



Illuminating the Path: Reachability-Aware Supervision Curation for LLM-Based Recommendation

Donghee Han Jiwon Jeong Hwanjun Song[†] Mun Yong Yi[†]

Korea Advanced Institute of Science and Technology, Daejeon, Republic of Korea
{handonghee, zzioni, songhwanjun, munyi}@kaist.ac.kr

[†]Corresponding author

Abstract

Eliciting high-quality reasoning trajectories to supervise Large Language Models (LLMs) in sequential recommendation is notoriously difficult. In massive item pools, standard data curation methods such as autonomous exploration or rejection sampling suffer from severe sparsity. LLMs’ context-utilization bias exacerbate this challenge by overusing noisy interaction histories, yielding diffuse and weakly grounded reasoning paths that frequently fail to retrieve the target item. To overcome this data curation bottleneck, we propose GEARec (Grounded Reasoning Curation for Anchor-based Recommendation), a framework based on Dual-Perspective Anchor Extraction and Hybrid Reasoning Generation. By combining the semantic strengths of LLMs with collaborative filtering (CF) signals, GEARec identifies a high-relevance anchor item from the user’s history to construct highly focused, reachability-aware reasoning trajectories for supervised fine-tuning. Experiments on multiple datasets demonstrate that GEARec effectively curates superior supervision signals, reducing context overuse and consistently outperforming strong baselines in recommendation accuracy.

1 Introduction

Large Language Models (LLMs) have recently been introduced to recommender systems to enable deep, semantic reasoning (Yang et al., 2024; Lin et al., 2025b). However, when deployed in real-world recommendation environments characterized by highly noisy data, they exhibit a recurring failure pattern (Xu et al., 2025a; Wang et al., 2025). Specifically, LLMs often exhibit context-utilization bias, a fundamental tendency to forcefully establish logical connections across all provided context (Shi et al., 2023; Ravaut et al., 2024; Tian et al., 2023; Valkanova and Yordanov, 2024). In sequential recommendation, where a

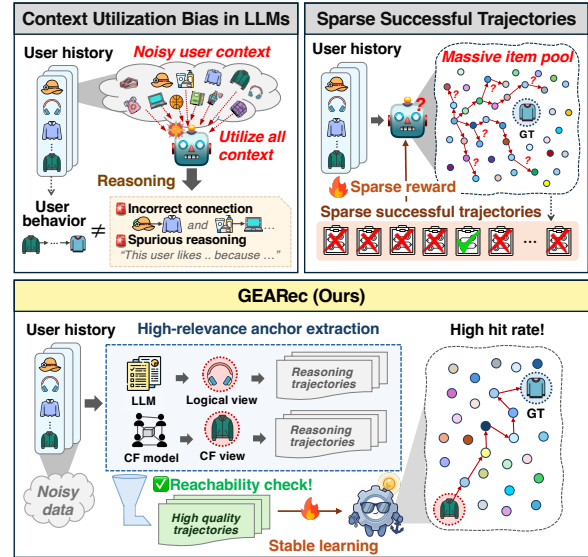


Figure 1: Existing methods fail due to context-utilization bias and reward sparsity, whereas our framework discovers a valid reasoning path guided by a high-relevance anchor.

user’s extensive interaction history is fed into the prompt, this bias compels the model to artificially fit random clicks or meaningless past interactions (noise) into its justification for the target item (Liu et al., 2024; Hou et al., 2024; Han et al., 2025; Turpin et al., 2023). Consequently, this leads to hallucinated reasoning that is logically disjointed and entirely disconnected from the user’s intent.

This bias precipitates severe exploration challenges when applying prevalent alignment techniques, such as Reinforcement Learning (RL) or Rejection Sampling (RS), to the recommendation domain (Tsai et al., 2024; Lin et al., 2025a). In conventional point-wise prediction tasks (e.g., rating prediction) where the action space is extremely narrow, entrusting models with autonomous exploration was viable because the probability of hitting the correct answer remained sufficiently high for optimization (Tsai et al., 2024; Kong et al., 2026; Fang et al., 2025). However, in practical,

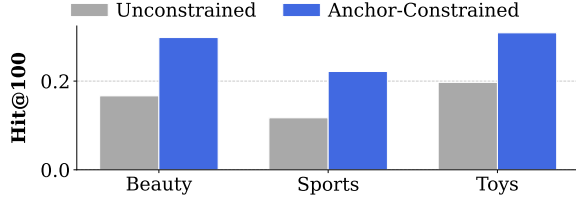


Figure 2: Best-of- N ($N=8$) reasoning under Unconstrained vs. Anchor-Constrained settings (Hit@100).

large-scale recommendation environments encompassing thousands of items in the candidate pool, depending entirely on the LLMs’ exploration capabilities falls critically short and leads to reward sparsity (Lin et al., 2025a). As context-utilization bias forces the model to over-interpret noise, the probability of generating incorrect paths surges, leading to an exceptionally low rate of successful trajectories (Yang et al., 2025; Zhao et al., 2025). This bottleneck inherently starves both RL exploration and supervised fine-tuning (SFT) of valid, target-reaching reasoning paths.

