Retrieval Augmented Conversational Recommendation with Reinforcement Learning

Zhenrui Yue[1], **Honglei Zhuang**[2], **Zhen Qin**[2], **Zhankui He**[2], **Huimin Zeng**[1], **Julian McAuley**[3], **Dong Wang**[1]
[1]University of Illinois Urbana-Champaign, [2]Google DeepMind, [3]UC San Diego
{zhenrui3, dwang24}@illinois.edu

Abstract

Large language models (LLMs) exhibit enhanced capabilities in language understanding and generation. By utilizing their embedded knowledge, LLMs are increasingly used as conversational recommender systems (CRS), achieving improved recommendation performance across diverse scenarios. However, existing LLM-based methods rely on pretrained knowledge without external retrieval mechanisms for novel items. Additionally, the lack of a unified corpus poses challenges for integrating retrieval augmentation into CRS. Motivated by these challenges, we present RAR, a novel two-stage retrieval augmented conversational recommendation framework that aligns retrieval and generation to enhance both performance and factuality. To support this framework and provide a unified corpus, we construct a large-scale movie corpus, comprising over 300k movies with rich metadata, such as titles, casts and plot summaries. Leveraging this data, our primary contribution is RAR, the first framework to departs from standard two-stage CRS by dynamically bridging retrieval and generation. First, a retriever model generates candidate items based on user history; in the subsequent stage, an LLM refines the recommendations by incorporating conversational context with retrieved results. In addition, we introduce a novel reinforcement learning (RL) method that leverages LLM feedback to iteratively update the retriever. By creating a collaborative feedback loop that reinforces sampled candidate sets with higher ranking metrics, RAR effectively mitigates the misalignment between the retrieval and generation stages. Furthermore, grounding the LLM in factual metadata allows our RL-driven approach to capture subtle user intentions and generate context-aware recommendations with reduced hallucinations. We validate our approach through extensive experiments on multiple CRS benchmarks, where RAR consistently outperforms state-of-the-art baseline methods.

1 Introduction

Conversational recommender systems (CRS) have emerged as a promising paradigm for providing personalized recommendations through natural language interactions (Zhang et al., 2018; Li et al., 2018; Kang et al., 2019; Hayati et al., 2020; Gao et al., 2023; He et al., 2023). Recent advancements in large language models (LLM) (Achiam et al., 2023; Dubey et al., 2024; Comanici et al., 2025) have further enabled CRS methods to utilize their extensive knowledge and diverse generation capabilities, demonstrating improvements across different recommendation scenarios (Feng et al., 2023; Yang & Chen, 2024; Li et al., 2024; Hui et al., 2026). For example, He et al. (2023) adopt LLMs to generate movie candidates and achieve superior performance compared to traditional CRS methods.

Nevertheless, language models often lack the essential knowledge to recommend the most suitable items (e.g., being unaware of relevant options) (He et al., 2023; Yang & Chen, 2024; Wu et al., 2024). As a solution, traditional CRS methods exploit search modules or knowledge graphs to provide additional contextual information (Zhang et al., 2018; Chen et al., 2019; Zhou et al., 2020). Similarly, LLM-based CRS methods leverage knowledge graph retrieval

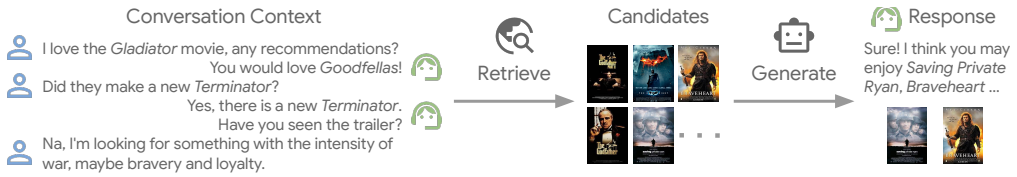

Figure 1: Our retrieval-augmented conversational recommendation framework, where a retriever gathers candidate items and the LLM generates the response conditioned on them.

or training on a broad range of items to improve recommendation relevance (Li et al., 2024; Jeon et al., 2025; Zare & Alamdari, 2025). A notable example is ReFICR, where LLMs are trained on extensive curated data to perform sub-tasks including indexing, retrieval and generation(Yang & Chen, 2024). In Figure 1, we illustrate this general approach of the two-stage retrieval augmented conversational recommendation.

While LLMs can accommodate novel items through additional training, scaling this process to extensive items and conversations is often infeasible, particularly when managing large volumes of items and noisy conversations (Jannach et al., 2021; Zhu et al., 2025; Surana et al., 2025). On the other hand, retrieving from knowledge graph requires significant efforts in data preprocessing, modeling and graph indexing. Graph retrieval can also be more computationally intensive when indexing or traversing through large-scale graphs (Li et al., 2024; Peng et al., 2024). Although embedding-based retrieval is a straightforward alternative (Reimers & Gurevych, 2019; Wang et al., 2022a; Chen et al., 2024), its effectiveness is limited by the lack of a unified corpus that provides comprehensive attribute-level metadata (e.g., plot, cast) corpus. As such, embedding-based retrieval remains largely under-explored in conversational recommendation. Furthermore, LLM-based CRS often exhibits retrieval–generation misalignment: when the retriever returns sub-optimal candidates (e.g., low-relevance items), LLM could amplify such deficiencies and cause deteriorated accuracy on cold-start or less-popular items (He et al., 2023; Kemper et al., 2024).

In this paper, we investigate how embedding-based retrieval augmentation can be used to improve two-stage conversational recommendation. To address the limitations of existing, small-scale movie corpora (e.g., REDIAL with ∼7k titles), we curate a comprehensive evaluation benchmark of over 300k movies, systematically enriched with metadata (e.g., titles, casts, and plot summaries). By establishing this robust, large-scale foundation, our primary contribution is RAR, a novel LLM-based retrieval augmented conversational recommendation framework. In the first stage, we employ a retriever model to select candidate items based on the conversational history. Subsequently, an LLM refines the recommendations by incorporating conversational context with retrieved candidates, enabling personalized and fine-grained recommendation results. To enhance retrieval-generation alignment, we further optimize the retriever by leveraging LLM feedback for reinforcement learning (RL). This feedback enables RAR to iteratively update the retriever through online, on-policy preference optimization. As a result, RAR can be adapted for any black-box LLMs and deliver high-quality, context-aware recommendations. We demonstrate the efficacy of RAR with extensive experiments, where RAR consistently outperforms baselines on multiple benchmark datasets. We summarize our contributions in the following[1]:

1. We present the first large-scale study of embedding-based retrieval augmentation for conversational recommendation. To support this scale, we synthesize a highly-enriched, structured baseline of over 300k films with comprehensive metadata.
2. We propose RAR, an LLM-based CRS framework with two-stage retrieval augmentation. By utilizing LLM feedback as reward signals, we design an online, on-policy reinforcement learning framework that enhances retrieval–generation alignment.
3. We demonstrate the effectiveness our approach by utilizing our curated retrieval corpus, where the RL post-trained RAR consistently outperforms state-of-the-art

[1]Our data and code can be accessed at https://github.com/Yueeeeeee/RAR.

baseline methods, achieving considerable improvements in recommendation performance on established benchmark datasets.

2 Related Work

2.1 Retrieval Augmented Generation

Retrieval augmented generation (RAG) enhances language modeling by integrating external knowledge into context (Lewis et al., 2020; Guu et al., 2020; Karpukhin et al., 2020). Beyond naïve RAG, performance is improved by refining the retriever (e.g., via joint optimization (Lin et al., 2024)) (Trivedi et al., 2023; Jiang et al., 2023; Shi et al., 2024; Sarthi et al., 2024), upgrading document encoding (Izacard & Grave, 2021; Borgeaud et al., 2022; Izacard et al., 2023), and selectively filtering retrieved knowledge to minimize irrelevant context (Yoran et al., 2024; Yan et al., 2024; Yue et al., 2024b; Jin et al., 2025). While concurrently applied to sequential recommendation (Wu et al., 2024; Kemper et al., 2024), RAG's potential for conversational recommendation remains largely under-explored. To bridge this gap, we build a unified movie metadata corpus and examine retrieval-augmented LLMs, aiming to enhance CRS performance via effective knowledge retrieval.

