RecLAIF: Reinforcement Learning from AI Feedback for Recommendation Systems

Xiaoxin He
National University of Singapore
Singapore
he.xiaoxin@u.nus.edu

Nurendra Choudhary
Amazon
Palo Alto, CA, USA
nurendc@amazon.com

Jieyi Jiang
Amazon
Seattle, WA, USA
jianjiey@amazon.com

Edward W Huang
Amazon
Palo Alto, CA, USA
ewhuang@amazon.com

Bryan Hooi
National University of Singapore
Singapore
dcsbhk@nus.edu.sg

Xavier Bresson
National University of Singapore
Singapore
xaviercs@nus.edu.sg

Karthik Subbian
Amazon
Palo Alto, CA, USA
ksubbian@amazon.com

Abstract

Traditional recommendation systems (RecSys) face critical challenges, including limited semantic understanding of user preferences, lack of explainability, and poor adaptability to evolving user needs. These limitations hinder their ability to provide personalized, context-aware recommendations and can erode user trust. Recent work has explored the use of Large Language Models (LLMs) to address these challenges. However, a key bottleneck in such approaches is the need for human-annotated data to further fine-tune the LLM world knowledge, which is both costly and time-consuming. To overcome this limitation, we propose leveraging AI feedback as an efficient alternative to human supervision.

In this work, we propose RecLAIF (Reinforcement Learning from AI Feedback for Recommendation Systems), a novel framework that optimizes RecSys through AI-generated feedback. RecLAIF employs LLMs in two roles: a recommender that generates personalized recommendations enriched with key reasoning features, and a judge that evaluates outputs based on relevance, diversity, and explainability. Using Direct Preference Optimization (DPO), RecLAIF iteratively fine-tunes the recommender to align with user-centric preferences, dynamically adapting to evolving needs.

Experiments on three real-world datasets demonstrate that RecLAIF outperforms traditional and LLM-based RecSys. Notably, our 7B model surpasses Claude 3 Sonnet (a model with hundreds of billions of parameters), achieving a 13.9% improvement in relevance while maintaining comparable diversity and explainability.

KDD'25, Toronto, ON, Canada

CCS Concepts

• **Information systems** → **Recommender systems**; • **Computing methodologies** → **Natural language processing**; **Reinforcement learning**.

Keywords

Large Language Models, Recommendation Systems, Reinforcement Learning from AI Feedback

ACM Reference Format:
Xiaoxin He, Nurendra Choudhary, Jieyi Jiang, Edward W Huang, Bryan Hooi, Xavier Bresson, and Karthik Subbian. 2025. RecLAIF: Reinforcement Learning from AI Feedback for Recommendation Systems. In *Proceedings of the KDD 2025 Workshop on Online and Adaptive Recommender Systems (OARS-KDD'25), August 03–07, 2025, Toronto, ON, Canada.* ACM, New York, NY, USA, 10 pages.

1 Introduction

Reinforcement Learning with Human Feedback (RLHF) has proven to be an effective method for aligning Large Language Models (LLMs) with human preferences [5, 15]. RLHF has been widely applied to improve models such as GPT-3 and its successors, enhancing their ability to generate outputs that better align with human expectations, reduce harmful content, and follow complex instructions. However, gathering human feedback at scale presents significant challenges due to its high cost and time-consuming nature. Moreover, as LLMs evolve and application domains shift, previously collected feedback data may become outdated, diminishing its utility for continued model refinement.

To address these limitations, recent advancements have introduced **Reinforcement Learning from AI Feedback (RLAIF)**, a paradigm where feedback is provided by another AI system rather than humans [2, 9, 24]. RLAIF offers a scalable and cost-effective alternative to RLHF, delivering consistent and high-quality feedback for training LLMs. These studies have shown that AI-generated

Figure 1: Comparison between Traditional RecSys and *RecLAIF* for the query "Baby Gift Christmas." Traditional RecSys fails to capture query intent, lacking semantic understanding, explainability, and adaptability to the global Christmas trend. *RecLAIF* identifies key contextual features (e.g., Christmas theme, gift-readiness) and ensures recommendations align with the seasonal trend, making them more relevant and explainable.

feedback can match or even surpass human evaluations in quality, while significantly reducing the time and resource constraints traditionally associated with RLHF.

While RLAIF has demonstrated promise in improving general-purpose LLMs, its potential in domain-specific applications, such as recommendation systems (RecSys), remains under-explored. RecSys play a pivotal role across domains such as e-commerce [7, 18], entertainment [3, 4], and social networks by tailoring content to users based on their preferences and needs. Despite their ubiquity, conventional RecSys face several limitations: (1) **Semantic Understanding Gap**: Traditional RecSys often struggle to comprehend the complex semantics in user queries and item attributes, leading to inaccurate recommendations. For example, in Figure 1, the query *"Baby Gift Christmas"* contains multiple semantic aspects—age appropriateness (baby), thematic relevance (Christmas), and occasion-specific intent (gift). Traditional RecSys primarily focus on generic baby toys, failing to capture the holiday theme and gift suitability. (2) **Lack of Explainability**: Conventional models primarily output just the items without providing insight into why they were recommended. In Figure 1, while the traditional RecSys suggests baby toys, it does not justify why these items are relevant. (3) **Limited Adaptability**: Many RecSys fail to adapt effectively to dynamic user preferences, relying on static feedback or batch updates that lag behind real-world trends. These challenges necessitate a paradigm shift toward more user-centric, interpretable, and adaptive recommendation systems. This leads us to a key research question: ***Can RLAIF be effectively applied to the recommendation setting to address these limitations?***

Applying RLAIF to recommendation introduces unique challenges. Unlike general RLAIF, recommendation systems operate in a distinct setting where certain types of human data, such as user-item purchase histories, are relatively easy to collect at scale and can provide valuable insights for improving recommendations. However, other annotations, such as the diversity of recommended items, are equally important for delivering satisfying recommendations but are significantly harder to obtain. Some features such as explainability (i.e., the rationale behind specific recommendations) are practically non-existent. Generating human annotations for such nuanced metrics requires considerable time and cost. With AI feedback serving as a viable alternative, effectively obtaining and integrating hybrid feedback from both humans and AI represents a unique challenge in the recommendation setting.