As illustrated in Figure 1, to mitigate these issues, we propose a novel data curation framework: **Grounded REasoning Curation for Anchor-based Recommendation (GEARec)**. Rather than relying on unconstrained random exploration in large search spaces, this framework explicitly identifies a core item (an anchor) and generates grounded reasoning. We empirically verified the importance of anchor-grounded reasoning through preliminary experiments (Figure 2, Appendix A.1): focusing the reasoning on a single anchor doubles the hit rate, highlighting the critical role of core item identification in the user’s history. This finding motivates us to establish anchor-grounded reasoning and target reachability as the core criteria for curating supervision data.

GEARec constructs a high-quality supervision dataset through a rigorous three-phase pipeline. First, **Dual-Perspective Anchor Extraction and Reasoning Generation** leverages both an LLM’s semantic understanding and a Collaborative Filtering (CF) model’s implicit signals to extract the most relevant historical anchor and generate diverse reasoning candidates. Second, **Reachability-Aware Trajectory Routing** acts as a strict empirical filter, evaluating these candidates against the massive item pool and discarding reasonings that fail to retrieve the actual target item. Through this process, we effectively block error propagation caused by flawed reasoning. Finally, **Alignment via Curated Trajectories** fine-tunes

the LLM exclusively on this reachability-verified dataset, training the LLM to effectively generate accurate recommendation paths based on the user’s key interaction.

To validate the efficacy of our approach, we conduct extensive experiments across diverse real-world recommendation domains. Empirical results demonstrate that the proposed framework significantly outperforms state-of-the-art LLM-based recommendation systems. Notably, these performance gains extend beyond overall accuracy; our method exhibits substantial improvements in recommending long-tail items, thereby confirming its capacity to effectively enhance recommendation diversity. Furthermore, comprehensive evaluations of reasoning validity and fidelity verify that our framework generates significantly more target-relevant outputs compared to baselines.

The contributions of our work are as follows:

- **Problem Diagnosis:** We identify context-utilization bias in LLM-based recommendation, where models fail to ground reasoning on the most decisive evidence.
- **Data Curation Framework:** We propose GEARec, a grounded data curation framework that combines dual-perspective anchor extraction and reachability-aware trajectory routing to construct higher-quality reasoning trajectories for training.
- **Empirical Validation:** Experiments on multiple real-world datasets show that GEARec improves recommendation performance over strong baselines, while also yielding favorable long-tail and backbone robustness.

2 Related Work

2.1 Sequential Recommendation

As a primary paradigm in recommendation systems, sequential recommendation aims to capture the chronological dependencies in user behavior. Foundational architectures, including GRU4Rec (Hidasi, 2015), SASRec (Kang and McAuley, 2018), and BERT4Rec (Sun et al., 2019), have proven highly effective at capturing the temporal dynamics of user-item interactions. Models such as FDSA (Zhang et al., 2019) and S³-Rec (Zhou et al., 2020) integrate text and contextual signals to further elevate recommendation performance.

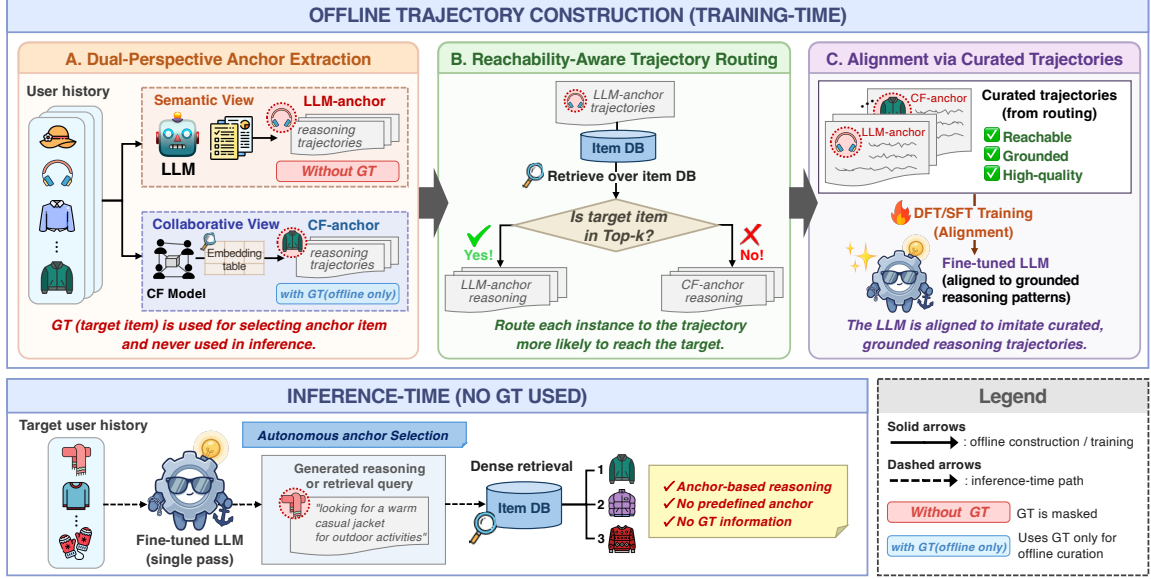


Figure 3: Overview of GEARec. (A) Anchor-grounded trajectory generation produces candidates from two views, a semantic view and a collaborative view. (B) Reachability-aware routing keeps a trajectory only if it retrieves the target within Top-K. (C) Alignment via DFT and semantic-to-collaborative projection. Curated reasoning trajectories are used for fine-tuning the LLM.