2.2 Conversational Recommendation

Conversational recommender systems (CRS) capture user preferences via multi-turn interactions to improve recommendations. Early methods relied on complex, multi-module systems for dialogue and recommendation (Zhang et al., 2018; Li et al., 2018; Kang et al., 2019; Hayati et al., 2020; Lei et al., 2020). Later, language modeling and knowledge integration enabled better contextualization of user intent (Chen et al., 2019; Zhou et al., 2020; Lu et al., 2021; Wang et al., 2022b), such as inferring preferences from incomplete knowledge graphs (Zhang et al., 2023). Recently, large language models (LLMs) (Achiam et al., 2023; Reid et al., 2024; Dubey et al., 2024) have transformed CRS through advanced dialogue capabilities and precise preference modeling (Gao et al., 2023; He et al., 2023; Li et al., 2024; He et al., 2024; Yang & Chen, 2024; Hui et al., 2026), including using LLMs for user simulation (Zhu et al., 2025). Despite these advances, integrating LLMs into CRS and improving the retrieval–generation alignment remain under-investigated.

2.3 Reinforcement Learning

Reinforcement learning (RL) maximizes cumulative rewards through environmental interaction (Sutton et al., 1998). RL has been adapted to align LLMs via human feedback (RLHF) (Ouyang et al., 2022), typically utilizing policy gradient variants (Sutton et al., 1999). Methods like A2C and PPO (Mnih et al., 2016; Schulman et al., 2017) leverage learned baselines and clipped surrogate objectives for stable training. Alternatively, Direct Preference Optimization (DPO) (Rafailov et al., 2023) efficiently optimizes offline pairwise preferences, though online RL methods consistently achieve superior performance (Xu et al., 2024). Recent online advances like GRPO evaluate candidates against a group-based baseline, significantly reducing memory overhead without sacrificing stability (Shao et al., 2024; Hu, 2025; Yue et al., 2025; 2026; Hübotter et al., 2026). In this work, we propose a novel online RL-driven approach for two-stage conversational recommender systems (CRS) and empirically compare different RL algorithms to identify the optimal strategy for recommendation.

3 Methodology

3.1 Setup

Our conversational recommendation framework is based on a two-stage setting with both retrieval and generation. Consider a conversation $\mathcal{C} = (r_t, s_t, I_t)_{t=1}^{T}$ consisting of T turns, r_t, s_t and I_t refer to the role (i.e., *seeker* or *recommender*), sequence (conversation turn) and mentioned items. Each element in I_t should be included in the corpus $\mathcal{I}$, namely

$\forall i \in I_t,\ i \in \mathcal{I}$. Typically, the *seeker* and *recommender* take turns to converse in $\mathcal{C}$ until the seeker finds desired item(s). Following previous research (Li et al., 2018; Chen et al., 2019; Wang et al., 2022b; He et al., 2023), the conversational recommender aims to generate a ranked list $\hat{I}_t$ for the t-th turn when r_t is the *recommender*, such that the generated $\hat{I}_t$ best matches the ground truth items I_t. Specifically for our two-stage framework:

- *Retrieval*: In the first stage, the retriever model f_{ret} uses the previous items as query to select an initial candidate set C_t, i.e., $C_t = f_{\text{ret}}(\{I_\tau\}_{\tau=1}^{t-1})$.
- *Generation*: Then, we leverage a LLM f_{llm} to generate $\hat{I}_t$ upon user conversation history and retrieved items. Formally, this is expressed as $\hat{I}_t = f_{\text{llm}}(\{s_k\}_{k=1}^{t-1}, C_t)$, enabling an in-depth analysis of user preferences and potential interests.

To fully leverage black-box LLMs without excessive training costs, we refrain from directly training f_{llm} and instead focus on optimizing the retriever model f_{ret}. This design choice offers two key advantages: (1) RAR can be combined with any LLM choice, regardless of whether the LLM is open- or closed-source; (2) by enhancing f_{ret}, we augment f_{llm} through the incorporation of up-to-date external knowledge (e.g., novel items). Our framework optimizes f_{ret} (parameterized by θ) to maximize the expected reward. Since the retriever and generator are not end-to-end differentiable, generator errors cannot be directly back-propagated. We therefore train f_{ret} via reinforcement learning, maximizing reward r over samples from the retriever across conversations ($\mathcal{C}$) and timesteps (t) in dataset $\mathcal{X}$:

$$\max_{\theta} \mathbb{E}_{\mathcal{C}\sim\mathcal{X}, t\sim\{1,\ldots,T(\mathcal{C})\}, C_t\sim f_{\text{ret}}(\{I_\tau\}_{\tau=1}^{t-1})}[r(f_{\text{llm}}(\{s_k\}_{k=1}^{t-1}, C_t), I_t)], \tag{1}$$

where r is a ranking-based reward for the sampled candidate set C_t calculated based on the output of f_{llm} (e.g., NDCG scores). That is, the post-training process updates θ by maximizing the ranking reward, thereby aligning the retrieval and ranking stages in RAR.

3.2 Corpus Construction

Existing conversational recommendation datasets either rely on large-scale knowledge graphs or on small, domain-specific corpus (Li et al., 2018; Hayati et al., 2020; He et al., 2023). Therefore, embedding-based retrieval is often impractical with existing data. To address this limitation, we collect and curate a unified text corpus for movie recommendation across datasets. Our initial corpus is built from all available entries across multiple sources, including IMDb genre, IMDb media and Inspired (Hayati et al., 2020; Raju; BrightData). Next, we augment the corpus with inferred entries from MovieLens, Redial and Reddit (Harper & Konstan, 2015; Li et al., 2018; He et al., 2023). Then, we aggregate duplicate film entries by cross-referencing and selecting the most reliable metadata. Additionally, we collect missing metadata from online resources for incomplete entries, we also map items in the corpus to the corresponding entries in the datasets (i.e., MovieLens, Inspired, Redial and Reddit). That is, we perform entity recognition to identify each movie mentioned in the conversation and map it to its corresponding entry in the corpus. Finally, we remove entries that were either incomplete or could not be collected, we report detailed information on corpus collection and preprocessing in Section A. In summary, we obtain 337,731 entries with comprehensive metadata, focusing primarily on English-language films (see example in Table 7). The corpus spans from as early as 1888 to upcoming releases in 2029, accounting for approximately half of all movies listed on IMDb as of 2025. Leveraging the collected corpus and constructed correspondence, we pretrain a retriever model on MovieLens by incorporating recommendation data along with negative examples sampled from the corpus (Harper & Konstan, 2015; Yue et al., 2024a). Next, the retriever is optimized with sequential patterns derived from the conversation data, we introduce the details in the following.

3.3 The proposed RAR

Retriever. For RAR, we adopt the linear recurrent units for sequential recommendation (LRURec) as f_{ret}, with a comparison of different models in Section 4. LRURec utilizes state space modeling (SSM) to efficiently train and infer on sequential input:

$$h_t = Ah_{t-1} + Be_t, \quad o_t = Ch_t + Ie_t, \tag{2}$$

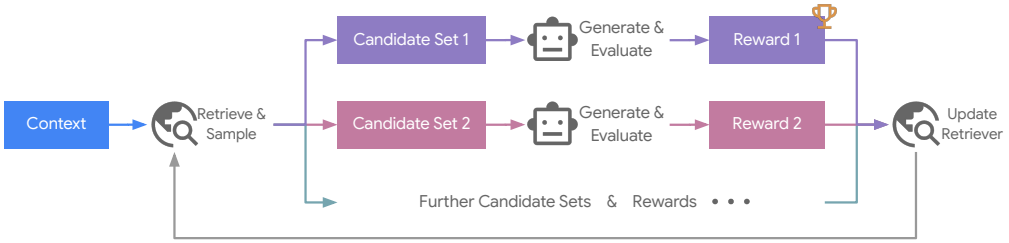

Figure 2: The proposed online, on-policy preference optimization in RAR iteratively refines the retriever model by sampling candidate sets, collecting LLM feedback, and then updating the retriever through reinforcement learning.

with A, B, C being matrices of shape $\mathbb{R}^{H\times H}$, and I is the identity matrix. h, e, o denote the hidden state, item embedding and output hidden state. By leveraging LRURec's linearity via parallel scan, we reduce time complexity to $\mathcal{O}(\log(t))$, significantly enhancing training and inference efficiency. To encode the corpus, we leverage Qwen 3 to build the embedding table (Zhang et al., 2025). The output state o_t is used to compute similarity scores for all corpus entries, and the top-k items are retrieved and stored as C_t for the generation stage.