To address these challenges, we propose **RecLAIF** (Reinforcement Learning from AI Feedback for Recommendation Systems), a novel framework that leverages the capabilities of LLMs in two roles: as a recommender and as a judge. The LLM-based recommender identifies key features from user queries, interaction histories, and item attributes to generate contextually relevant and personalized recommendations. The LLM-based judge evaluates the quality of recommendations using metrics such as relevance, diversity, and explainability. By using metrics that are more important to users compared to final outcomes such as click-through rates, we are designing a user-centric evaluation paradigm. Using Direct Preference Optimization (DPO), we achieve continuous improvement by leveraging feedback from the judge. Feedback loops iteratively

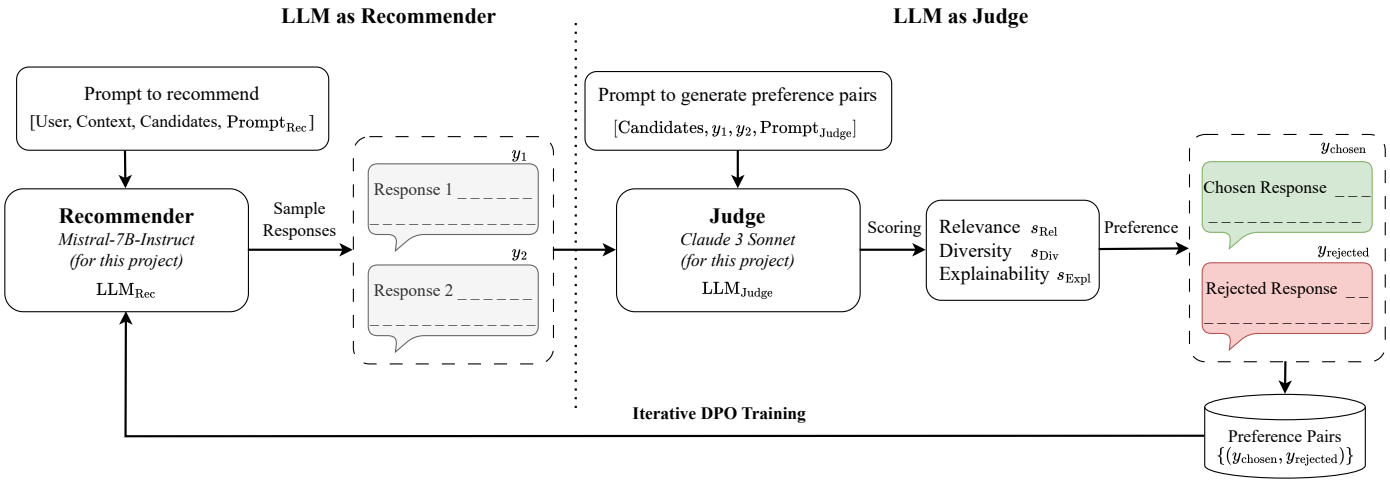

Figure 2: Overview of the *RecLAIF* Framework. The proposed framework leverages LLMs in dual roles: as a recommender to generate candidate responses and as a judge to evaluate outputs based on relevance, diversity, and explainability. The judge produces preference pairs by comparing sampled responses and identifying the chosen and rejected outputs. These preference pairs are used to iteratively fine-tune the recommender using Direct Preference Optimization (DPO).

refine the recommender, enabling it to adapt to evolving user preferences and data landscapes while reducing reliance on expensive human evaluations. The overview is shown in Figure 2.

RecLAIF effectively addresses core limitations of traditional RecSys. It enhances semantic understanding by leveraging LLMs' ability to process complex queries, improves explainability by explicitly identifying key reasoning features, and increases adaptability through continuous learning from AI-generated feedback.

Through extensive experiments on three real-world datasets, we demonstrate that RecLAIF significantly outperforms baseline recommendation systems in terms of accuracy, diversity, and explainability. These results highlight the transformative potential of RLAIF in modernizing recommendation systems and bridging the gap between user-centric evaluation and dynamic adaptability.

The contributions of this paper are summarized as follows:

- Introduction of *RecLAIF* Framework: We propose a novel dual-LLM framework, with LLMs serving as both a recommender and a judge, enabling iterative optimization using RLAIF principles.
- Advancing Explainability: By explicitly identifying and presenting the key features underlying recommendations, RecLAIF enhances the interpretability and transparency of recommendation outputs.
- Improved Metrics and Adaptability: Through AI-driven evaluation and optimization, RecLAIF achieves superior performance in terms of relevance, diversity, and explainability compared to traditional RecSys methods.
- Comprehensive Experimental Validation: We conduct extensive experiments on real-world datasets to demonstrate the effectiveness of RecLAIF, providing benchmarks and insights for future research.

2 Related Work

Large Language Models for Recommendation Systems (RecSys). There are three representative modeling paradigms for LLMs in RecSys [12, 13]: (1) LLM as Embedder. LLMs serve as feature encoders, transforming user and item textual features into vector representations that are subsequently fed into traditional RecSys [16, 29, 31, 32]. (2) LLM as Explainer. LLMs are used to reason over text features of items and users, such as predicting user interest and summarizing item features. These explanations generated by LLMs are used as auxiliary information; they are encoded into embeddings and integrated into RecSys [11, 19, 27, 30]. (3) LLM as Recommendation System. This kind of method either converts the recommendation task into a prompt for LLMs or uses the user and item embeddings as soft prompts to align the RecSys with LLM [6, 10, 14, 22]. In this work, we explore the use of LLM as a judge for RecSys. Recent studies [21, 25] have also employed LLMs as a judge for recommendation, but their focus has been limited to evaluating text relevance. In contrast, our approach leverages LLMs as a judge for more complex tasks—evaluating not only relevance but also diversity and explainability, which are more user-centric metrics, thereby fully utilizing the capabilities of LLMs.