2.2 LLM-based Recommendation Systems

Early LLM-based recommenders primarily functioned as zero-shot rerankers (Hou et al., 2024) or binary classifiers (Bao et al., 2023). Recently, generative and policy-optimized frameworks have emerged. BIGRec (Bao et al., 2025) introduced SFT to generate retrieval queries, while RL methods such as Rec-R1 (Lin et al., 2025a) and RecZero (Kong et al., 2026) utilize GRPO to optimize rank-based metrics and rating prediction accuracy, respectively.

Despite these strides, existing reasoning-enhanced LLMs exhibit structural limitations that preclude their deployment in large-scale sequential recommendation. Crucially, models such as RecZero (Kong et al., 2026), Rec-SAVER (Tsai et al., 2024), and Reason4Rec (Fang et al., 2025) retreat to heavily simplified point-wise rating prediction tasks. By artificially restricting the action space to evaluating a single predefined item, they often overlook the fundamental challenge of navigating a massive global item pool. Furthermore, in actual sequential recommendation scenarios, methods such as Rec-R1 suffer from extreme reward sparsity.

2.3 Positioning of GEARec

Existing LLM-based recommendation studies primarily focus on *how to utilize* LLM-derived signals. This is typically achieved either by **zero-shot**

query generation (Han et al., 2025) to fuse semantic queries with collaborative representations, or via **reasoning distillation** (Wang et al., 2024; Han et al., 2026) to compress LLM knowledge into smaller recommenders. However, these downstream applications remain vulnerable to the initial context-utilization bias inherent in LLM generation. GEARec introduces an orthogonal paradigm: **training-data quality control**. Instead of asking how to use the signals, we ask a question that is logically upstream of both approaches: *which LLM-generated trajectories are trustworthy enough to serve as supervision?* By enforcing our reachability criterion, GEARec curates a high-fidelity dataset that filters out ungrounded noise. Consequently, our curated trajectories can seamlessly feed into existing query-driven pipelines or reasoning distillers, enhancing their performance without architecture changes.

3 Methodology: GEARec

Figure 3 illustrates an overview of **Grounded REasoning Curation for Anchor-based Recommendation (GEARec)**, a data curation framework designed to overcome context-utilization bias and exploration collapse. The training pipeline consists of three phases: (1) Dual-Perspective Anchor Extraction and Reasoning Generation, where we generate candidate trajectories via an LLM and a CF-based model;

(2) Reachability-Aware Trajectory Routing, which selects the optimal reasoning path based on Top- K hit success; and (3) Alignment via Curated Trajectories, where the selected reasoning is used for fine-tuning.

3.1 Dual-Perspective Anchor Extraction and Reasoning Generation

To prevent the LLM from integrating historical noise, we extract a single, high-relevance anchor item a^* from the user’s history H_u and generate a reasoning trajectory τ . To ensure diversity and robustness, we construct two distinct types of reasoning paths. We utilize Ground Truth (GT) differently for each type to minimize post-hoc justification and overextended reasoning.

3.1.1 LLM Reasoning (Without GT)

As illustrated in Figure 3(A), our offline curation process begins by identifying the optimal semantic anchor. To prevent data leakage, the GT item i^+ is strictly masked from the input prompt. We apply a systematic *prefix-enforcement* strategy to extract the most relevant anchor from the user’s interactions. Specifically, we select the N most recent items from the history H_u as anchor candidates. For each candidate c_k , we explicitly force the LLM to begin its generated response with c_k . This hard constraint compels the model to construct a reasoning path conditionally grounded on that specific item. We then evaluate these candidate trajectories using a pretrained dense retriever (e.g., BGE (Xiao et al., 2023)). The historical item that yields the best (i.e., lowest) retrieval rank for the hidden target i^+ is formally designated as the optimal semantic anchor, a_{LLM} .

Having established a_{LLM} , we execute a Best-of- N sampling strategy to construct the final reasoning path. The frozen LLM generates a set of N diverse candidate trajectories, denoted as $\mathcal{T}(a_{LLM})$, strictly anchored on a_{LLM} . Finally, we evaluate these paths and select the trajectory that achieves the best retrieval accuracy for the target item:

$$\tau_{LLM} = \underset{\tau \in \mathcal{T}(a_{LLM})}{\operatorname{argmin}} \operatorname{Rank}(\tau \rightarrow i^+), \quad (1)$$

where $\operatorname{Rank}(\tau \rightarrow i^+)$ represents the retrieval rank of the target item i^+ when using the generated trajectory τ as the search query.

By unifying the generation and selection process into Eq. (1), this target-masked stage captures the LLM’s pure semantic deduction. Crucially, this entire selection mechanism operates

only during offline supervision curation, ensuring that the resulting trajectory τ_{LLM} maximizes empirical reachability before being used to fine-tune the model.