Generator. In the generation stage, our primary objective is to produce refined recommendations $\hat{I}_t$ by integrating the conversation context $\{s_k\}_{k=1}^{t-1}$ and the retrieved items C_t. To achieve this, we employ a black-box LLM f_{llm} and construct an input prompt that combines clear instructions, detailed item metadata, and the conversation history. The LLM is instructed to generate k recommendations based on this comprehensive context. Further details of our prompt design are provided in Section A. For each conversation, we divide it into sub-conversations by splitting at turns where the role r_t is designated as *recommender*. We also exclude ground truth items that have already appeared in earlier conversation turns to avoid shortcut learning (He et al., 2023; Geirhos et al., 2020). During inference, we retrieve relevant items from the corpus to construct the prompt. The generator f_{llm} then produces recommendations, which are post-processed to yield the final ranked list $\hat{I}_t$. Unlike prior works (He et al., 2023; Kemper et al., 2024), we pair a simple retriever with a black-box LLM to inject external, up-to-date knowledge into the generation stage, enabling the LLM to access novel items for fine-grained, context-aware recommendations.

Retriever Preference Optimization. While RAR provides a retrieval augmented generation framework for conversational recommendation, these two stages are not jointly trained for optimal alignment. Yet overlooking the dynamics between both stages can result in a sub-optimal solution (Ma et al., 2020; Higley et al., 2022; Lin et al., 2024). We thus propose an online, on-policy preference optimization method that leverages real-time feedback from the LLM generator f_{llm}. This allows RAR to iteratively sample candidate sets from the retriever and utilize LLM feedback to optimize current policy (see Figure 2). In our setup, we consider a frozen generator f_{llm} and focus on post-training the retriever (parametrized by θ). This approach enables efficient optimization of the smaller f_{ret} to achieve improved retrieval. Specifically for candidate set $C_t = \{c_i\}_{i=1}^{k}$ sampled from π_θ, we define the likelihood of C_t given history items $\{I_\tau\}_{\tau=1}^{t-1}$ using the Plackett-Luce model (Plackett, 1975):

$$P_\theta(C_t|\{I_\tau\}_{\tau=1}^{t-1}) = \prod_{i=1}^{k} \frac{\exp(s_{\sigma(i)})}{\sum_{j\in\mathcal{I}\setminus\{\sigma(1),\ldots,\sigma(i-1)\}} \exp(s_j)}, \tag{3}$$

where σ is the permutation that ranks items by descending retriever score, and $s_{\sigma(i)}$ denotes the score assigned by f_{ret} to the i-th ranked item. This formulation treats the candidate set as a sequential selection process without replacement: at each step, an item is drawn from the remaining pool with probability proportional to its retriever score. For each turn, we sample multiple candidate sets and use f_{llm} to rank and compute rewards. We then apply policy gradients on the resulting advantages and log-likelihood to optimize the retriever, effectively enhancing the synergy between the retrieval and ranking stages.

Pairwise and Multi-Sample Reinforcement Learning. For pairwise RL, we adopt an online, on-policy DPO approach to maximize the probability of retrieving a favored candidate set C_w over a disfavored set C_l. Because f_{llm} acts as a black-box generator, we compute the final reward as the NDCG ranking score by identifying the rank of the ground truth items directly from the LLM's output. At each timestep, we sample two candidate sets of size k from policy π_θ; the set that yields the higher NDCG score is designated C_w. In other words, the candidate set where the label item achieves a higher rank is considered preferable. Given history $\{I_\tau\}_{\tau=1}^{t-1}$ and candidate sets, we minimize the preference loss $\mathcal{L}_{\text{dpo}}$:

$$\mathcal{L}_{\text{dpo}} = -\log\sigma(\beta\log\frac{\pi_\theta(C_w|\{I_\tau\}_{\tau=1}^{t-1})}{\pi_{\text{ref}}(C_w|\{I_\tau\}_{\tau=1}^{t-1})} - \beta\log\frac{\pi_\theta(C_l|\{I_\tau\}_{\tau=1}^{t-1})}{\pi_{\text{ref}}(C_l|\{I_\tau\}_{\tau=1}^{t-1})}) \tag{4}$$

where β controls the strength of preference learning, while the reference model π_{ref} constrains the policy updates. This objective is designed to increase the relative probability of sampling C_w over C_l while preventing excessive policy divergence. In addition to online DPO, we further introduce a multi-sample RL approach based on GRPO (Shao et al., 2024). GRPO samples a group of g candidate sets and computes the advantages by standardizing the rewards within the group (i.e., $\hat{A}_i = \frac{r_i - \texttt{mean}([r_1, r_2, \ldots, r_g])}{\texttt{std}([r_1, r_2, \ldots, r_g])}$):

$$\mathcal{L}_{\text{grpo}} = \mathbb{E}_{\mathcal{C}\sim\mathcal{X}, t\sim\{1,\ldots,T(\mathcal{C})\}, \{C_i\}_{i=1}^{g}\sim f_{\text{ret}}(\{I_\tau\}_{\tau=1}^{t-1})}\left[-\frac{1}{g}\sum_{i=1}^{g}\log\pi_\theta(C_i|\{I_\tau\}_{\tau=1}^{t-1})\hat{A}_i\right] + \beta\mathbb{D}_{KL}, \tag{5}$$

where β controls the regularization strength. The resulting algorithm, detailed in Section A.2, is lightweight, on-policy and can be seamlessly combined with further optimizations.

Training Objective. To stabilize reinforcement preference learning and maintain the retriever's standalone quality, we incorporate a supervised negative log-likelihood term, $\mathcal{L}_{\text{nll}}$. Consequently, the full preference optimization objective is:

$$\mathcal{L} = \mathcal{L}_{\text{nll}} + \mathcal{L}_{\text{rl}}, \tag{6}$$

where $\mathcal{L}_{\text{rl}}$ represents either the DPO or GRPO loss. While DPO trains the policy to favor preferred candidate sets by maximizing their relative selection probability, GRPO refines policy gradients by calculating advantages across groups of samples to amplify the likelihood of high-rewarding sets. By integrating RL loss with the negative log-likelihood term, RAR bridges the retrieval and generation stages, continually refining f_{ret} to ensure the delivery of diverse, high-quality candidate sets for conversational recommendation.

4 Experiments

4.1 Experiment Settings

Our model is evaluated on three widely-used datasets for conversational recommendation: Inspired, Redial and Reddit (Hayati et al., 2020; Li et al., 2018; He et al., 2023). We adopt multiple baselines for comparison, including both *traditional* and *LLM-based* methods. In particular, we adopt *traditional CRS*: KBRD, KGSF and UniCRS (Chen et al., 2019; Zhou et al., 2020; Wang et al., 2022b); *sequential methods*: SASRec, FMLPRec and LRURec (Kang & McAuley, 2018; Zhou et al., 2022; Yue et al., 2024a); *retrieval augmented CRS with supervised fine-tuning* (SFT) methods include Qwen3-8B (Qwen), GPT-5 mini (GPT) and Gemini 3 Flash (Gem) (Yang et al., 2025; Singh et al., 2025), where we perform SFT on the retriever model. Similarly, we experiment RAR with Qwen, GPT and Gemini as LLM generator. For our retriever model, we compare GRU4Rec, SASRec, FMLPRec and LRURec (Hidasi, 2015; Kang & McAuley, 2018; Zhou et al., 2022; Yue et al., 2024a). For all methods and datasets, the maximum history length was set to 64 (i.e. history items), and the retrieval size was set to $k = 25$ in our main experiments. The evaluation metrics are NDCG and Recall at 5 and 10. For all evaluated methods, we saved the model with the best validation NDCG@10 score. Further training and evaluation details are reported in Section A.2.