Reinforcement Learning from AI Feedback (RLAIF). LLMs trained purely on large-scale text corpora often exhibit behaviors misaligned with user preferences and societal values [28]. Reinforcement Learning from Human Feedback (RLHF) [5, 15] addresses this issue by incorporating human judgments into the training loop to better align model outputs with desired responses. The typical RLHF pipeline consists of three main steps: (1) *Supervised Fine-Tuning (SFT):* An LLM is first fine-tuned on a dataset of instruction-response pairs, often curated or partially annotated by humans. (2) *Reward Model (RM) Training:* Human annotators compare pairs of model outputs to indicate which is preferable, creating a preference dataset used to train a reward model that predicts human-preferred outputs.

(3) *Policy Optimization via RL:* The LLM is then fine-tuned against the learned reward model using a reinforcement learning algorithm (e.g., DPO [17] or PPO [20]), guiding it to produce responses that maximize the learned reward signal.

RLHF has been shown to substantially improve the quality, helpfulness, and safety of model outputs, with one notable success being its use in ChatGPT [5, 15]. However, a major bottleneck of RLHF is the cost and time required to collect high-quality human labels at scale, limiting broader applicability.

Reinforcement Learning from AI Feedback (RLAIF) [2, 9] was introduced as an alternative that trains the reward model on preferences generated by an off-the-shelf LLM instead of humans, often achieving performance on-par with using human feedback. Direct-RLAIF [9] further simplifies RLAIF by circumventing reward model training altogether; it obtains rewards directly from an off-the-shelf LLM during reinforcement learning, reducing the overhead of collecting and processing preference data.

3 Formalization

This section establishes the notation and formalizes key concepts related to retrieval, recommendation tasks, LLMs for recommendations, and fine-tuning LLM-based recommenders.

Recommendation Tasks. The core task of recommendation systems is to provide a ranked list of items $\{i_k\}_{k=1}^{N}, i_k \in \mathcal{I}$ for a user $u \in \mathcal{U}$ given a specific context $c \in \mathcal{C}$. Here, $\mathcal{U}$ and $\mathcal{I}$ represent the universal sets of users and items, respectively. We formalize the goal as follows:

$$\{i_k\}_{k=1}^{N} \leftarrow \text{RecSys}(u, c, \mathcal{I}), \quad u \in \mathcal{U},\ i_k \in \mathcal{I}, c \in \mathcal{C}.$$

For sequential recommendation tasks, the context c represents the user's interaction history (e.g., purchase history), and the objective is to predict the next item to be purchased, where $N = 1$. For retrieval tasks, the context c is typically a user query, and the objective is to retrieve a list of items $\{i_k\}_{k=1}^{N}, i_k \in \mathcal{I}$ that best matches the query.

LLM as Recommender. Due to the input token limit of LLMs, it is often infeasible to include all items in the input window simultaneously. A common practice is to sample a subset of items $\mathcal{I}' \subset \mathcal{I}$ as the candidate set. An LLM acts as a recommendation system by taking the context c (e.g., a user's purchase history or a query) and the sampled candidate set $\mathcal{I}'$ as input, along with a prompt P_{Rec} that specifies the recommendation task. The LLM processes these inputs to generate a ranked list of items that best align with the user's preferences. Formally, the prediction is defined as:

$$\{i_k\}_{k=1}^{N} \leftarrow \text{LLM}_{\text{Rec}}^{\theta}(u, c, \mathcal{I}', P_{\text{Rec}}),$$

where θ represents the parameters of the LLM, c is the context, $\mathcal{I}'$ is the sampled candidate set, and P_{Rec} is the instruction prompt guiding the task.

Fine-tuning LLM-based Recommenders. Supervised fine-tuning (SFT) is a crucial step in aligning LLMs with recommendation tasks. By fine-tuning on recommendation-specific datasets, the LLM learns to generate accurate recommendations using instruction-based training pairs and a language modeling loss.

Given a user u with a context c (e.g., a user's purchase history or a query) and a candidate set $\mathcal{I}' \subset \mathcal{I}$, the training pairs (x_u, y_u) are defined as (we omit subscript 'u' in x_u, y_u for conciseness):

$$x = [u, c, \mathcal{I}', P_{\text{Rec}}], \quad y = \{i_k\}_{k=1}^{N}.$$

The fine-tuning objective minimizes the cross-entropy loss between the predicted output tokens and the target output y. Formally, the objective is:

$$\mathcal{L}_{\text{SFT}}(\theta) = -\mathbb{E}_{(x,y)\sim P_{\text{train}}} \sum_{t=1}^{|y|} \log P_\theta\left(y_t \mid x, y_{<t}\right), \tag{1}$$

where $|y|$ is the length of the target output y, y_t is the t-th token, and $y_{<t}$ represents all preceding tokens. $P_\theta\left(y_t \mid x, y_{<t}\right)$ is the probability of generating token y_t conditioned on the input x and its preceding tokens.

DPO Training. Direct Preference Optimization (DPO) is a fine-tuning method that directly optimizes the likelihood of preferred outputs over less preferred ones without requiring an explicit reward model. For a preference pair $(y_{\text{chosen}}, y_{\text{rejected}})$ generated by an LLM judge, the DPO objective aligns the model p_θ with user-centric preferences while ensuring stability through a reference model p_{ref}.

The DPO loss function is defined as:

$$\mathcal{L}_{\text{DPO}}(\theta; \pi_{\text{ref}}) = -\mathbb{E}_{(x, y_{\text{chosen}}, y_{\text{rejected}})\sim P_{\text{pair}}} \log \sigma\left(\beta \log \frac{\pi_\theta(y_{\text{chosen}} \mid x)}{\pi_{\text{ref}}(y_{\text{chosen}} \mid x)} - \beta \log \frac{\pi_\theta(y_{\text{rejected}} \mid x)}{\pi_{\text{ref}}(y_{\text{rejected}} \mid x)}\right), \tag{2}$$

Where $\pi_\theta(y \mid x)$ represents the probability of output y given input x under the fine-tuned model p_θ, and $\pi_{\text{ref}}(y \mid x)$ denotes the probability of output y under a reference model p_{ref}, such as a pre-trained or baseline model. The terms y_{chosen} and y_{rejected} correspond to the preferred and less-preferred outputs, respectively. The parameter β acts as a scaling factor that controls the strength of preference enforcement, balancing the fine-tuned model's adherence to the preference signal. Finally, the sigmoid function $\sigma(\cdot)$ ensures numerical stability during optimization by mapping values to a bounded range.