3.1.2 CF Reasoning (With GT Guidance)

Since autonomous semantic reasoning often fails to capture non-intuitive, global consumption patterns, we introduce a CF fallback. We employ pretrained SASRec embeddings to identify the CF anchor. Let $\mathbf{e}_i^{CF}$ denote the SASRec representation of item i . The CF anchor a_{CF} is selected by measuring the embedding similarity against the target item:

$$a_{CF} = \arg \max_{j \in H_u} \langle \mathbf{e}_j^{CF}, \mathbf{e}_{i^+}^{CF} \rangle. \quad (2)$$

Because the CF anchor represents an implicit statistical signal that the LLM cannot natively comprehend, we explicitly provide both the CF anchor a_{CF} and the target item i^+ (GT) in the prompt. We then force the LLM to articulate the relation between them, effectively translating CF knowledge into a structured narrative:

$$\tau_{CF} = \operatorname{LLM}_{gen}(\mathcal{P}(H_u, a_{CF}, i^+)). \quad (3)$$

3.2 Reachability-Aware Trajectory Routing

A critical flaw in existing generation pipelines is the inclusion of “unreachable” reasoning paths that sound plausible but fail to retrieve the actual target item from the massive candidate pool. To curate a highly effective SFT dataset, we route the generated trajectories based on their actual retrieval performance. Figure 3(B) illustrates our reachability-aware routing process.

We employ a pretrained dense retriever to convert the generated reasoning τ into a query embedding $\mathbf{q} \in \mathbb{R}^d$. If the target item i^+ falls within the Top- K retrieved items using the LLM’s autonomous query embedding $\mathbf{q}_{LLM}$, we prioritize it, preserving the model’s existing capabilities. However, if $\mathbf{q}_{LLM}$ fails to reach the Top- K target, we discard it and fall back to the CF-guided reasoning τ_{CF} to inject the necessary collaborative guidance.

Crucially, we impose a strict collaborative prerequisite before adopting the CF reasoning. We verify whether the target item i^+ falls within the Top- M retrieved items using the CF anchor’s native item embedding ($\mathbf{e}_a^{CF}$). Forcing the LLM to generate a GT-guided reasoning path based on an

anchor with no latent correlation induces excessive post-hoc rationalization, resulting in forced, illogical, and hallucinated causal links. Therefore, the CF reasoning is exclusively utilized if the anchor inherently demonstrates sufficient collaborative relevance to the target.

$$\tau_{final} = \begin{cases} \tau_{LLM}, & \text{if } i^+ \in \text{Top-}K(\mathbf{q}_{LLM}) \\ \tau_{CF}, & \text{else if } i^+ \in \text{Top-}M(\mathbf{e}_a^{CF}) \\ \text{Discard}, & \text{otherwise} \end{cases} \quad (4)$$

Ultimately, we filter out any instances where both reasoning mechanisms fail these strict reachability constraints, retaining the resulting τ_{final} to construct a high-quality dataset $\mathcal{D}_{SFT}$ for alignment.

3.3 Alignment via Curated Trajectories

Using $\mathcal{D}_{SFT}$, we optimize the model to ensure stable generative reasoning while mapping the resulting semantic trajectories to the collaborative latent space for practical item retrieval.

3.3.1 Smooth Optimization via DFT

The final phase aligns the recommendation model using the curated dataset $\mathcal{D}_{SFT}$. While teacher-forcing-based SFT is commonly used, recent theoretical analyses reveal that its gradients implicitly encode a skewed reward structure. This severely restricts the model’s generalization capabilities compared to reinforcement learning and often causes catastrophic forgetting of pre-trained knowledge.

To overcome this without the computational overhead of offline RL, we adopt Dynamic Fine-Tuning (DFT)(Wu et al., 2025). DFT stabilizes gradient updates by dynamically reweighting the objective function at the token level using the inverse of the policy probability. Let x denote the input context (prompt), and let $y^* = \tau_{final}$ denote the target reasoning sequence. For brevity, let $p_t = \pi_\theta(y_t^* | y_{<t}^*, x)$ denote the conditional probability of generating the t -th token. The DFT loss is then formulated as follows:

$$\mathcal{L}_{DFT}(\theta) = \mathbb{E}_{(x, y^*) \sim \mathcal{D}_{SFT}} \left[- \sum_{t=1}^{|y^*|} \text{sg} \left(\frac{1}{p_t} \right) \log p_t \right] \quad (5)$$

where $\text{sg}(\cdot)$ denotes the stop-gradient operator. This token-level importance sampling effectively neutralizes the problematic reward structure of standard SFT, acting as a corrective inverse ratio. It allows the LLM to smoothly adapt its reasoning

pathways to the recommendation domain, preserving its core semantic intelligence while achieving generalization.

3.3.2 Semantic-to-Collaborative Alignment

To ensure that the generated reasoning directly translates into actionable recommendations within a massive item pool, we introduce an independent retrieval alignment phase. This phase explicitly projects the generated textual representations from the LLM’s semantic space into the collaborative recommendation space.