Method	Inspired				Redial				Reddit			
	N@5	R@5	N@10	R@10	N@5	R@5	N@10	R@10	N@5	R@5	N@10	R@10
KBRD	.0466	.0815	.0472	.0732	.0388	.0582	.0453	.078	.0066	.0079	.0078	.0098
KGSF	.0656	.0815	.0673	.0869	.0434	.0672	.0497	.0864	.0155	.0172	.0155	.0172
UniCRS	.0676	.0927	.0750	.1032	.0425	.0646	.0504	.0887	.0258	.0376	.0363	.0479
SASRec	.0564	.0870	.0655	.1304	.0558	.0795	.0681	.1176	.0288	.0401	.0339	.0545
FMLPRec	.0620	.0815	.0726	.1141	.0504	.0784	.0639	.1123	.0315	.0434	.0342	.0534
LRURec	.0671	.0978	.0793	.1359	.0539	.0771	.0650	.1111	.0316	.0430	.0346	.0522
SFT_{Qwen}	.0609	.0938	.0626	.0990	.0454	.0684	.0526	.0907	.0344	.0484	.0394	.0633
SFT_{Gem}	.0859	.1076	.1034	.1544	.0574	.0828	.0662	.1097	.0455	.0604	.0497	.0708
SFT_{GPT}	.0997	.1214	.1091	.1491	.0599	.0887	.0700	.1197	.0489	.0651	.0558	.0843
RAR_{Qwen}	.0693	.0980	.0773	.1241	.0491	.0704	.0569	.0947	.0368	.0536	.0444	.0770
RAR_{Gem}	.0916	.1145	.1046	.1587	.0632	.0894	**.0721**	.1169	.0502	.0661	.0531	.0799
RAR_{GPT}	**.1091**	**.1422**	**.1180**	**.1700**	**.0620**	**.0932**	.0718	**.1236**	***.0551***	***.0716***	***.0593***	***.0846***

Table 1: Main results of the online DPO-based RAR and baseline methods on conversational recommendation. For clarity, the best results for each dataset and metric are highlighted in **bold**, while the second-best results are underlined.

Retriever	N@10	R@10	N@20	R@20
GRU4Rec	.1250	.2015	.1351	.2558
BERT4Rec	.1469	.2337	.1598	.2889
SASRec	.1430	.2363	.1637	.3086
FMLPRec	.1460	.2411	.1689	.3125
LRURec	**.1483**	**.2508**	**.1700**	**.3365**

Table 2: Different retriever model performance pretrained on MovieLens.

Dataset	Method	N@5	R@5	N@10	R@10
Inspired	DPO	.0693	.0980	.0773	**.1241**
	GRPO	**.0753**	**.0997**	**.0807**	.1162
Redial	DPO	.0491	.0704	.0569	.0947
	GRPO	**.0493**	**.0706**	**.0575**	**.0954**
Reddit	DPO	.0368	.0536	.0444	**.0770**
	GRPO	**.0385**	**.0547**	**.0450**	.0765

Table 3: DPO & GRPO Comparison on Qwen.

4.2 Experiment Results and Analysis

RQ1. How does RAR perform in conversational recommendation? We first discuss the evaluation results for conversational recommendation datasets, as reported in Table 1. Based on the presented results, we have several key observations: (1) Overall Performance: Across all datasets and metrics, RAR demonstrates a clear and consistent superiority over traditional, sequential, and SFT baselines, highlighting its robust capability to retrieve and generate high-quality recommendations. In particular, RAR achieves an average improvement of 7.60% over the best baseline results. (2) Impact of Base LLMs: Among the evaluated models enhanced by RAR, GPT delivers the highest overall performance across the datasets, closely followed by Gemini, and then Qwen. This indicates the importance of the LLMs' inherent capabilities, where GPT and Gemini stand as the state-of-the-art closed-source models, while Qwen presents a powerful open-source alternative for conversational recommendation. (3) Comparison to Traditional and SFT Baselines: Overall, baseline comparisons reveal that while SFT-based retrieval augmentation outperforms traditional methods, RAR further surpasses standard fine-tuning. It achieves average gains of 11.9%, 7.5% and 7.7% for Qwen, Gemini and GPT backbones over their SFT-only counterparts, respectively, demonstrating the efficacy of our RL-based post-training. (4) Trends Across Metrics: Although methods exhibit some variability across metrics, the trends are consistent: performance gains of RAR against the best baselines are particularly pronounced on top-k metrics such as N@5 and R@5. For example, the ranking performance improvement for @5 metrics averages 9.33% compared to 5.86% for @10 metrics, suggesting that our preference optimization strategy is especially effective at pushing the most relevant items to the top of the list. Overall, our findings show that RAR effectively integrates retrieval augmentation and RL-based preference optimization to enhance recommendation performance.

RQ2. Which retriever model works best? To analyze the retriever choice in RAR, we evaluate the performance of several different retriever models on the MovieLens dataset. In particular, we pretrain each retriever model and evaluate their retrieval performance

Method	Average Performance			
	N@5	R@5	N@10	R@10
$\text{SimPO}_{\text{Qwen}}$	.0499	.0699	.0575	.0946
$\text{SimPO}_{\text{Gem}}$	.0635	.0825	.0718	.1136
$\text{SimPO}_{\text{GPT}}$	.0727	.1011	.0792	.1203
DPO_{Qwen}	**.0517**	**.0740**	**.0595**	**.0986**
DPO_{Gem}	**.0683**	**.0900**	**.0766**	**.1185**
DPO_{GPT}	**.0754**	**.1023**	**.0830**	**.1260**

Table 4: Performance of SimPO and RAR's online DPO averaged across datasets.

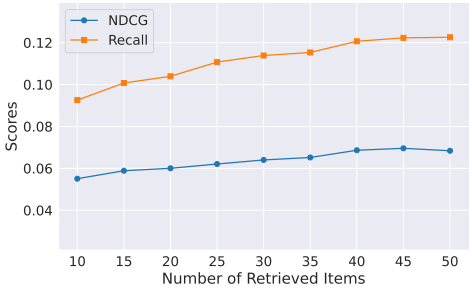

Figure 3: Performance changes in RAR with different number of retrieved items.

on our collected corpus, with results presented in Table 2. Overall, LRURec achieves the highest scores across all reported metrics, recording an N@20 of 0.1700 and an R@20 of 0.3365, thereby outperforming all other retriever models. A similar trend is observed for the @10 metrics, where LRURec scores an N@10 of 0.1483 and an R@10 of 0.2508, establishing a consistent margin over other models. Among the baseline methods, FMLPRec serves as the strongest competitor, achieving the second-best results across most metrics (such as an R@20 of 0.3125 and an R@10 of 0.2411), followed closely by SASRec and BERT4Rec. Considering the efficiency advantages of LRURec in both training and inference, the findings indicate that LRURec is more effective at retrieving candidate items based on user history, leading to higher-quality retrieval outcomes for the following generation stage in RAR.

RQ3. How do different RL algorithms compare? We further analyze the core components of our online preference optimization by comparing two variants of our proposed method: the pairwise online DPO and the multi-sample GRPO (8 samples per example) using the Qwen backbone (Shao et al., 2024). The results are summarized in Table 3. As expected, we observe that our GRPO variant generally outperforms the pairwise DPO across most evaluation metrics and datasets. Specifically, the GRPO variant yields notable improvements on the N@5 metric, achieving 0.0385 compared to DPO's 0.0368 on Reddit. We attribute GRPO's superior performance to its multi-trajectory sampling, which yields robust relative advantage estimates for top-k ranking. Due to GRPO's high inference costs and slower training, we adopt DPO as the default for RAR, which notably maintains 98.6% of GRPO's average performance at a fraction of the computational cost. For DPO variants, we also adopt different formulations of Equation (4) for comparison and present the average results across datasets in Table 4. Here, we report SimPO (see details in Section A) as an alternative (Meng et al., 2025). Overall, RAR with DPO demonstrates consistently superior results compared to SimPO across all evaluated LLM backbones. For instance, RAR with Qwen achieves N@5 and R@5 scores of 0.0517 and 0.0740 respectively, outperforming SimPO on these crucial top-k metrics. This dominating trend holds strictly true for both the Gemini and GPT backbones, where DPO consistently yields the higher performance across metrics. These findings validate that our proposed DPO-based preference optimization is highly effective at capturing nuanced user preferences, highlighting its potential for delivering higher-quality retrieval results in two-stage conversational recommendation.

