**5.974**	**2.423**	**3.160**	**2.015**	**2.316**
	Improve	26.7%	11.9%	28.0%	18.6%	28.9%	23.5%	15.9%	7.9%	22.8%	15.7%	27.3%	22.3%
SGL	Backbone	4.780	8.384	2.981	4.139	2.397	2.870	3.415	5.762	2.143	2.898	1.729	2.039
	Base LLM	5.425	9.048	3.426	4.590	2.773	3.249	3.641	6.193	2.273	3.089	1.827	2.159
	SFT+DPO	**5.728**	**9.674**	**3.709**	**4.975**	**3.050**	**3.566**	**3.897**	**6.105**	**2.476**	**3.183**	**2.012**	**2.300**
	Improve	19.8%	15.4%	24.4%	20.2%	27.3%	24.2%	14.1%	6.0%	15.5%	9.8%	16.4%	12.8%
NeuMF	Backbone	3.699	7.398	2.297	3.472	1.839	2.312	3.254	5.440	2.047	2.745	1.654	1.937
	Base LLM	4.818	**8.441**	3.061	**4.213**	2.482	**2.947**	**3.283**	**5.754**	2.025	**2.823**	1.614	1.943
	SFT+DPO	**4.894**	8.043	**3.111**	4.116	**2.530**	2.938	3.224	5.579	**2.052**	2.808	**1.670**	**1.978**
	Improve	32.3%	8.7%	35.4%	18.5%	37.6%	27.1%	-0.9%	2.6%	0.2%	2.3%	0.9%	2.1%
NGCF	Backbone	5.292	8.763	3.487	4.584	2.900	3.339	3.583	5.952	2.253	3.011	1.819	2.128
	Base LLM	5.634	9.427	3.596	4.802	2.933	3.419	3.656	6.142	2.267	3.069	1.813	2.143
	SFT+DPO	**6.108**	**9.826**	**4.024**	**5.213**	**3.344**	**3.827**	**3.744**	**6.200**	**2.360**	**3.148**	**1.908**	**2.230**
	Improve	15.4%	12.1%	15.4%	13.7%	15.3%	14.6%	4.5%	4.2%	4.7%	4.6%	4.9%	4.8%
SimpleX	Backbone	4.742	8.080	2.908	3.973	2.308	2.739	3.649	6.193	2.292	3.107	1.850	2.182
	Base LLM	4.989	8.062	3.081	4.065	2.463	2.863	3.817	**6.361**	2.342	3.157	1.860	2.192
	SFT+DPO	**5.292**	**8.801**	**3.307**	**4.431**	**2.655**	**3.113**	**4.197**	**6.895**	**2.790**	**3.660**	**2.330**	**2.687**
	Improve	11.6%	8.9%	13.7%	11.5%	15.0%	13.6%	15.0%	11.3%	21.7%	17.8%	25.9%	23.2%

Table 9: Performance comparison of user simulators on recommendation scenarios (Cont.). (In percentage)



(a) Knowledge Factors (b) Stimulus Factors (c) Evaluation Factors

Figure 7: An overview of the three factor categories: knowledge, stimulus, and evaluation factors.

literary domains, user interaction history and descriptive tags may be less decisive compared to core bibliographic information.

These domain-specific patterns reflect differences in the nature of the datasets and the prior knowledge encoded within LLMs. In gaming, categorical labels and titles are rich semantic anchors, supporting detailed user preference modeling. Meanwhile, in books, temporal context and authorship carry substantial weight, likely because publication timing and author identity strongly shape readers' interests and reception. LLMs trained on broad textual corpora inherently capture such domain-dependent nuances, which is reflected in the differential feature importance observed.

Overall, this comparative study highlights that while *Title* is universally critical across domains, the secondary features that most effectively support recommendations differ—*Tags* and *Specifications* for games versus *Published time* and *Author* for books—emphasizing the need for domain-aware feature selection and modeling strategies when leveraging LLMs for recommendation tasks.