The generated reasoning τ_{final} is passed through a pretrained dense retriever to extract its semantic representation. To bridge the modality gap and align these semantics with the recommendation domain, we map both the textual reasoning and the raw item features into a shared collaborative latent space as follows:

$$\begin{aligned} \mathbf{q}_u &= \mathbf{W} \cdot \text{Encoder}(\tau_{final}), \\ \mathbf{e}_i &= \mathbf{W} \cdot \mathbf{h}_i, \end{aligned} \quad (6)$$

where $\mathbf{W}$ is a shared learnable projection layer for both the user query and the item representations, and $\mathbf{h}_i$ represents the pre-extracted item embeddings from the same pretrained retriever.

To align the projected user query with the item representations, we independently optimize the projection layer using a contrastive InfoNCE loss. Denoting the target item as i^+ and the negative samples as j^- , the objective is formulated as:

$$s(u, i) = \exp(\langle \mathbf{q}_u, \mathbf{e}_i \rangle / T) \quad (7)$$

$$\mathcal{L}_{NCE} = -\log \frac{s(u, i^+)}{s(u, i^+) + \sum_{j=1}^B s(u, j^-)} \quad (8)$$

where T is the temperature hyperparameter and B is the number of negative samples.

3.4 Inference Pipeline

During the inference phase, GEARec operates as a streamlined, autonomous retrieval system. Given a user’s interaction history H_u , the process follows a single-pass execution: First, the fine-tuned LLM receives the history and autonomously identifies the anchor item and generates the trajectory τ . Unlike the offline data curation phase, this process requires no external GT-based guidance, reachability routing, or CF fallback. Crucially, the GT item is strictly excluded from the pipeline, completely eliminating any risk of label leakage at test time. Second, the generated text τ is transformed into a

Table 1: Performance comparison of existing methods. The best results for each metric are highlighted in bold, and the second-best results are underlined. "H@k" and "N@k" denote Hit Rate and NDCG at rank k, respectively.

Category	Model	Sports		Beauty		Toys		Yelp	
		H@10	N@10	H@10	N@10	H@10	N@10	H@10	N@10
CRM	GRU4Rec	0.0204	0.0110	0.0283	0.0137	0.0176	0.0084	0.0263	0.0134
	SASRec	0.0350	0.0192	0.0605	0.0318	0.0675	0.0374	0.0390	0.0194
	BERT4Rec	0.0191	0.0099	0.0347	0.0170	0.0203	0.0099	0.0090	0.0045
	FDSA	0.0288	0.0156	0.0407	0.0208	0.0381	0.0189	0.0544	0.0293
	S ³ -Rec	0.0385	0.0204	0.0647	0.0327	0.0700	0.0376	0.0341	0.0168
LLM EM	AlphaRec (MLP)	0.0268	0.0135	0.0456	0.0238	0.0407	0.0221	0.0073	0.0034
	AlphaRec (LGCN)	0.0328	0.0177	0.0432	0.0242	0.0179	0.0091	0.0070	0.0031
	LEARN	0.0188	0.0098	0.0286	0.0136	0.0349	0.0181	0.0096	0.0042
	SLMRec	0.0425	0.0209	0.0757	0.0390	0.0759	0.0399	0.0572	0.0354
	DLLM2Rec	0.0292	0.0143	0.0496	0.0242	0.0589	0.0315	0.0224	0.0112
	LLMEmb	0.0389	0.0205	0.0754	<u>0.0398</u>	0.0838	0.0458	0.0205	0.0103
	LLM-SRec	<u>0.0429</u>	0.0171	<u>0.0767</u>	0.0358	0.0682	0.0327	<u>0.0588</u>	0.0267
LLM GM	BIGRec (SFT)	0.0110	0.0059	0.0280	0.0151	0.0425	0.0241	0.0146	0.0100
	SLIM (Distillation)	0.0403	<u>0.0210</u>	0.0664	0.0354	0.0758	0.0408	0.0558	<u>0.0437</u>
	Rec-R1 (RL)	0.0171	0.0089	0.0523	0.0291	0.0702	0.0389	0.0185	0.0103
	RecZero (RL)	0.0283	0.0154	0.0585	0.0337	<u>0.0842</u>	<u>0.0478</u>	0.0156	0.0089
Ours	 GEARec	0.0450	0.0247	0.0833	0.0487	0.0909	0.0520	0.0755	0.0613
Improvement		+4.90%	+17.62%	+8.60%	+22.36%	+7.96%	+8.79%	+28.40%	+40.27%

dense vector via the same encoder. Finally, this representation is mapped through the projection layer $\mathbf{W}$ to yield the user query embedding $\mathbf{q}_u$, which is used to perform similarity search against the item embeddings.

4 Experiments

We evaluate whether reachable supervision, our core criterion, effectively resolves context-utilization bias while remaining practical. The research questions follow:

RQ1 Overall Performance: Does our framework improve recommendation accuracy across diverse domains?

RQ2 Design Validation & Ablation: How does each mechanism (anchor extraction, reachability routing) contribute to the framework, and do these ablations validate our core design choices?

RQ3 In-depth Recommendation Quality: Does the framework enhance recommendation quality in the existing method (e.g., long-tail items, explainability)?

4.1 Experimental Settings

Datasets & Evaluation Metrics We evaluate our framework on four diverse datasets (*Sports*, *Beauty*, *Toys*, and *Yelp*) using Hit Rate (HR@ K) and NDCG@ K . Crucially, to mirror real-world challenges, all evaluations are strictly conducted against the entire item pool (exceeding 10K items) rather than sampled subsets.