RQ4. How do key hyperparameters impact the performance of RAR? Here, we investigate the impact of two critical hyperparameters on the performance of RAR: the number of retrieved items and the β value. First, we evaluate the performance changes when varying the number of retrieved candidate items, as illustrated in Figure 3. The analysis reveals a clear, positive trend: as the number of retrieved items increases, both NDCG and Recall steadily improve. Recall naturally benefits from the expanded candidate pool, which increases the likelihood of capturing relevant items. More importantly, the ranking-sensitive metric, NDCG, also maintains an upward trajectory up to 45 items. This demonstrates the LLM generator's robust capacity to effectively filter and rank candidates without being immediately overwhelmed by noise. Furthermore, we examine the influence of the β parameter, as summarized in Table 5. Comparing various settings against the standard SFT baseline ($\beta = 0$), we observe that introducing a small β value significantly enhances recommendation accuracy. Specifically, setting $\beta = 0.05$ achieves peak performance across all evaluated

β	Average Performance			
	N@5	R@5	N@10	R@10
0 (SFT)	.0469	.0702	.0515	.0843
0.05	**.0513**	**.0741**	**.0580**	**.0953**
0.1	.0482	.0692	.0547	.0893
0.2	.0485	.0711	.0544	.0888

Table 5: Recommendation performance of RAR with Qwen using different β values.

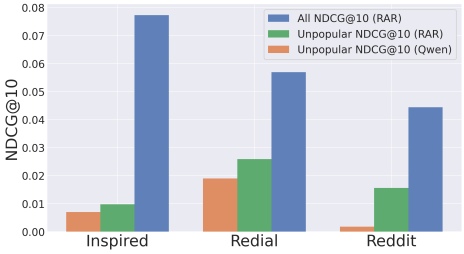

Figure 4: N@10 on different item groups.

Method	Inspired				Redial				Reddit			
	N@5	R@5	N@10	R@10	N@5	R@5	N@10	R@10	N@5	R@5	N@10	R@10
Qwen_{on}	**.0730**	**.0997**	.0764	.1210	.0438	.0626	.0501	.0826	.0367	.0525	.0402	.0629
Qwen_{off}	.0693	.0980	**.0773**	**.1241**	**.0491**	**.0704**	**.0569**	**.0947**	**.0368**	**.0536**	**.0444**	**.0770**
GPT_{on}	.1006	.1327	.1115	.1648	**0.646**	**.0975**	**.0748**	**.1285**	**.0575**	**.0748**	**.0615**	**.0872**
GPT_{off}	**.1091**	**.1422**	**.1180**	**.1700**	.0620	.0932	.0718	.1236	.0536	.0711	.0585	.0859

Table 6: Recommendation performance of LLMs with thinking enabled and disabled.

metrics, improving N@10 from 0.0515 to 0.0580 and R@10 from 0.0843 to 0.0953. Increasing β further leads to a slight decline in performance; therefore, our experiments indicate that the optimal range for β lies within $[0.05, 0.1]$. Overall, these observations suggest that RAR exhibits strong robustness across different hyperparameter choices, maintaining stable, high-quality recommendations without requiring exhaustive, fine-grained tuning.

RQ5. Does explicit reasoning improve LLM performance in CRS? To investigate whether extended reasoning inherently improve recommendation quality, we evaluated Qwen and GPT models with their explicit "thinking" capabilities enabled (on) and disabled (off). As detailed in Table 6, the impact of reasoning is not universally beneficial and varies significantly across scenarios. For the Qwen model, omitting the explicit reasoning phase consistently delivered competitive or superior performance across most datasets while significantly reducing inference costs. Conversely, GPT's behavior was highly dataset-dependent: GPT_{off} achieved the highest metrics on the Inspired dataset, whereas GPT_{on} performed strictly better on both Redial and Reddit. Ultimately, these findings indicate that forcing additional reasoning does not guarantee better recommendations; rather, its utility is dependent on both the underlying model and the specific data distribution.

RQ6. Does RAR improve popularity bias and hallucination? Finally, we analyze RAR (with the Qwen f_{llm}) on two known LLM-based CRS challenges: popularity bias and item hallucination (He et al., 2023; Yang & Chen, 2024). By categorizing items unseen in training into an unpopular group (Figure 4), we confirm popularity bias exists; overall NDCG@10 scores substantially exceed those of unpopular items (e.g., Reddit scores just 0.0013). Nevertheless, RAR generally outperforms the LLM-only Qwen baseline. Leveraging a retriever with rich semantic embeddings and online RL-based post-training yields a nearly 4x improvement on unpopular items over Qwen. Additionally, we observe a dramatic reduction in hallucination rates, with under 1% unmatched titles across all datasets. Ultimately, while not completely immune to popularity bias, RAR's retrieval augmentation design effectively mitigates hallucination, marking a critical step toward more reliable and trustworthy CRS.

5 Conclusion

In this work, we introduce RAR, a two-stage framework for retrieval augmented conversational recommendation. For this purpose, we collect a unified text corpus of over 300k films within the movie domain to support our recommendation tasks. RAR employs a lightweight retriever to efficiently identify potential candidate items based on user interaction history. In addition, a black-box LLM is utilized to refine these candidates by integrating

contextual data and retrieved information, thereby capturing subtle user preferences in natural language. To further align retrieval with generation, our online, on-policy preference optimization leverages LLM feedback to iteratively enhance the retriever's performance. Extensive experiments on benchmark datasets validate our approach, demonstrating that by effectively aligning the retriever and generator through online preference optimization, RAR consistently outperforms state-of-the-art baselines in conversational recommendation.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, et al. Improving language models by retrieving from trillions of tokens. In *International conference on machine learning*, pp. 2206–2240. PMLR, 2022.

BrightData. Imdb media. https://huggingface.co/datasets/BrightData/IMDb-Media. Accessed: December 2024.

Jianlyu Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. M3-embedding: Multi-linguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 2318–2335, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.137. URL https://aclanthology.org/2024.findings-acl.137/.

Qibin Chen, Junyang Lin, Yichang Zhang, Ming Ding, Yukuo Cen, Hongxia Yang, and Jie Tang. Towards knowledge-based recommender dialog system. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 1803–1813, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1189. URL https://aclanthology.org/D19-1189/.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Yue Feng, Shuchang Liu, Zhenghai Xue, Qingpeng Cai, Lantao Hu, Peng Jiang, Kun Gai, and Fei Sun. A large language model enhanced conversational recommender system. *arXiv preprint arXiv:2308.06212*, 2023.

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

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. Retrieval augmented language model pre-training. In *International conference on machine learning*, pp. 3929–3938. PMLR, 2020.

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

Shirley Anugrah Hayati, Dongyeop Kang, Qingxiaoyang Zhu, Weiyan Shi, and Zhou Yu. Inspired: Toward sociable recommendation dialog systems. *arXiv preprint arXiv:2009.14306*, 2020.

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

Zhankui He, Zhouhang Xie, Harald Steck, Dawen Liang, Rahul Jha, Nathan Kallus, and Julian McAuley. Reindex-then-adapt: Improving large language models for conversational recommendation. *arXiv preprint arXiv:2405.12119*, 2024.

B Hidasi. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939*, 2015.

Karl Higley, Even Oldridge, Ronay Ak, Sara Rabhi, and Gabriel de Souza Pereira Moreira. Building and deploying a multi-stage recommender system with merlin. In *Proceedings of the 16th ACM Conference on Recommender Systems*, pp. 632–635, 2022.