The DPO objective encourages the model p_θ to increase the likelihood of the preferred output y_{chosen} relative to the reference model p_{ref} while decreasing the likelihood of the less preferred output y_{rejected}. By leveraging preference pairs, DPO bypasses the need for an explicit reward model, simplifying the training pipeline.

This approach ensures the fine-tuned model progressively aligns with user-centric preferences while maintaining smooth and stable updates through its dependence on the reference model.

4 Method

In this section, we present the proposed *RecLAIF* framework, which tackles the challenges introduced in Section 1 by leveraging LLMs in two roles: as a Recommender for generating personalized, explainable recommendations and as a Judge for providing structured AI feedback to iteratively refine the system. The framework integrates hybrid supervision—leveraging both structured human data (e.g., user-item purchase histories) and AI-generated feedback—to improve recommendation quality. An overview of the framework is illustrated in Figure 2.

4.1 LLM as Recommender

We reformulate the recommendation task as a natural language problem to leverage the reasoning capabilities of large language models.

The LLM takes an input prompt x, which encodes the context c, e.g., a user's purchase history, the sampled candidate set $\mathcal{I}'$, and a prompt P_{Rec} that specifies the recommendation task. The goal is to generate: 1) Key features K_u, which explain the reasoning behind the recommendations, and 2) Top-k recommendations R_u, ranked based on their relevance to the user's preferences.

Formally, the LLM generates:

$$(R_u, K_u) \leftarrow \text{LLM}^{\theta}_{\text{Rec}}([u, c, \mathcal{I}', P_{\text{Rec}}]), \tag{3}$$

where θ represents the LLM parameters, and $[\cdot]$ denotes the concatenation of inputs.

Instruction Design. The instruction P_{Rec} guides the LLM to simultaneously identify key features and rank the most relevant items from the candidate set. The goal is to structure the task in a way that ensures the LLM considers user preferences, maintains diversity in recommendations, and produces clear explanations for its outputs. Below, we provide a detailed example of the instruction prompt used in a music recommendation task [3]:

Prompt Template for LLM as Recommender

You are a music recommendation engine. Your task is to suggest the best artists or bands based on the user's listening history.

Listening History:
Depeche Mode, The Birthday Massacre, No Doubt, Crystal Castles, Phoenix, Emilie Simon, Eths

Available Artists/Bands:
Bassnectar, Julian Casablancas, Earth Crisis, Angus & Julia Stone, Buju Banton, Head Automatica, Snow Patrol, 10 Years, Tarja, Gentle Giant, Astor Piazzolla, Black Sun Empire, Holly Brook, The Horrors, Catherine Wheel, 45 Grave, Impending Doom, Band of Skulls, Hannah Montana, Crossfade

Task:
1. Identify 2-3 key preferences based on the user's history (e.g., genre, style, mood, instrumentation).
2. Select the top three artists or bands from the candidates that best match the key preferences. Ensure the recommendations are relevant, diverse, and ordered from most to least likely to be enjoyed. Provide the artist/band name and a brief reason for each.

Output Format:
"""
Key Preferences:
- Preference 1: [Description]
- Preference 2: [Description]
[...]

Recommended Artists/Bands (from most to least likely):
1. [Artist/Band Name] - [Reason for Recommendation]
2. [Artist/Band Name] - [Reason for Recommendation]
[...]
"""

Example Output. The following output demonstrates the structured reasoning and explainability provided by the LLM_{Rec}.

Example Response for LLM as Recommender

Key Preferences:
- Preference 1: Alternative/Indie Rock and Electronic/Synth-pop genres
- Preference 2: Dark, moody, and atmospheric vibes
- Preference 3: Female vocals

Recommended Artists/Bands (from most to least likely):
1. The Horrors - Their music blends alternative rock with electronic and synth-pop elements, creating a dark and atmospheric sound that aligns with the user's preferences.
2. Band of Skulls - This alternative rock band has a moody and atmospheric sound, often incorporating electronic and synth elements, which could appeal to the user's taste.
3. Catherine Wheel - With their blend of alternative rock and electronic influences, as well as the presence of female vocals, Catherine Wheel could be an interesting discovery for the user.

The recommendations focus on aligning with the user's identified preferences for genre, mood, and vocal characteristics while ensuring diversity in the suggested bands. The top recommendation, *The Horrors*, strongly aligns with all three preferences, while *Band of Skulls* and *Catherine Wheel* provide complementary yet relevant options.

This structured output improves both explainability and user trust by explicitly presenting the reasoning behind the recommendations and highlighting the most relevant features.

Fine-tuning the Recommender. To enhance the reasoning capabilities of the recommender (Mistral-7B-Instruct [8] in this work), we fine-tune it on responses generated by a stronger LLM (e.g., Claude 3 Sonnet) using Supervised Fine-Tuning (SFT). This process distills knowledge from a more powerful model into a deployable, cost-efficient recommender.

4.2 LLM as Judge

We introduce another LLM (Claude 3 Sonnet for this project), LLM-Judge , to evaluate the quality of outputs generated by the LLM-Rec based on three dimensions: relevance, diversity, and explainability. These evaluations are used to determine preference pairs $(y_{\text{chosen}}, y_{\text{rejected}})$, providing feedback for optimizing the recommender system.

Diverse Output Sampling. To generate candidate outputs, the LLM_{Rec} is prompted with input x. Sampling with high temperature T and nucleus sampling (threshold p) introduces variability into the outputs:

$$y_1 \sim \text{LLM}_{\text{Rec}}(x; T, p), \quad y_2 \sim \text{LLM}_{\text{Rec}}(x; T, p). \tag{4}$$

Scoring and Preference Pair Construction. The $\text{LLM}_{\text{Judge}}$ evaluates y_1 and y_2 by reasoning step-by-step through intermediate scores for:

- **Relevance** s_{Rel}: Matching recommendations to the user's preferences.
- **Diversity** s_{Div}: Ensuring variety across categories, brands, or other attributes.
- **Explainability** s_{Exp}: Clearly describing key features and reasoning for recommendations.