Baselines We select a comprehensive set of baselines categorized into three paradigms:

(1) Conventional Recommendation Models (CRM): Representative sequential models relying on collaborative signals, including GRU4Rec (Hidasi, 2015), BERT4Rec (Sun et al., 2019), SASRec (Kang and McAuley, 2018), FDSA (Zhang et al., 2019), and S³-Rec (Zhou et al., 2020).

(2) LLM Embedding-based Methods (LLM-EM): Recent approaches utilizing LLM-extracted features for recommendation, such as AlphaRec (Sheng et al., 2025), LEARN (Jia et al., 2025), SLMRec (Xu et al., 2025b), DLLM2Rec (Cui et al., 2024), LLMEmb (Liu et al., 2025), and LLM-SRec (Kim et al., 2025).

(3) LLM Generation-based Methods (LLM-GM): State-of-the-art generative models including BIGRec (Bao et al., 2025), SLIM (Wang et al., 2024), Rec-R1 (Lin et al., 2025a), and RecZero (Kong et al., 2026).

Implementation Details Our framework was implemented using the *Gemma-3* family (Team et al., 2025). Specifically, we employed *gemma-3-1b-it* for generating user reasoning trajectories and comparative generative baselines. The larger *gemma-3-4b-it* was utilized for item description enrichment and embedding-based baselines. For text encoding, we used *mxbai-embed-large-v1* (Li and Li, 2024) as the primary dense encoder. Source code is available at <https://github.com/venzino-han/GEARec>.

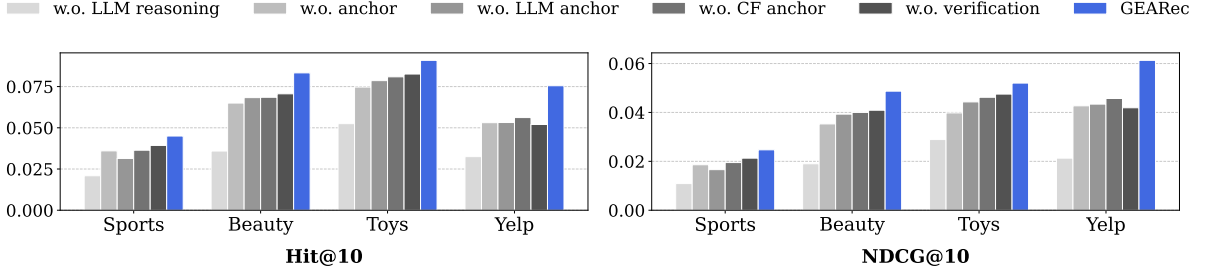


Figure 4: Ablation Study.

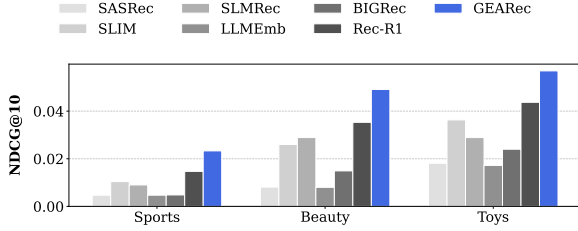


Figure 5: Evaluation on long-tail items.

4.2 Overall Performance (RQ1)

We compare GEARRec against state-of-the-art baselines under full-pool evaluation. Table 1 reports the results. GEARRec consistently and significantly outperforms all strong baselines, achieving an average improvement of 17% and up to 40% over the best-performing baseline. In particular, GEARRec robustly surpasses both SFT-based and RL-based approaches. RL-based baselines severely suffer from reward sparsity in massive item pools, leading to suboptimal retrieval. Conversely, naive SFT methods that simply target raw metadata fail to leverage the LLM’s inherent reasoning capabilities, leading to diminished recommendation performance. By grounding the reasoning process on a high-relevance anchor and integrating the semantic trajectory with the CF space, our framework successfully curates high-quality training data that translates directly into superior recommendation accuracy.

4.3 Ablation Study (RQ2)

To investigate the contribution of individual components within the GEARRec framework, we conducted a comprehensive ablation study. As shown in Figure 4, the performance drops across different variants demonstrate the necessity of each module.

(1) w.o. LLM Reasoning. We evaluated the impact of reasoning by training the model to directly target raw item metadata (e.g., titles, brands, and

categories) instead of generated reasoning paths. This variant exhibited the most significant performance drop. These results confirm that simply targeting metadata without reasoning fails to leverage the semantic intelligence required for complex user preference modeling.

(2) w.o. anchor. In this setting, we generated reasoning paths directly from the GT without designating a specific anchor item. Without an explicit anchor, the model tends to fall into *post-hoc rationalization* or hallucination, where it forcefully links irrelevant history to the target, leading to poor generalization. This decline suggests that GT access alone does not explain the gains, and that anchor-grounded curation is essential.

(3) w.o. LLM anchor. We removed the semantic anchor (a_{LLM}) to rely solely on the collaborative perspective through CF-based reasoning. The resulting performance decline indicates that while collaborative signals capture global trends, they often miss the nuanced semantic connections and user intents that only an LLM can identify.