Jian Hu. Reinforce++: A simple and efficient approach for aligning large language models. *arXiv preprint arXiv:2501.03262*, 2025.

Jonas Hübotter, Frederike Lübeck, Lejs Behric, Anton Baumann, Marco Bagatella, Daniel Marta, Ido Hakimi, Idan Shenfeld, Thomas Kleine Buening, Carlos Guestrin, et al. Reinforcement learning via self-distillation. *arXiv preprint arXiv:2601.20802*, 2026.

Zheng Hui, Xiaokai Wei, Yexi Jiang, Kevin Gao, Chen Wang, Se-eun Yoon, Rachit Pareek, and Michelle Gong. Toward safe and human-aligned game conversational recommendation via multi-agent decomposition. In *Findings of the Association for Computational Linguistics: EACL 2026*, pp. 4568–4584, 2026.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

Gautier Izacard and Édouard Grave. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pp. 874–880, 2021.

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. Atlas: Few-shot learning with retrieval augmented language models. *Journal of Machine Learning Research*, 24(251):1–43, 2023.

Dietmar Jannach, Ahtsham Manzoor, Wanling Cai, and Li Chen. A survey on conversational recommender systems. *ACM Computing Surveys (CSUR)*, 54(5):1–36, 2021.

Hyunsik Jeon, Satoshi Koide, Yu Wang, Zhankui He, and Julian McAuley. Adapting large vision-language models to visually-aware conversational recommendation. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 1037–1048, 2025.

Zhengbao Jiang, Frank F Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. Active retrieval augmented generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 7969–7992, 2023.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *arXiv preprint arXiv:2503.09516*, 2025.

Dongyeop Kang, Anusha Balakrishnan, Pararth Shah, Paul Crook, Y-Lan Boureau, and Jason Weston. Recommendation as a communication game: Self-supervised bot-play for goal-oriented dialogue. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 1951–1961, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1203. URL https://aclanthology.org/D19-1203/.

Wang-Cheng Kang and Julian McAuley. Self-attentive sequential recommendation. In *2018 IEEE international conference on data mining (ICDM)*, pp. 197–206. IEEE, 2018.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6769–6781, 2020.

Sara Kemper, Justin Cui, Kai Dicarlantonio, Kathy Lin, Danjie Tang, Anton Korikov, and Scott Sanner. Retrieval-augmented conversational recommendation with prompt-based semi-structured natural language state tracking. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 2786–2790, 2024.

Wenqiang Lei, Xiangnan He, Yisong Miao, Qingyun Wu, Richang Hong, Min-Yen Kan, and Tat-Seng Chua. Estimation-action-reflection: Towards deep interaction between conversational and recommender systems. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, pp. 304–312, 2020.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive NLP tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474, 2020.

Chuang Li, Yang Deng, Hengchang Hu, Min-Yen Kan, and Haizhou Li. Incorporating external knowledge and goal guidance for llm-based conversational recommender systems. *arXiv preprint arXiv:2405.01868*, 2024.

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

Xi Victoria Lin, Xilun Chen, Mingda Chen, Weijia Shi, Maria Lomeli, Richard James, Pedro Rodriguez, Jacob Kahn, Gergely Szilvasy, Mike Lewis, et al. RA-DIT: Retrieval-augmented dual instruction tuning. In *The Twelfth International Conference on Learning Representations*, 2024.

Yu Lu, Junwei Bao, Yan Song, Zichen Ma, Shuguang Cui, Youzheng Wu, and Xiaodong He. RevCore: Review-augmented conversational recommendation. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (eds.), *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pp. 1161–1173, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-acl.99. URL https://aclanthology.org/2021.findings-acl.99/.

Jiaqi Ma, Zhe Zhao, Xinyang Yi, Ji Yang, Minmin Chen, Jiaxi Tang, Lichan Hong, and Ed H Chi. Off-policy learning in two-stage recommender systems. In *Proceedings of The Web Conference 2020*, pp. 463–473, 2020.

Yu Meng, Mengzhou Xia, and Danqi Chen. Simpo: Simple preference optimization with a reference-free reward. *Advances in Neural Information Processing Systems*, 37:124198–124235, 2024.

Yu Meng, Mengzhou Xia, and Danqi Chen. Simpo: Simple preference optimization with a reference-free reward. *Advances in Neural Information Processing Systems*, 37:124198–124235, 2025.

Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pp. 1928–1937. PmLR, 2016.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022.

Boci Peng, Yun Zhu, Yongchao Liu, Xiaohe Bo, Haizhou Shi, Chuntao Hong, Yan Zhang, and Siliang Tang. Graph retrieval-augmented generation: A survey. *arXiv preprint arXiv:2408.08921*, 2024.

Robin L Plackett. The analysis of permutations. *Journal of the Royal Statistical Society Series C: Applied Statistics*, 24(2):193–202, 1975.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36:53728–53741, 2023.

Chidambara Raju. Imdb movies dataset based on genre. https://www.kaggle.com/datasets/rajugc/imdb-movies-dataset-based-on-genre. Accessed: December 2024.

Machel Reid, Nikolay Savinov, Denis Teplyashin, Dmitry Lepikhin, Timothy Lillicrap, Jean-baptiste Alayrac, Radu Soricut, Angeliki Lazaridou, Orhan Firat, Julian Schrittwieser, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024.

Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 3982–3992, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1410. URL https://aclanthology.org/D19-1410/.

Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher D Manning. RAPTOR: Recursive abstractive processing for tree-organized retrieval. In *The Twelfth International Conference on Learning Representations*, 2024.

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

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Richard James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. REPLUG: Retrieval-augmented black-box language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 8364–8377, 2024.

Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, et al. Openai gpt-5 system card. *arXiv preprint arXiv:2601.03267*, 2025.

Rohan Surana, Junda Wu, Zhouhang Xie, Yu Xia, Harald Steck, Dawen Liang, Nathan Kallus, and Julian McAuley. From reviews to dialogues: Active synthesis for zero-shot llm-based conversational recommender system. *arXiv preprint arXiv:2504.15476*, 2025.

Richard S Sutton, Andrew G Barto, et al. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems*, 12, 1999.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10014–10037, 2023.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*, 2022a.

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

Yueqi Wang, Zhenrui Yue, Huimin Zeng, Dong Wang, and Julian McAuley. Train once, deploy anywhere: Matryoshka representation learning for multimodal recommendation. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 13461–13472, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.786. URL https://aclanthology.org/2024.findings-emnlp.786/.

Junda Wu, Cheng-Chun Chang, Tong Yu, Zhankui He, Jianing Wang, Yupeng Hou, and Julian McAuley. Coral: Collaborative retrieval-augmented large language models improve long-tail recommendation. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 3391–3401, 2024.

Shusheng Xu, Wei Fu, Jiaxuan Gao, Wenjie Ye, Weilin Liu, Zhiyu Mei, Guangju Wang, Chao Yu, and Yi Wu. Is dpo superior to ppo for llm alignment? a comprehensive study. *arXiv preprint arXiv:2404.10719*, 2024.

Shi-Qi Yan, Jia-Chen Gu, Yun Zhu, and Zhen-Hua Ling. Corrective retrieval augmented generation. *arXiv preprint arXiv:2401.15884*, 2024.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Ting Yang and Li Chen. Unleashing the retrieval potential of large language models in conversational recommender systems. In *Proceedings of the 18th ACM Conference on Recommender Systems*, pp. 43–52, 2024.

Ori Yoran, Tomer Wolfson, Ori Ram, and Jonathan Berant. Making retrieval-augmented language models robust to irrelevant context. In *The Twelfth International Conference on Learning Representations*, 2024.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.

Zhenrui Yue, Yueqi Wang, Zhankui He, Huimin Zeng, Julian McAuley, and Dong Wang. Linear recurrent units for sequential recommendation. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, pp. 930–938, 2024a.

Zhenrui Yue, Honglei Zhuang, Aijun Bai, Kai Hui, Rolf Jagerman, Hansi Zeng, Zhen Qin, Dong Wang, Xuanhui Wang, and Michael Bendersky. Inference scaling for long-context retrieval augmented generation. *arXiv preprint arXiv:2410.04343*, 2024b.