D.4 Analysis of Confidence Level Calibration

To examine how training affects model confidence, we analyze the entropy of behavior prediction $\mathcal{H}(P(\boldsymbol{Y}|\boldsymbol{X} \oplus \boldsymbol{C}))$ in equation 2 following the setting in Section 3.2 on the test set across different training stages.

Table 12 presents the entropy of behavior predictions across different training stages. The results reveal a substantial and progressive reduction in prediction entropy as the model undergoes supervised fine-tuning (SFT) and preference optimization (DPO/GRPO). The base model (Llama-3.2-3B-Instruct) exhibits relatively high entropy, indicating considerable uncertainty in its behavior predictions. After SFT, entropy drops dramatically, demonstrating that supervised learning on high-quality

Feature	Accuracy (Retained)	Accuracy (Removed)
History	0.2578	**0.7188**
Price	0.3135	0.7041
Player Reviews	0.3770	0.7002
Specifications	0.3672	0.6807
Tags	0.5059	0.6445
Released Time	0.3389	0.6953
Developer	0.3203	0.6973
Title	**0.5400**	0.5938

Table 10: Accuracy on the Steam dataset under feature ablation settings

Feature	Accuracy (Retained)	Accuracy (Removed)
Rating	0.4658	0.8291
History	0.2637	0.8652
Published at (Interaction Time)	0.6865	0.8535
Author	0.6836	0.8506
Description	0.2656	0.8633
Tags	0.2510	**0.8730**
Title	**0.7080**	0.8291

Table 11: Accuracy on the Goodreads aataset under feature ablation settings.

Model	Entropy
Base (Llama-3.2-3B-Instruct)	1.252 ± 0.138
+ SFT	0.219 ± 0.064
+ SFT + DPO	0.074 ± 0.032
+ SFT + GRPO	0.019 ± 0.009
Teacher (Qwen-2.5-32B-Instruct)	0.011 ± 0.006

Table 12: Entropy of behavior prediction across different training stages on the test set.

decision-making processes significantly improves the model's confidence. Further refinement through DPO reduces entropy, and GRPO achieves even lower entropy, approaching the confidence level of the teacher model (Qwen-2.5-32B-Instruct).

The observed entropy reduction through training stages is both expected and desirable for a user simulator generating complete decision-making processes. Lower entropy indicates that the model produces less ambiguous evaluations and more definitive, high-confidence choices, which aligns with the goal of simulating realistic user behavior where users typically make clear decisions after deliberation. The progressive confidence increase from base model to SFT to preference optimization stages demonstrates that our training pipeline successfully guides the model toward more decisive and human-like decision-making patterns.

D.5 Style Analysis of Decision-Making Processes

To better understand the decision-making processes of the user simulators, we conduct a statistical analysis of the generated decision-making process across different training stages on the test set. We measure several metrics to characterize the decision-making processes, including the length of generations and automated text similarity metrics (BLEU and ROUGE) using the teacher model's decision processes as reference.

Model	ROUGE-L	ROUGE-2	BLEU-2	Length
Llama-3.2-3B-Instruct	0.300	0.239	0.186	273
+ SFT	0.377	0.291	0.264	218
+ SFT + GRPO	0.376	0.287	0.261	217
+ SFT + DPO	0.376	0.307	0.266	238
Qwen2.5-3B-Instruct	0.282	0.176	0.159	179
+ SFT	0.380	0.293	0.264	220
+ SFT + GRPO	0.380	0.292	0.264	220
+ SFT + DPO	0.375	0.313	0.263	253
Qwen2.5-32B-Instruct				218

Table 13: Statistical analysis of decision-making process styles across training stages on the test set. ROUGE and BLEU scores are computed using the teacher model's decision processes as reference.