These intermediate scores serve as reasoning steps (akin to Chain-of-Thought [26]) to help the $\text{LLM}_{\text{Judge}}$ determine the preference pair in a single step:

$$(y_{\text{chosen}}, y_{\text{rejected}}) \leftarrow \text{LLM}_{\text{Judge}}(\mathcal{I}', y_1, y_2, P_{\text{Judge}}), \tag{5}$$

where the scoring dimensions $s_{\text{Rel}}, s_{\text{Div}}, s_{\text{Exp}}$ are generated in intermediate steps to guide the LLM's decision.

Below is an example instruction prompt for LLM as a judge for a music recommendation task [3]. Prompt templates for other datasets are provided in Appendix **??**.

Prompt Template for LLM as Judge

You are an expert evaluator tasked with assessing two sets of music recommendations based on three criteria: relevance, diversity, and explainability. You will compare the options and determine which set better meets the criteria, choosing one as the preferred set and the other as the rejected set. Follow the instructions below:

Listening History: {history}
Available Artists/Bands: {candidates}
Option A: {option a}
Option B: {option b}
Next Preferred Artist/Band: {label}

Criteria for Evaluation:
- Relevance: How closely do the recommended artists/bands match the "Next Preferred Artist/Band" provided?
- Diversity: Are the recommended artists/bands diverse enough to cover different styles, genres, or moods while still aligning with user preferences?
- Explainability: Do the identified key preferences accurately capture the user's listening history and provide clear reasons for the recommendations?

Task:
1. Review the listening history and the two options.
2. Evaluate each option based on the three criteria.
3. Provide a score (0-5) for each criterion and a brief explanation of your reasoning.
4. Select the option that best fulfills these criteria as the chosen option and the other as the rejected option.

Output Format:
"'
Option A Evaluation:
- Relevance Score: [0-5] - [Explanation]
- Diversity Score: [0-5] - [Explanation]
- Explainability Score: [0-5] - [Explanation]

Option B Evaluation:
- Relevance Score: [0-5] - [Explanation]
- Diversity Score: [0-5] - [Explanation]
- Explainability Score: [0-5] - [Explanation]

Decision:
- Chosen Option: [A or B]
- Reasoning: [Provide a summary explaining why this option was chosen over the other, highlighting the strengths and weaknesses observed based on the criteria.]
"'

The $\text{LLM}_{\text{Judge}}$ automates evaluation with consistency and scalability, mitigating reliance on human intervention. By integrating multi-dimensional assessments into a single reasoning step, it promotes balanced recommendations while providing high-quality supervision signals. These preference pairs facilitate iterative optimization using DPO, progressively aligning recommender outputs with user preferences.

4.3 Direct Preference Optimization (DPO)

The DPO training process consists of three steps. First, preference pairs $(y_{\text{chosen}}, y_{\text{rejected}})$ are collected from the judge Eq. (5). Second, the recommender model LLM_{Rec} is fine-tuned using the DPO loss function, as defined in Eq. (2). The objective is to maximize the likelihood of the preferred outputs y_{chosen} while minimizing that of the less preferred ones y_{rejected}. Finally, the evaluation and optimization steps are repeated iteratively, enabling the recommender to progressively align its outputs with user-centric preferences over time.

The pseudocode for the entire iterative training process is presented in Algorithm 1.

5 Experiment

Datasets. We evaluate *RecLAIF* on three real-world benchmark datasets varying in size and domain: `ESCI` [18], `Beauty` [7], and `LastFM` [3]. Summary statistics for these datasets are provided in Table 1.

`ESCI`. This dataset contains challenging Amazon search queries paired with up to 40 potentially relevant results. Each result is annotated with ESCI relevance judgments—Exact, Substitute, Complement, or Irrelevant—indicating the relevance of each product to the query [18]. For simplicity, we use only the English queries and filter out those paired with fewer than 10 "Exact" relevance

Algorithm 1: Training Process for RecLAIF

Input: Initial LLM recommender model LLM_{Rec}, Judge model $\text{LLM}_{\text{Judge}}$, Input data $\mathcal{D}$, Temperature T, Nucleus threshold p, DPO scaling factor β

Output: Fine-tuned recommender model $\text{LLM}^{\star}_{\text{Rec}}$

for *iteration* $i \leftarrow 1$ ***to*** N **do**

  Step 1: Generate Candidate Outputs with LLM_{Rec};

  for *each input* $x \in \mathcal{D}$ **do**

    Sample two candidate outputs:

$$y_1 \sim \text{LLM}_{\text{Rec}}(x; T, p), \quad y_2 \sim \text{LLM}_{\text{Rec}}(x; T, p)$$

  Step 2: Generate Preference Pairs with $\text{LLM}_{\text{Judge}}$;

  for *each pair* (y_1, y_2) **do**

    Evaluate outputs using scoring criteria:

$$(y_{\text{chosen}}, y_{\text{rejected}}) = \text{LLM}_{\text{Judge}}(\mathcal{I}', y_1, y_2; P_{\text{Judge}})$$

    Store preference pairs $(y_{\text{chosen}}, y_{\text{rejected}})$;

  Step 3: Fine-Tune Recommender using DPO;

  for *each preference pair* $(y_{chosen}, y_{rejected})$ **do**

    Update recommender model LLM_{Rec} by optimizing the DPO loss in Eq. (2).

return Fine-tuned recommender model $\text{LLM}^{\star}_{\text{Rec}}$;

products. Each query is associated with 15 randomly sampled candidates, comprising 10 "Exact" products and 5 non-"Exact" products (i.e., Substitute, Complement, or Irrelevant). The task involves query-product retrieval: given a query and 15 candidates, retrieve 5 relevant products and rank them in descending order of likelihood to recommend. The dataset is split into 60% training, 20% validation, and 20% testing.

`Beauty` (Amazon Reviews). This dataset, collected in 2023, includes user reviews, item metadata, and user-item or bought-together graphs [7]. We focus on the "All Beauty" domain and evaluate a sequential recommendation task: predict the next item of interest (i_{N+1}) given a user's historical interaction sequence ($i_1, ..., i_N$), where $N = 10$ is the sequence length. Each item i is associated with metadata in the form of a descriptive sentence. The dataset includes 10 randomly sampled candidates for each user, and the items in the interaction sequence are ordered chronologically. We follow the official dataset splits available on Hugging Face [1].