Zhenrui Yue, Bowen Jin, Huimin Zeng, Honglei Zhuang, Zhen Qin, Jinsung Yoon, Lanyu Shang, Jiawei Han, and Dong Wang. Hybrid latent reasoning via reinforcement learning. *arXiv preprint arXiv:2505.18454*, 2025.

Zhenrui Yue, Kartikeya Upasani, Xianjun Yang, Suyu Ge, Shaoliang Nie, Yuning Mao, Zhe Liu, and Dong Wang. Dr. zero: Self-evolving search agents without training data. *arXiv preprint arXiv:2601.07055*, 2026.

Gholamreza Zare and P Malekpour Alamdari. Conversational graph-llm reasoning for interactive preference modeling and explainable recommendation, 2025.

Xiaoyu Zhang, Xin Xin, Dongdong Li, Wenxuan Liu, Pengjie Ren, Zhumin Chen, Jun Ma, and Zhaochun Ren. Variational reasoning over incomplete knowledge graphs for conversational recommendation. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*, pp. 231–239, 2023.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, et al. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*, 2025.

Yongfeng Zhang, Xu Chen, Qingyao Ai, Liu Yang, and W Bruce Croft. Towards conversational search and recommendation: System ask, user respond. In *Proceedings of the 27th acm international conference on information and knowledge management*, pp. 177–186, 2018.

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

Kun Zhou, Hui Yu, Wayne Xin Zhao, and Ji-Rong Wen. Filter-enhanced mlp is all you need for sequential recommendation. In *Proceedings of the ACM web conference 2022*, pp. 2388–2399, 2022.

Lixi Zhu, Xiaowen Huang, and Jitao Sang. A llm-based controllable, scalable, human-involved user simulator framework for conversational recommender systems. In *Proceedings of the ACM on Web Conference 2025*, pp. 4653–4661, 2025.

Attribute	Value
ID	tt0111161
Title	The Shawshank Redemption
Year	1994
Genre	Drama
Director	Frank Darabont
Cast	Tim Robbins, Morgan Freeman ...
Plot	Chronicles the experiences of a ...

Table 7: An example movie with metadata in our corpus.

A Appendix

A.1 Corpus Collection

We report the details on the collection and processing of our unified movie text corpus. The process began by gathering an initial set of movie titles and identifiers (e.g., IMDb/TMDB) from multiple open-source datasets, including IMDb genre, IMDb media, and Inspired (Hayati et al., 2020; Raju; BrightData). We then augmented this initial set with movies mentioned in widely-used recommendation benchmarks, namely MovieLens, Redial and Reddit (Harper & Konstan, 2015; Li et al., 2018; He et al., 2023). A primary motivation for building a consolidated, offline corpus was to overcome the limitations of existing APIs. While services like TMDB[2] provide extensive data, their APIs operate on a per-query basis with strict rate limits, making it infeasible to fetch metadata for all titles via individual HTTP requests. Furthermore, API responses are not directly linked to the specific movie mentions within our target conversational datasets. Our approach addresses these issues by creating a versioned, fully annotated dataset that can be indexed and processed in a single pass, thereby eliminating rate-limit bottlenecks and ensuring direct correspondence between metadata and conversational mentions. To construct this corpus, we first consolidated duplicated entries by verifying movie names and identifiers, resolving any conflicts by selecting the most informative metadata. For items with missing information, we carefully collected the missing values from TMDB or other online resources. We deliberately avoided direct scraping of copyrighted text fields from IMDb due to licensing restrictions[3]. Subsequently, we mapped the items in our corpus to their corresponding entries in the recommendation datasets (i.e., MovieLens, Inspired, Redial, and Reddit). This involved performing entity recognition and string matching to identify each movie mentioned in the conversational or sequential data and link it to its entry in our corpus. Any movies for which we could not find essential metadata (director, cast, genre, plot) were removed to maintain data quality. After processing, our final corpus contains 337,731 movie entries with comprehensive metadata, focusing primarily on English-language films (an example is shown in Table 7). The collection spans a wide temporal range, from films made as early as 1888 to upcoming releases in 2029, representing roughly half of all films listed on IMDb as of December 2024. By establishing explicit item correspondence to each of the benchmarks, the collected corpus serves as a robust and readily accessible knowledge base to train both the retriever and LLM components in RAR.

A.2 Implementation & Additional Results

Datasets. Our model is evaluated on three widely-used benchmark datasets for conversational recommendation: Inspired, Redial and Reddit (Hayati et al., 2020; Li et al., 2018; He et al., 2023). To ensure the retriever has a robust initial understanding of the movie domain, we pretrain it on the MovieLens-20M dataset (Harper & Konstan, 2015). For the pretraining, we segment the dataset into short user sessions, using a 30-minute inactivity threshold to generate meaningful interaction sequences. During preprocessing, we construct input

[2]https://www.themoviedb.org/
[3]https://developer.imdb.com/

Dataset	#Train	#Val	#Test
MovieLens	536,127	67,015	67,015
Inspired	1,507	206	183
Redial	24,095	2,647	3,445
Reddit	12,481	2,947	1,511

Table 8: Dataset Statistics.

prompts with conversational context and incorporate additional retrieval augmentation from the collected corpus (see Section 3.2). For items retrieved during this process, we utilize a comprehensive set of attributes: *title*, *year*, *genre*, *director*, *cast* and *plot*. These attributes serve a critical dual purpose: they are first used to generate the initial item embeddings for training the retriever model, and subsequently, they function as structured context for the LLM generator. Detailed statistics for each dataset, including the number of training (#Train), validation (#Val) and test (#Test) examples, are reported in Table 8.

Baselines. For baseline models, we adopt *text-based* SASRec, FMLP-Rec and LRURec following the fMRLRec implementation (Kang & McAuley, 2018; Zhou et al., 2022; Yue et al., 2024a; Wang et al., 2024). We also utilize *supervised fine-tuning* (SFT) to learn the retriever model in combination with LLM as baselines. This approach aligns with the objective in sequential recommendation and maximizes the likelihood of ground truth items w.r.t. θ. We report the details of baseline methods:

- *Knowledge-Based Recommender Dialog (KBRD)* integrates knowledge graph to understand conversational context and user preferences, enabling multi-turn reasoning over entities for recommendation (Chen et al., 2019).
- *Knowledge Graph-based Semantic Fusion (KGSF)* utilizes knowledge graph and a gated fusion mechanism to dynamically integrate semantic information from both conversation and item attributes (Zhou et al., 2020).
- *Unified Conversational Recommender System (UniCRS)* presents a unified framework to handle diverse conversational goals, including recommendation and chitchat with a prompt-based approach via a shared encoder-decoder model (Wang et al., 2022b).
- *Self-Attentive Sequential Recommendation (SASRec)* is the first transformer-based sequential recommender. SASRec uses unidirectional self-attention to capture transition patterns (Kang & McAuley, 2018).
- *Filter-enhanced MLP for Recommendation (FMLP-Rec)* also adopts an all-MLP architecture with filter-enhanced layers. FMLP-Rec applies fast Fourier transform to improve representation learning (Zhou et al., 2022).
- *Linear Recurrence Units for Recommendation (LRURec)* is based on linear recurrence and is optimized for paralleled training. LRURec thus provides both efficient training and inference speed (Yue et al., 2024a).

All models are implemented and trained according to the methodologies described in the original works, with unspecified hyperparameters used as recommended. For item encoding, we use the Qwen-8B embedding model and encode items with their available metadata in the format of key-value pairs (Zhang et al., 2025). For each recommender / retriever, we initialize with two layers and search the dropout rates among [0.2, 0.4]. The retriever models are pretrained on MovieLens 20M using an 8:1:1 train / validation / test split, where the data is split into sessions with a maximum time gap of 30 minutes. In pretraining, we sample 100 negative examples at each time step and utilize in-batch negatives to compute the negative log likelihood loss. The models with best validation performance are saved and evaluated on the test sets.