Table 13 presents the statistical analysis of reasoning styles across different training stages. Several key insights emerge from these results:

Consistent reasoning length. The token length of generated decision-making processes remains relatively consistent across fine-tuned models and the teacher model, indicating that student models do not drastically shorten or extend their generations after training. This suggests that the models learn to generate appropriately detailed decision processes.

SFT learns reasoning structure. Supervised fine-tuning (SFT) dramatically increases ROUGE and BLEU scores compared to base models, demonstrating that students primarily learn the teacher's reasoning style and structural patterns during the SFT phase. This indicates successful knowledge transfer of the decision-making framework.

Preference optimization refines content. The DPO and GRPO stages show largely unchanged text similarity metrics despite significantly improved user preference alignment. This suggests that preference optimization primarily refines the semantic content and decision quality within the already learned reasoning style acquired during SFT, rather than altering the structural patterns of the decision-making process.

These findings reveal that our training pipeline effectively transfers both the reasoning structure (via SFT) and decision quality (via preference optimization) from the teacher model to student models, resulting in user simulators that generate human-like decision-making processes with high confidence and alignment with actual user preferences.

D.6 Domain Generalization of User Simulators

To evaluate the generalizability of our trained user simulator across different domains, we conduct a specialized cross-domain experiment. We split our multi-domain dataset into two disjoint groups and train models on one group while testing on the other, simulating a zero-shot cross-domain transfer scenario.

Experimental Setup. We partition the datasets as follows: **Group 1**: *Goodreads, MobileRec, MIND, Amazon (Fashion, Office)* and **Group 2**: *KuaiRec2, LastFM, MovieLens, Steam, Amazon (Grocery, Beauty)*. Then, we train our model using SFT+DPO on each group separately and evaluate its performance on the held-out group, comparing against the base model, the full-data trained model, and GPT-5 as the baseline.

Table 14 presents the cross-domain generalization results. The cross-domain fine-tuned models demonstrate substantial performance improvements over the base model across nearly all domains. This indicates that the decision-making patterns learned from one set of domains effectively transfer to unseen domains. Notably, in several domains (e.g., Amazon Fashion, Amazon Office, Amazon Grocery, Goodreads), the cross-domain fine-tuned model outperforms GPT-5. This demonstrates that the learned decision-making logic possesses meaningful domain generalization capability and is not merely memorizing domain-specific patterns. While cross-domain performance is strong, it generally falls short of the full-data trained model. This gap highlights the value of our multi-domain data construction approach, which enables the model to capture a more comprehensive and shared understanding of user decision-making across diverse contexts. The performance gap between cross-domain and full-data training varies across domains. For instance, MIND and LastFM show larger gaps, suggesting these domains may have more distinctive user behavior patterns that benefit from in-domain training data.

These findings validate that our framework learns generalizable decision-making patterns rather than domain-specific heuristics, while also confirming the necessity of multi-domain training for optimal performance across diverse recommendation scenarios.

E Prompts and Templates

E.1 Prompt Templates of User Simulation Scene Construction

We designed prompts based on user feedback datasets to construct conversational recommendation scenes. The prompts mainly consist of three components: first, the **user profile**, which introduces the user's personal information; second, the **interaction history**, which outlines the user-item interaction; and third, exposures with the description of each exposed items. Upon the available information in each domain, we provide a detailed template for datasets of each domain as follows.

Model	Group 1 (Trained on Group 2)					Group 2 (Trained on Group 1)					
	Fashion	Office	MIND	Goodreads	MobileRec	Beauty	Grocery	KuaiRec2	LastFM	MovieLens	Steam
Llama-3.2-3B-Instruct	22.3	21.1	19.9	27.1	22.1	21.9	22.3	22.5	18.3	23.3	26.2
+ SFT + DPO (Full Data)	53.9	52.6	34.0	86.9	34.7	49.8	51.0	48.7	35.7	67.3	70.0
+ SFT + DPO (Cross-Domain)	48.8	41.2	26.1	68.5	31.4	37.4	45.8	39.2	28.1	54.6	50.0
GPT-5	46.3	35.0	33.2	78.9	25.3	41.6	38.1	30.8	37.2	61.1	45.2

Table 14: Cross-domain generalization performance. Models trained on one group are evaluated on the other group in a zero-shot transfer setting. Accuracy (%) is reported for all domains.