`LastFM`. This dataset contains users' listening histories from the Last.fm online music service, with artist names as associated features [3]. We evaluate a sequential recommendation task: predict the next musician or band a user will listen to (i_{N+1}) based on their historical listening sequence ($i_1, ..., i_N$), where $N = 20$. For each user, 20 candidates are randomly sampled. We preprocess the data and create splits according to the methodology described in Chen et al. [4].

Baselines. For `ESCI`, which is a retrieval task, we consider two types of baselines: (1) Retrieval-based methods, including the sparse retriever BM25 and the dense retriever BLAIR [7]; (2) LLM-based methods, which include open-source 7B models such as Mistral [8], Llama [23], and Qwen [1], as well as closed-source models such as Claude 3 Sonnet. Claude is used in a black-box setting with zero-shot prompting, without any fine-tuning or in-context examples. The prompts and inputs used to query Claude are detailed in Section 4.1, allowing anyone with access to Claude Sonnet via Amazon Bedrock to reproduce our results. For `Beauty` and `LastFM`, which are sequential recommendation tasks, we consider (1) traditional RecSys methods and (2) LLM-based RecSys methods as baselines.

[1] https://huggingface.co/datasets/McAuley-Lab/Amazon-Reviews-2023

Table 1: Summary statistics of datasets

Dataset	Task	#User	#Item	# Interactions
`ESCI`	Retrieval	4,863	129,023	143,982
`Beauty`	Recommendation	253	341	2535
`LastFM`	Recommendation	1,220	4,606	73,510

Implementation Details. For LLM as a recommender, we use the Mistral-7B-Instruct (henceforth referred to as Mistral) as the base model for performing the retrieval and recommendation tasks. For LLM as a judge, we use Claude 3 Sonnet (henceforth referred to as Claude) to provide AI feedback and evaluate the recommendations.

To enhance the performance of Mistral in recommendation tasks, we adopt a supervised fine-tuning (SFT) approach on recommendation-specific data. The fine-tuning process involves the following steps: (1) Generating Ground Truth: We query Claude with the same prompt used for Mistral when acting as a recommender (as detailed in Section 4.1). Claude's responses are treated as the "ground truth" for the task. (2) Fine-Tuning Mistral: Using an identical prompt and Claude's responses as training data, we fine-tune Mistral to align its outputs more closely with the high-quality recommendations generated by Claude.

This approach enables Mistral to learn from Claude's reasoning and recommendations, improving its ability to perform recommendation tasks effectively.

Evaluation Metrics. We evaluate the performance of *RecLAIF* across three dimensions: relevance, diversity, and explainability.

- **Relevance.** Ground-truth labels are used to calculate standard relevance metrics, including precision@K, NDCG@K, and Hits@K, which measure how well the recommended items align with user preferences.
- **Diversity.** To assess diversity, we embed the text attributes of items using Sentence Transformer and compute pairwise cosine distances between embeddings. A higher average cosine distance indicates greater diversity in the recommendations.
- **Explainability.** We employ Claude 3 Sonnet as the evaluator. Claude assigns a score ranging from 1 to 100 based on the clarity of the recommendations' explanations, along with how well they align with the user's needs.
- **Validity Ratio.** Following [4, 11], we include the Validity Ratio to assess LM-based methods' adherence to instructions and their ability to generate appropriate, coherent responses.

Using Claude to evaluate explainability is motivated by the complexity of defining and quantifying explainability through traditional methods. A recent study [33] verified that utilizing LLMs as

Table 2: Comparison of retrieval-based and LLM-based methods on the ESCI dataset across three metrics: Relevance (Prec@5, NDCG@5), Diversity (Pairwise Distance), and Explainability (Expl. Score). The best performance for each metric is highlighted in bold. '-' indicates that the evaluation is not applicable.

Method	Relevance		Diversity	Expl.
	Prec@5	NDCG@5		
Retrieval-Based				
BM25	0.7289	0.7335	**0.3558**	–
BLAIR	0.7423	0.7441	0.3486	–
LLM-Based				
Mistral-7B-Instruct	0.7520	0.7685	0.3481	86.08
Llama-3-8B-Instruct	0.7555	0.7695	0.3454	86.48
Qwen2.5-7B-Instruct	0.7518	0.7700	0.3467	86.24
Claude 3 Sonnet	0.7951	0.8059	0.3323	88.40
RecLAIF	**0.8195**	**0.8279**	0.3164	**88.90**

evaluators can provide an accurate, reproducible, and cost-effective solution for assessing recommendation explanation texts.

5.1 Retrieval Performance on ESCI

Table 2 presents the performance of retrieval-based and LLM-based methods on the ESCI dataset, evaluated across relevance, diversity, and explainability. Retrieval-based methods like BM25 (NDCG@5 = 0.7335) and BLAIR (0.7441) achieve competitive diversity but underperform in relevance and do not provide explainability.

LLM-based methods significantly improve relevance over traditional methods, with Claude 3 Sonnet (NDCG@5 = 0.8059) achieving the strongest baseline performance. However, these methods generally sacrifice diversity, with pairwise distance scores slightly lower than retrieval-based methods.

Our proposed *RecLAIF* achieves the best performance in both relevance (NDCG@5 = 0.8279) and explainability, demonstrating its ability to retrieve accurate and interpretable results. While *RecLAIF* does not achieve the highest diversity score, it effectively balances precision and item diversity, addressing the trade-off inherent in retrieval tasks.

5.2 Recommendation Performance

The results are presented in Table 3, which compares traditional recommendation methods, LLM-based methods, and our proposed *RecLAIF* framework on the Beauty and LastFM datasets across four metrics: Validity Ratio, Relevance (Hit@1, NDCG@3), Diversity, and Explainability.