(4) w.o. CF anchor. Conversely, we relied exclusively on the LLM’s semantic anchor and omitted the CF fallback mechanism. This variant proves that text-based semantic extraction alone is insufficient to capture non-intuitive or global mass consumption patterns. These results emphasize that the synergy between semantic and collaborative anchors is essential for robust recommendation.

(5) w.o. verification. Finally, we evaluated the model without the rank-based verification process by training it on all generated trajectories regardless of their retrieval success. The performance degradation suggests that filtering out unreachable reasoning paths is important for maintaining reasoning trajectories that are better aligned with user behavior and retrieval outcomes. This is particularly important because trajectories that sound plausible but fail to achieve actual item retrieval

Table 2: LLM-as-Judge evaluation on Beauty.

Judge	Opponent	Win	Tie	Lose	Win % (excl. tie)
gemin-3.1-flash-lite	Rec-R1	76	19	5	93.8 %
gemin-3.1-flash-lite	GT-Based	35	41	24	59.3 %
gpt-5.4-mini	Rec-R1	77	19	4	95.1 %
gpt-5.4-mini	GT-Based	32	42	26	55.2 %

Table 3: Backbone robustness (NDCG@10).

Configuration	Sports	Beauty	Toys	Yelp
Best baseline	0.0210	0.0398	0.0478	0.0437
<i>Same-LLM control (LLMEmb on Qwen3.5-0.8B)</i>				
LLMEmb (Qwen3.5-0.8B)	0.0081	0.0122	0.0198	0.0020
<i>GEARec: CF backbone swap</i>				
GEARec (SASRec, Gemma-3-1B, main)	0.0247	0.0487	0.0520	0.0613
GEARec (S ³ -Rec, Gemma-3-1B)	0.0220	0.0450	0.0513	0.0584
<i>GEARec: LLM family swap</i>				
GEARec (SASRec, Qwen3.5-0.8B)	0.0246	0.0479	0.0540	0.0560

can mislead the alignment process.

4.4 Recommendation Quality (RQ3)

Long-Tail Evaluation Long-tail items, those in the bottom 80% by popularity, are the recommender’s standard failure mode: existing methods over-fit to head items and fail to capture long-tail demand. As illustrated in Figure 5, GEARec significantly improves the performance on long-tail items. By anchoring the reasoning process on a single historical item rather than indiscriminately aggregating all past interactions, the model is compelled to articulate a focused, unbiased signal.

Reasoning Quality Analysis We conduct an LLM-as-a-Judge evaluation on 100 user/item pairs from Beauty against Rec-R1 (RL baseline) and a GT-based oracle. We used the anonymized protocol with five evaluation criteria. Table 2 shows the evaluation results on 2 different judges (*gemin-3.1-flash-lite* and *gpt-5.4-mini*). GEARec wins most of the time against Rec-R1 and the GT-based oracle, confirming the qualitative dominance. This results shows that our grounded and curated reasoning is more target-relevant and less context-utilization biased than the existing methods. More detailed results and evaluation protocols are provided in the Appendix C.5 and A.9.

4.5 In-depth Analysis

In this subsection, we provide in-depth analysis of the framework’s performance.

Backbone Robustness To confirm that our performance gains stem from the reachable supervision criterion rather than specific architecture choices, we evaluate GEARec across different collaborative filtering and LLM backbones. Switching the CF architecture (from SASRec to S³-Rec)

Table 4: Computational cost on a single NVIDIA A6000 (48 GB) for Beauty dataset.

Phase	Component	Time
Curation (offline)	vLLM generation	< 15 min
Training (offline)	DFT alignment + projection	< 30 min
Inference (online)	Generation + Dense retrieval	16.5 min (44.1 ms/user)

or changing the LLM family (from Gemma3 to Qwen3.5) consistently preserves our framework’s superiority over the baselines.

Computational Cost Table 4 summarizes the end-to-end computational cost of GEARec on a single A6000 GPU. The one-time offline overhead is minimal, requiring less than 15 minutes for data curation and under 30 minutes for training. Crucially, the online inference phase achieves a latency of just 44.1 ms per user. This confirms that our framework is highly computationally efficient and fully practical for deployment.

Additional Analyses Due to space limitations, additional analyses are provided in the Appendix. These include a quantitative analysis of context-utilization bias (Appendix C.1), a comparison of our fine-tuned 1B model’s performance against larger LLMs without fine-tuning (Appendix C.2), anchor-window robustness analysis (Appendix C.3), a routing composition analysis proving the successful prevention of error propagation (Appendix C.4), and qualitative case studies illustrating our model’s robust reasoning fidelity (Appendix C.6).

5 Conclusion

In this paper, we presented GEARec, a novel reachability-aware training-data curation framework designed to resolve context-utilization bias in LLM-based recommendation. By anchoring the reasoning on a anchor item and filtering noise through reachability-aware routing, our method curates highly focused supervision that effectively bridges semantic reasoning with CF signals. Extensive full-pool evaluations demonstrate that GEARec consistently surpasses strong baselines, significantly improving both long-tail recommendation accuracy and qualitative reasoning fidelity. Crucially, our experiments reveal that a compact 1B model fine-tuned on our curated data outperforms much larger, zero-shot LLMs (e.g., up to 80B parameters), proving that reachable supervision quality dominates raw parameter scale.