Template of Movielens

*%**Template of user profile*** *## User Profile*
Gender: {GENDER}
Age: {AGE}
Occupation: {OCCUPATION}
Location: {LOCATION}

*%**Template of interaction history***
A movie viewed {TIME DIFF}
{TITLE} - {GENRES}
Rating: {RATING}/5.0
*%**Template of exposure list***
NEW RELEASE{TITLE} - {GENRES}

Template of Amazon

*%**Template of user profile***
User Profile

*%**Template of interaction history***
A product bought {TIME DIFF}
Product name: {SHORT TITLE}
Category: {CATEGORY}
Price: {REV PRICE}
Ratings: {AVERAGE RATING} ({RATING NUMBER})
My Behavior:
Rating: {RATING}/5
Review: {REVIEW SUMMARY}
{REVIEW TEXT}

*%**Template of exposure list***
Product name: {SHORT TITLE}
Category: {CATEGORY}
Price: {REV PRICE}
Ratings: {AVERAGE RATING} ({RATING NUMBER})

Template of KuaiRec

*%**Template of user profile***
User Profile
Country: China
Registered {REGISTER DAYS} days ago
{FRIEND USER NUM} Friends - {FOLLOW USER NUM} Followers - {FANS USER NUM} Fans
Active Degree: {ACTIVE DEGREE}
Role: {ROLE}

*%**Template of interaction history***
A video watched {TIME DIFF}
Title: {CAPTION} {CATEGORY} {COVER TEXT}
Duration: {VIDEO DURATION}
{PLAY COUNT} plays - {LIKE COUNT} likes - {COMMENT COUNT} comments - {SHARE COUNT} shares
My Behavior: {OPERATION}

*%**Template of exposure list***
Title: {CAPTION} {CATEGORY} {COVER TEXT}
Duration: {VIDEO DURATION}
{PLAY COUNT} plays - {LIKE COUNT} likes - {COMMENT COUNT} comments - {SHARE COUNT} shares

Template of Goodreads

*%**Template of user profile***
User Profile

*%**Template of interaction history***
A book viewed {TIME DIFF}
Title: {TITLE} ({GENRES})
Description: {DESCRIPTION}
Author: {AUTHORS}
Published at {PUBLICATION YEAR} - {PUBLISHER} - {NUM PAGES} pages
Rating: {AVERAGE RATING} - {RATINGS COUNT} ratings
My Behavior: {READ STATUS} {RATING UPDATED} {REVIEW UPDATED}

*%**Template of exposure list***
Title: {TITLE} ({GENRES})
Author: {AUTHORS}
Published at {PUBLICATION YEAR} - {PUBLISHER} - {NUM PAGES} pages
Rating: {AVERAGE RATING} - {RATINGS COUNT} ratings

Template of MobileRec

*%**Template of user profile***
User Profile

*%**Template of interaction history***
{APP NAME} ({AVG RATING}/5.0 - {NUM REVIEWS} reviews)
Category: {APP CATEGORY}
Developer: {DEVELOPER NAME}
Price: {PRICE}
Description: {SHORT DESCRIPTION}
My Rating: {RATING}
My Reviews: {REVIEW}

*%**Template of exposure list***
{APP NAME} ({AVG RATING}/5.0 - {NUM REVIEWS} reviews)
Category: {APP CATEGORY}
Developer: {DEVELOPER NAME}
Price: {PRICE}

Template of LastFM

*%**Template of user profile***
User Profile

*%**Template of interaction history***
An artist listened {TIME DIFF}
Artist name: {NAME}
Tags: {TAGS}
My Behavior: Tagging with {TAG VALUE}