For SFT baselines and RAR, we adopt LRURec as the retriever and further optimize the retriever model with the proposed online, on-policy preference optimization. The adopted

Example Prompt

You are an expert in movie recommendations. Analyze the provided conversation history to identify the user's preferences, such as genres and actors. Then, rank the candidate movies by how well they match these preferences. Return your answer as a numbered list with each movie on a new line in the format: '<rank>. <movie name>'. Do not include any additional commentary, formatting or chattiness.

`<Retrieved Candidates w/ Metadata>`

Conversation history:

`<Conversation Context>`

Figure 5: Example prompt for RAR. The prompt comprises of instructions, retrieved candidates, followed by the conversation context.

Algorithm 1 Preference Optimization in RAR

Require: Conversational dataset $\mathcal{X}$, pretrained retriever f_{ret} (policy π_θ), frozen LLM generator f_{llm}, reward function $r(\cdot,\cdot)$, group size g ($g = 2$ for DPO), hyperparameters β, learning rate η.
Ensure: Optimized retriever parameters θ.

Initialize policy π_θ with parameters from pretrained f_{ret}.
Initialize reference policy $\pi_{\text{ref}} \leftarrow \pi_\theta$. ▷ Copy initial weights
for each training epoch **do**
  for each conversation $\mathcal{C} = (r_t, s_t, I_t)_{t=1}^{T}$ in $\mathcal{X}$ **do**
    for $t = 1, \ldots, T$ **do**
      if r_t is *recommender* **then**
        Let history items be $\mathcal{I}_{\text{hist}} \leftarrow \{I_\tau\}_{\tau=1}^{t-1}$.
        Let history sequence be $\mathcal{S}_{\text{hist}} \leftarrow \{s_k\}_{k=1}^{t-1}$.
        Sample $\{C_i\}_{i=1}^{g} \sim \pi_\theta(\cdot|\mathcal{I}_{\text{hist}})$.
        Initialize rewards list $R \leftarrow []$.
        for $i = 1, \ldots, g$ **do**
          Generate ranked list $\hat{I}_i \leftarrow f_{\text{llm}}(\mathcal{S}_{\text{hist}}, C_i)$.
          Calculate reward $r_i \leftarrow r(\hat{I}_i, I_t)$. ▷ e.g., NDCG as reward
          Append r_i to R.
        Compute $\mathcal{L}_{\text{rl}}$ with Equation (4) or Equation (5).
        Compute NLL loss $\mathcal{L}_{\text{nll}} \leftarrow -\log \pi_\theta(I_t|\mathcal{I}_{\text{hist}})$.
        Compute final loss $\mathcal{L} \leftarrow \mathcal{L}_{\text{nll}} + \mathcal{L}_{\text{rl}}$.
        Update parameters $\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}$.
return Optimized parameters θ.

LLMs are Qwen (Qwen3-8B), GPT-5 mini (gpt-5-mini-2025-08-07) and Gemini 3 Flash (gemini-3-flash-preview) (Yang et al., 2025; Singh et al., 2025)[4]. For training sets that are greater than 2.5k, we subsample to 2.5k for efficiency. Similarly, we search dropout rates among [0.1, 0.2]. Our learning rate is searched from [5e-5, 1e-4] and we adopt cosine scheduling for learning rate with warm-up steps of 100. SFT baseline models are trained without $\mathcal{L}_{\text{dpo}}$ while in RAR, the β values are searched within [0.05, 0.1, 0.2]. After sampling from the retriever, we annotate the candidate sets C_1 and C_2 using the LLM model. If any label item appears in both sets, we select the one with the higher label rank as C_w. If a label item is present in only one candidate set, that set becomes C_w. If neither set contains a label item, we resample until one of these conditions is met. Similar to baseline implementation, we keep models with best validation performance for evaluation. Our prompt is constructed as illustrated in Figure 5. The LLM generation configuration is set to

[4]https://deepmind.google/technologies/gemini/flash

Method	Inspired				Redial				Reddit			
	N@5	R@5	N@10	R@10	N@5	R@5	N@10	R@10	N@5	R@5	N@10	R@10
SFT_{Qwen}	.0741	.0869	.0741	.0869	.0507	.0736	.0543	.0843	.0291	.0410	.0305	.0450
SFT_{GPT4o}	.0547	.0652	.0600	.0815	.0514	.0742	.0585	.0959	.0340	.0467	.0371	.0556
SFT_{Gem2}	.0793	.0867	**.1032**	.1250	.0588	.0870	.0684	.1163	.0418	.0543	.0443	.0623
RAR_{Qwen}	.0793	.1032	.0793	.1032	.0540	.0780	.0583	.0907	.0326	.0444	.0330	.0457
RAR_{GPT4o}	.0768	.0978	.0820	.1141	.0522	.0771	.0595	.0994	.0369	.0510	.0393	.0582
RAR_{Gem2}	**.0831**	**.1087**	.0995	**.1576**	**.0620**	**.0934**	**.0755**	**.1349**	**.0449**	**.0583**	**.0477**	**.0669**

Table 9: Additional results of RAR. For clarity, the best results for each dataset and metric are highlighted in **bold**, while the second-best results are underlined.

default, with the thinking efforts setting to `low` or `None` in our main experiments. All baseline methods and RAR are evaluated under identical conditions. Based on the LLM prediction, we perform string matching to compute the highest rank of label items, evaluation metrics are implemented following (He et al., 2023).

For alternative RL methods, we adopt the pairwise SimPO (Meng et al., 2024) and the multi-sample GRPO (Shao et al., 2024). GRPO details can be found in our main text, and the formulation for SimPO in our case can be formulated as (Meng et al., 2025):

$$\mathcal{L}_{\text{simpo}} = -\log \sigma(\beta \log P_\theta(C_w|\{I_\tau\}_{\tau=1}^{t-1}) - \beta \log P_\theta(C_l|\{I_\tau\}_{\tau=1}^{t-1}) - \gamma), \tag{7}$$

where the the log probabilities are computed similar to Equation (3). Hyperparameters such as γ are searched as recommended in the original paper. For GRPO, we set the group size to eight and compute the loss using Equation (5). Across all methods, we adhere to an online, on-policy setting to train and evaluate the retriever models. While our methodology formally introduces the reference model π_{ref} in accordance with standard DPO, our final experiments omit this reference constraint. We empirically found that removing the reference model yields superior recommendation performance, a phenomenon consistent with recent findings in RLVR literature (Yue et al., 2025; Yu et al., 2025). Specifically, we omit the reference model for two primary reasons: (1) our joint optimization with the supervised negative log-likelihood loss ($\mathcal{L}_{nll}$) serves as a highly effective surrogate to prevent policy drift and maintain in-domain performance, rendering the KL penalty from a reference model largely redundant; and (2) while retriever models fundamentally require dropout to achieve optimal generalization, the inconsistency introduced by dropout destabilizes the log-likelihood calculations of the reference model, thereby failing to provide a stable or effective constraint on the policy. To compute hallucination rates, we perform fuzzy matching and consider a prediction to be hallucination if its similarity score to any of the provided candidate titles is below 0.85.

We present additional results to demonstrate that RAR generalizes well to other recent large language models. Specifically, we evaluate Qwen 2.5 (Qwen2.5-7B-Instruct), GPT-4o mini (gpt-4o-mini-2024-07-18), and Gemini 2.0 Flash (gemini-2.0-flash-001) (Yang et al., 2024; Hurst et al., 2024; Comanici et al., 2025). The results are presented in Table 9, from which we observe: (1) First, applying our method (RAR) consistently improves recommendation performance over the standard SFT baseline across all three foundation models. Whether using Qwen 2.5, GPT-4o mini or Gemini 2.0 Flash, the RAR variants yield higher NDCG and Recall scores in almost every scenario. (2) the RAR formulation utilizing Gemini 2.0 Flash (RAR_{Gem2}) achieves the best overall results, securing the highest scores across all metrics for the Redial and Reddit datasets, as well as three out of four metrics for the Inspired dataset. In summary, these findings confirm that our approach is model-agnostic and generalizes highly effectively to various state-of-the-art large language models, reliably boosting performance beyond standard fine-tuning techniques.