For the Beauty dataset, traditional methods such as SASRec achieve perfect ValidRatio but perform poorly on relevance and diversity. LLM-based methods, such as Mistral-7B-Instruct and Claude, improve relevance significantly over traditional models, with Claude achieving strong NDCG@3. However, these methods trade off diversity, with scores lower than desired. *RecLAIF* outperforms all baselines across relevance and diversity, achieving the highest Hit@1 and NDCG@3 while maintaining a balanced diversity score. Notably, *RecLAIF* achieves competitive explainability, highlighting its ability to generate interpretable recommendations.

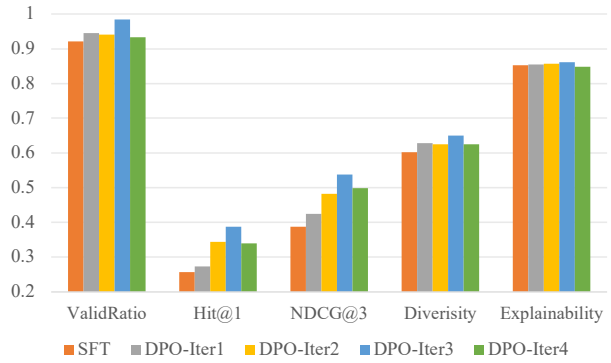

Figure 3: Performance comparison of the recommender model trained with SFT and successive DPO iterations on the Beauty dataset across multiple metrics.

For the LastFM dataset, a similar trend is observed. Traditional models exhibit high ValidRatio but fail to perform competitively in relevance metrics. LLM-based methods again improve relevance, with Claude achieving strong results. However, *RecLAIF* achieves the best overall performance, with the highest Hit@1 and NDCG@3, while also achieving a well-balanced diversity score.

These results demonstrate that *RecLAIF* effectively addresses the limitations of both traditional and LLM-based methods, delivering superior relevance, diversity, and explainability while maintaining strong performance across datasets.

5.3 Ablation Study

To evaluate the effectiveness of DPO in fine-tuning the recommender, we conduct an ablation study analyzing performance across multiple DPO iterations. Specifically, we compare the baseline SFT model with the recommender fine-tuned iteratively using DPO (denoted as DPO-Iter1 through DPO-Iter4). The results are reported on the Beauty dataset using five metrics: ValidRatio, Hit@1, NDCG@3, Diversity, and Explainability.

The results, summarized in Figure 3, demonstrate that DPO progressively enhances the recommender's performance. We observed significant improvements in relevance (Hit@1 and NDCG@3), diversity, and explainability compared to the baseline SFT. Notably, performance gains saturate around DPO-Iter3, indicating diminishing returns with further iterations, which might be due to overfitting. This trend highlights the efficacy of DPO in leveraging AI feedback to optimize the recommender iteratively, achieving substantial improvements while maintaining stability.

6 Conclusion

We introduced *RecLAIF*, a dual-LLM framework that integrates Reinforcement Learning from AI Feedback (RLAIF) to enhance recommendation systems in terms of relevance, diversity, and explainability. By leveraging LLMs as both recommenders and judges,

Table 3: Comparison of traditional recommendation methods, LLM-based methods, and our proposed *RecLAIF* framework on the `Beauty` and `LastFM` datasets. Metrics include Validity Ratio (ValidRatio), Relevance (Hit@1, NDCG@3), Diversity (Pairwise Distance), and Explainability (Expl. Score). The best performance for each metric is highlighted in bold. '–' indicates that the evaluation is not applicable.

		`Beauty`					`LastFM`				
		ValidRatio	Hit@1	NDCG@3	Diversity	Expl.	ValidRatio	Hit@1	NDCG@3	Diversity	Expl.
Traditional	SASRec	**1.0000**	0.0474	0.1255	0.3120	–	**1.0000**	0.3581	0.4792	0.5803	–
LLM-Based	Mistral-7B-Instruct	0.6047	0.1304	0.2370	0.4051	**87.98**	0.8878	0.2982	0.3991	0.6968	87.88
	Claude	0.9921	0.2806	0.4725	0.6439	87.86	0.9830	0.4080	0.5399	**0.7676**	88.58
Ours	*RecLAIF*	0.9842	**0.3874**	**0.5375**	**0.6493**	86.14	**1.0000**	**0.5126**	**0.6517**	0.7572	**88.79**

RecLAIF provides a scalable alternative to human feedback while addressing limitations of traditional RecSys.

Experiments on three real-world datasets demonstrate that *RecLAIF* significantly outperforms baseline methods in relevance, diversity, and explainability, showcasing the potential of RLAIF as a cost-effective and adaptive solution for modern RecSys.

Limitation and Future work. Currently, the judge is a general-purpose, static LLM. Future work could explore fine-tuning or self-teaching the LLM judge on the evaluation task, enhancing its effectiveness and domain adaptability.

References

[1] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609* (2023).

[2] Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. 2022. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073* (2022).

[3] Ivan Cantador, Peter Brusilovsky, and Tsvi Kuflik. 2011. Second workshop on information heterogeneity and fusion in recommender systems (HetRec2011). In *Proceedings of the Fifth ACM Conference on Recommender Systems* (Chicago, Illinois, USA) *(RecSys '11)*. Association for Computing Machinery, New York, NY, USA, 387–388. doi:10.1145/2043932.2044016

[4] Yuxin Chen, Junfei Tan, An Zhang, Zhengyi Yang, Leheng Sheng, Enzhi Zhang, Xiang Wang, and Tat-Seng Chua. 2024. On Softmax Direct Preference Optimization for Recommendation. *arXiv preprint arXiv:2406.09215* (2024).

[5] Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. *Advances in neural information processing systems* 30 (2017).

[6] Zhankui He, Zhouhang Xie, Rahul Jha, Harald Steck, Dawen Liang, Yesu Feng, Bodhisattwa Prasad Majumder, Nathan Kallus, and Julian McAuley. 2023. Large language models as zero-shot conversational recommenders. In *Proceedings of the 32nd ACM international conference on information and knowledge management*. 720–730.

[7] Yupeng Hou, Jiacheng Li, Zhankui He, An Yan, Xiusi Chen, and Julian McAuley. 2024. Bridging Language and Items for Retrieval and Recommendation. *arXiv preprint arXiv:2403.03952* (2024).

[8] Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7B. *arXiv preprint arXiv:2310.06825* (2023).