*%**Template of exposure list***
Artist name: {NAME} ({LISTENED FRIENDS})
Tags: {TAGS}

Template of Steam

*%**Template of user profile***
User Profile
A user who has purchased {PRODUCTS} games

*%**Template of interaction history***
A game purchased {TIME DIFF}
Title: {APP NAME} [{GENRES}]
Developed by {DEVELOPER} - Published by {PUBLISHER}
Released: {RELEASE DATE}
Tags: {TAGS}
Specifications: {SPECS}
Player Reviews: {SENTIMENT}
Price: {PRICE}
My Behavior: Played {HOURS PLAYED} hours - Review: {REVIEW TEXT}

*%**Template of exposure list***
Title: {APP NAME} [{GENRES}]
Developed by {DEVELOPER} - Published by {PUBLISHER}
Released: {RELEASE DATE}
Tags: {TAGS}
Specifications: {SPECS}
Player Reviews: {SENTIMENT}
Price: {PRICE}

Template of MIND

*%**Template of user profile*** *## User Profile*

*%**Template of exposure list***
{TITLE} (Category: {CATEGORY})

*%**Template of interaction history***
{TITLE} (Category: {CATEGORY}) { Viewed {TIME DIFF} }

The text within the curly braces {} represents placeholders that need to be filled with specific information from the dataset. Specifically, the "TIME DIFF" field denotes the time difference between the interaction and the current time, where interactions less than 1 minute old are marked as "just now," and older interactions show "X min/hour/day/month/year ago." It is important to note that different datasets contain varying levels of information, and some datasets may be missing certain components, such as the user profile or user behavior description, where we leave them blank.

Here we show an example of a constructed user simulation scene.

Example of the User Simulation Scene

User Profile:
Gender: female
Age: 35-44
Occupation: customer service
Location: Schenectady, NY

User Interaction History:
A movie viewed 4 minutes ago:
Simple Plan, A (1998) - Crime -Thriller
Rating: 4.0/5.0

A movie viewed 3 minutes ago:
Crimson Tide (1995) - Drama -Thriller -War
Rating: 2.0/5.0

A movie viewed 3 minutes ago:
Enemy of the State (1998) - Action -Thriller
Rating: 3.0/5.0

A movie viewed 3 minutes ago:
Die Hard 2 (1990) - Action -Thriller
Rating: 3.0/5.0

A movie viewed 3 minutes ago:
Six Days Seven Nights (1998) - Adventure -Comedy -Romance
Rating: 5.0/5.0

Exposure List:
[A] American Beauty (1999) - Comedy - Drama
[B] [New Release] Hamlet (2000) - Drama
[C] Spawn (1997) - Action - Adventure - Sci-Fi - Thriller
[D] Hana-bi (1997) - Comedy - Crime - Drama
[E] Awakenings (1990) - Drama
[F] Drunks (1997) - Drama
[G] Gandhi (1982) - Drama
[H] The Governess (1998) - Drama - Romance
[I] Whatever (1998) - Drama
[J] Firelight (1997) - Drama
[K] The Best Man (1999) - Drama
[L] My Favorite Year (1982) - Comedy

E.2 Instruction Prompts for User Simulation Scene

We design two types of prompts to guide the LLM: (1) direct prediction, and (2) prediction with decision-making process. The direct prediction prompt directly asks it to predict the user's next action based on the user simulation scene. And the prompt with decision-making process introduces an intermediate reasoning stage inspired by the cognitive process of human decision-making.

Prompt for Direct Prediction

You are a sophisticated user behavior emulator. Given a user profile and an exposure list, generate a user behavior in the given exposure list. Each choice in the exposure list is indicated by a alphabetic identifier [A], [B], [C], etc. Your output should be a choice identifier from the exposure list, for example, "Behavior: [G]".
{User Profile}
{Interaction History}
{Exposures}

Prompt for Behavior Prediction with Decision-making Process

You are a sophisticated user behavior emulator, tasked with simulating user responses within a general recommendation context. Given a user profile and an exposure list, generate a detailed, first-person intent statement that reflects the user's behavior. Your simulations should be adapted for diverse recommendation domains such as media, businesses, and e-commerce.

Intent Structure and Content:
The intent should be structured as a logical progression through the following stages, each marked by a corresponding label:

- ***Stimulus:*** *[Describe the initial motivation or need that initiates the user's thought process. This should connect to their profile's spatial, temporal, thematic preferences, causal, and social factors.].*
  - ***Stimulus Factors:*** *[List 1-3 most relevant factors from: Internal States (Boredom, Hunger, Thirst, Fatigue/Restlessness, Emotional State, Curiosity, Need for Achievement, Inspiration), External Cues (Time of Day, Day of Week, Weather, Location, Social Factors, Special Occasion, Notification, Advertising, Financial Situation, Availability)].*
- ***Knowledge:*** *[Describe the user's thought process as they gain knowledge from the exposure list. Highlight specific attributes of the options that resonate with the user's preferences, drawing on the user profile.].*
  - ***Knowledge Factors:*** *[List 2-4 most influential factors from: Product/Service Attributes (Price, Quality, Features, Convenience, Novelty, Brand Reputation, Personal Relevance (Functional, Thematic, Identity-Based), Emotional Appeal, Time Commitment, Risk), Information Source & Presentation (Visual Presentation, Recommendation Source, Review Content/Sentiment, Rating Score/Distribution, Social Proof), User's Prior Knowledge (Past Experience, User Preferences/History)].*
- ***Evaluation:*** *[Explain the user's internal justification for their preference.]*
  - ***Evaluation Style:*** *[Specify 1 style of the evaluation process, such as Logical, Intuitive, Impulsive, Habitual].*

Output Format:

Thought:

-Stimulus: [Stimulus Description]

-Stimulus Factors: [Factor 1], [Factor 2]

-Knowledge: [Knowledge Description]

-Knowledge Factors: [Factor 1], [Factor 2], [Factor 3]

-Evaluation: [Evaluation Description]

-Evaluation Style: [Evaluation Style]

Behavior: *[Behavior]*

Constraints:

- *While multiple behaviors might be considered in the early stages, the final intent and decision should align with a* ***single*** *behavior.*
- *The behavior can be represented by a single label from the choices in the exposure list, enclosed in square brackets (e.g., [X]).*
- *Use "I" to reflect the first-person perspective of the user.*

{User Profile}
{Interaction History}
{Exposures}

F Case Study

In the following, we present specific case studies to illustrate how a fine-tuned LLM can effectively perform user simulation tasks. Below shows the model input, which includes the task-specific prompt (see Appendix E.1), the user's profile and interaction history, as well as an exposure list of available options.

Example of User Simulation Scene

You are a sophisticated user behavior emulator... (System message, omitted. See Appendix E.1)

 # User Profile
Gender: male
Age: 35-44
Occupation: other
Location: Cincinnati, OH

 # History
A movie viewed just now, Pink Floyd - The Wall (1982) Drama|Musical|War, My Behavior: Rating: 3.0/5.0.
A movie viewed just now, My Bodyguard (1980) Drama, My Behavior: Rating: 3.0/5.0.
A movie viewed just now, Agnes of God (1985) Drama|Mystery, My Behavior: Rating: 3.0/5.0.
A movie viewed just now, Color of Money, The (1986) Drama, My Behavior: Rating: 4.0/5.0.
A movie viewed just now, Betrayed (1988) Drama|Thriller, My Behavior: Rating: 3.0/5.0.
A movie viewed just now, Radio Days (1987) Comedy|Drama, My Behavior: Rating: 4.0/5.0.
A movie viewed just now, Mosquito Coast, The (1986) Drama, My Behavior: Rating: 3.0/5.0.
A movie viewed just now, Brother from Another Planet, The (1984) Drama|Sci-Fi, My Behavior: Rating: 4.0/5.0.