[9] Harrison Lee, Samrat Phatale, Hassan Mansoor, Kellie Ren Lu, Thomas Mesnard, Johan Ferret, Colton Bishop, Ethan Hall, Victor Carbune, and Abhinav Rastogi. 2023. Rlaif: Scaling reinforcement learning from human feedback with ai feedback. (2023).

[10] Yuxuan Lei, Jianxun Lian, Jing Yao, Xu Huang, Defu Lian, and Xing Xie. 2023. Recexplainer: Aligning large language models for recommendation model interpretability. *arXiv preprint arXiv:2311.10947* (2023).

[11] Jiayi Liao, Sihang Li, Zhengyi Yang, Jiancan Wu, Yancheng Yuan, Xiang Wang, and Xiangnan He. 2023. Llara: Aligning large language models with sequential recommenders. *arXiv preprint arXiv:2312.02445* (2023).

[12] Jianghao Lin, Xinyi Dai, Yunjia Xi, Weiwen Liu, Bo Chen, Hao Zhang, Yong Liu, Chuhan Wu, Xiangyang Li, Chenxu Zhu, et al. 2023. How can recommender systems benefit from large language models: A survey. *arXiv preprint arXiv:2306.05817* (2023).

[13] Jianghao Lin, Xinyi Dai, Yunjia Xi, Weiwen Liu, Bo Chen, Hao Zhang, Yong Liu, Chuhan Wu, Xiangyang Li, Chenxu Zhu, Huifeng Guo, Yong Yu, Ruiming Tang, and Weinan Zhang. 2024. How Can Recommender Systems Benefit from Large Language Models: A Survey. *ACM Trans. Inf. Syst.* (jul 2024). doi:10.1145/3678004

[14] Yucong Luo, Mingyue Cheng, Hao Zhang, Junyu Lu, Qi Liu, and Enhong Chen. 2023. Unlocking the potential of large language models for explainable recommendations. *arXiv preprint arXiv:2312.15661* (2023).

[15] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems* 35 (2022), 27730–27744.

[16] Zhaopeng Qiu, Xian Wu, Jingyue Gao, and Wei Fan. 2021. U-BERT: Pre-training user representations for improved recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 4320–4327.

[17] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2024. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems* 36 (2024).

[18] Chandan K. Reddy, Lluís Màrquez, Fran Valero, Nikhil Rao, Hugo Zaragoza, Sambaran Bandyopadhyay, Arnab Biswas, Anlu Xing, and Karthik Subbian. 2022. Shopping Queries Dataset: A Large-Scale ESCI Benchmark for Improving Product Search. (2022). arXiv:2206.06588

[19] Xubin Ren, Wei Wei, Lianghao Xia, Lixin Su, Suqi Cheng, Junfeng Wang, Dawei Yin, and Chao Huang. 2024. Representation learning with large language models for recommendation. In *Proceedings of the ACM on Web Conference 2024*. 3464–3475.

[20] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347* (2017).

[21] Amit Sharma, Hua Li, Xue Li, and Jian Jiao. 2024. Optimizing Novelty of Top-k Recommendations using Large Language Models and Reinforcement Learning. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 5669–5679.

[22] Guangsi Shi, Xiaofeng Deng, Linhao Luo, Lijuan Xia, Lei Bao, Bei Ye, Fei Du, Shirui Pan, and Yuxiao Li. 2024. Llm-powered explanations: Unraveling recommendations through subgraph reasoning. *arXiv preprint arXiv:2406.15859* (2024).

[23] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288* (2023).

[24] Tianlu Wang, Ilia Kulikov, Olga Golovneva, Ping Yu, Weizhe Yuan, Jane Dwivedi-Yu, Richard Yuanzhe Pang, Maryam Fazel-Zarandi, Jason Weston, and Xian Li. 2024. Self-taught evaluators. *arXiv preprint arXiv:2408.02666* (2024).

[25] Ziyan Wang, Yingpeng Du, Zhu Sun, Haoyan Chua, Kaidong Feng, Wenya Wang, and Jie Zhang. 2024. Re2LLM: Reflective Reinforcement Large Language Model for Session-based Recommendation. *arXiv preprint arXiv:2403.16427* (2024).

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

[27] Wei Wei, Xubin Ren, Jiabin Tang, Qinyong Wang, Lixin Su, Suqi Cheng, Junfeng Wang, Dawei Yin, and Chao Huang. 2024. Llmrec: Large language models with graph augmentation for recommendation. In *Proceedings of the 17th ACM*

International Conference on Web Search and Data Mining. 806–815.

[28] Laura Weidinger, John Mellor, Maribeth Rauh, Conor Griffin, Jonathan Uesato, Po-Sen Huang, Myra Cheng, Mia Glaese, Borja Balle, Atoosa Kasirzadeh, et al. 2021. Ethical and social risks of harm from language models. *arXiv preprint arXiv:2112.04359* (2021).

[29] Chuhan Wu, Fangzhao Wu, Tao Qi, and Yongfeng Huang. 2021. Empowering news recommendation with pre-trained language models. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*. 1652–1656.

[30] Yunjia Xi, Weiwen Liu, Jianghao Lin, Jieming Zhu, Bo Chen, Ruiming Tang, Weinan Zhang, Rui Zhang, and Yong Yu. 2023. Towards open-world recommendation with knowledge augmentation from large language models. *arXiv preprint arXiv:2306.10933* (2023).

[31] Shitao Xiao, Zheng Liu, Yingxia Shao, Tao Di, Bhuvan Middha, Fangzhao Wu, and Xing Xie. 2022. Training large-scale news recommenders with pretrained language models in the loop. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 4215–4225.

[32] Shaowei Yao, Jiwei Tan, Xi Chen, Juhao Zhang, Xiaoyi Zeng, and Keping Yang. 2022. ReprBERT: distilling BERT to an efficient representation-based relevance model for e-commerce. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 4363–4371.

[33] Xiaoyu Zhang, Yishan Li, Jiayin Wang, Bowen Sun, Weizhi Ma, Peijie Sun, and Min Zhang. 2024. Large Language Models as Evaluators for Recommendation Explanations. arXiv:2406.03248 [cs.IR] https://arxiv.org/abs/2406.03248