 # Exposure List
A. Psycho (1998) - Crime|Horror|Thriller
B. Footloose (1984) - Drama
C. Resurrection Man (1998) - Drama|Thriller
D. Curtis's Charm (1995) - Comedy|Drama
E. Psycho II (1983) - Horror|Thriller

The model's output consists of the simulated user's decision-making process and the final selected option.

Example of Output

Thought:
Stimulus: *I'm feeling a bit restless and in the mood for a movie to unwind after a long day. I'm in the mood for something that's not too heavy but also not just another typical comedy or action film.*
Stimulus Factors: *Emotional State, Time of Day*

Knowledge: *From the exposure list, I noticed that "Footloose" (1984) and "Resurrection Man" (1998) are dramas, which aligns well with my recent movie preferences. "Footloose" seems to have a strong narrative and a mix of drama and possibly some action, which I find appealing. "Resurrection Man" is a thriller and also a drama, which also fits my current mood.*
Knowledge Factors: *Product/Service Attributes, Information Source & Presentation*

Evaluation: *I think "Footloose" is the better choice because it seems to have a good balance of drama and a touch of action, which I think will be a nice break from the heavy dramas I've been watching. The action elements in "Resurrection Man" are intriguing, but I think "Footloose" will be more in line with my recent viewing history and preferences.*
Evaluation Style: *Logical*

Behavior: B

The results demonstrate that the fine-tuned LLM is capable of analyzing the recommendation task from both the *stimulus* and *knowledge* perspectives, thereby exhibiting a human-understandable reasoning process. Ultimately, it evaluates the available options based on logical reasoning and arrives at the most accurate selection.

In Appendix D.2, we conduct a statistical analysis of the factors referenced by the model when processing recommendation tasks. Here we provide serveral additional examples of the decision-making process of the user simulator.

Example of Decision-Making Process

Thought:
Stimulus: *I am feeling a bit restless and in need of some entertainment after a long day at work. I am in the mood for something that resonates with my past viewing preferences and has a good mix of drama and comedy.*
Stimulus Factors: *Emotional State, Past Experience*

Knowledge: *From the exposure list, I have considered a few options. "American Graffiti" stands out as it combines drama and comedy, genres that I have rated highly in the past. "Outlaw Josey Wales" is another interesting option, but the Western genre is not as appealing to me as the dramas or comedies. "Blood Feast" is a horror film, which is not my preferred genre. "Little Big League" is a children's movie, which is not aligned with my current mood.*
Knowledge Factors: *Personal Relevance, Past Experience, User Preferences/History*

Evaluation: *I am leaning towards "American Graffiti" because it offers a blend of drama and comedy, genres that I have found to be more engaging and relatable. The other options do not align as closely with my current mood or past viewing history.*
Evaluation Style: *Logical*

Example of Decision-Making Process

Thought:
Stimulus: *I'm feeling a bit nostalgic and in the mood to discover some new music that aligns with my taste in alternative and indie rock. I'm also influenced by the social aspect of seeing artists my friends have listened to.*
Stimulus Factors: *Emotional State, Social Factors*

Knowledge: *From the exposure list, I noticed that Belle and Sebastian have tags that resonate with my interest in alternative and indie rock, while The Microphones also caught my attention with their post-rock and rock tags. I also noted that one of my friends has listened to The Microphones, which adds to the appeal.*
Knowledge Factors: *Product/Service Attributes (Genre, Social Proof), Information Source & Presentation (Tags, Social Factors)*

Evaluation: *I feel that Belle and Sebastian have a sound that aligns well with my taste in alternative rock, and the fact that a friend has also listened to The Microphones makes me curious to explore their music as well.*
Evaluation Style: *Logical*