6 Limitations

While GEARec demonstrates significant improvements in recommendation accuracy and reasoning fidelity, we identify a few practical limitations. First, because our framework requires LLM-based text generation during both the offline data curation and online inference phases, it inevitably incurs higher computational costs and inference latency compared to traditional, purely embedding-based recommendation models. Second, the framework’s hybrid reasoning intrinsically relies on the quality of the underlying Collaborative Filtering (CF) model; in highly sparse domains where the base CF model struggles, the statistical validity of the CF anchor may correspondingly degrade. Finally, similar to other LLM-driven recommenders, the efficacy of our semantic reasoning is heavily dependent on rich textual metadata, inherently constraining its application in environments where items are represented solely by opaque IDs without meaningful descriptions. Addressing these dependencies to scale gracefully across extreme cold-start or strictly non-semantic domains remains an important direction for future research.

7 Ethics Statement

Use of Scientific Artifacts Our research relied on open-source tools, including PyTorch (Paszke et al., 2019), and pre-trained language models such as Gemma3 distributed through the Hugging Face library (Wolf et al., 2019). For experiments involving commercial LLM services, we used the Google Gemini API in accordance with the providers policy¹.

Use of AI Assistants Google Gemini was used solely for language polishing and wording refinement, and not for producing the core research ideas or methodology. Some portions of the code used in the experiments were developed with the assistance of Cursor².

8 Acknowledgments

This research was supported by the National Research Foundation of Korea (NRF) grants funded by the Korean government (MSIT) (Grant No. RS-2022-NR068758 (50%), Grant No. RS-2025-

16071337 (50%), and Grant No. RS-2024-00334343).

References

- Zhuoxi Bai, Ning Wu, Fengyu Cai, Xinyi Zhu, and Yun Xiong. 2024. Aligning large language model with direct multi-preference optimization for recommendation. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pages 76–86.
- Keqin Bao, Jizhi Zhang, Wenjie Wang, Yang Zhang, Zhengyi Yang, Yanchen Luo, Chong Chen, Fuli Feng, and Qi Tian. 2025. A bi-step grounding paradigm for large language models in recommendation systems. *ACM Transactions on Recommender Systems*, 3(4):1–27.
- Keqin Bao, Jizhi Zhang, Yang Zhang, Wenjie Wang, Fuli Feng, and Xiangnan He. 2023. Tallrec: An effective and efficient tuning framework to align large language model with recommendation. In *Proceedings of the 17th ACM Conference on Recommender Systems*, pages 1007–1014.
- Yu Cui, Feng Liu, Pengbo Wang, Bohao Wang, Heng Tang, Yi Wan, Jun Wang, and Jiawei Chen. 2024. Distillation matters: Empowering sequential recommenders to match the performance of large language models. In *Proceedings of the 18th ACM Conference on Recommender Systems, RecSys ’24*, page 507517, New York, NY, USA. Association for Computing Machinery.
- Yi Fang, Wenjie Wang, Yang Zhang, Fengbin Zhu, Qifan Wang, Fuli Feng, and Xiangnan He. 2025. Reason4rec: Large language models for recommendation with deliberative user preference alignment. *arXiv preprint arXiv:2502.02061*.
- Shijie Geng, Shuchang Liu, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. 2022. Recommendation as language processing (rlp): A unified pretrain, personalized prompt & predict paradigm (p5). In *Proceedings of the 16th ACM Conference on Recommender Systems, RecSys ’22*, page 299315, New York, NY, USA. Association for Computing Machinery.
- Donghee Han, Daeyoung Roh, A Young Kim, Hwanjun Song, and Mun Yong Yi. 2026. Reasoning-to-encoder distillation for recommendation.
- Donghee Han, Hwanjun Song, and Mun Yong Yi. 2025. Rethinking llm-based recommendations: A query generation-based, training-free approach. *arXiv preprint arXiv:2504.11889*.
- B Hidasi. 2015. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939*.

¹<https://ai.google.dev/>

²<https://cursor.com/>

- Yupeng Hou, Junjie Zhang, Zihan Lin, Hongyu Lu, Ruobing Xie, Julian McAuley, and Wayne Xin Zhao. 2024. Large language models are zero-shot rankers for recommender systems. In *European Conference on Information Retrieval*, pages 364–381. Springer.
- Jian Jia, Yipei Wang, Yan Li, Honggang Chen, Xuehan Bai, Zhaocheng Liu, Jian Liang, Quan Chen, Han Li, Peng Jiang, and Kun Gai. 2025. [Learn: knowledge adaptation from large language model to recommendation for practical industrial application](#). In *Proceedings of the Thirty-Ninth AAAI Conference on Artificial Intelligence and Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence and Fifteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI’25/IAAI’25/EAAI’25. AAAI Press.
- Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *2018 IEEE international conference on data mining (ICDM)*, pages 197–206. IEEE.
- Sein Kim, Hongseok Kang, Kibum Kim, Jiwan Kim, Donghyun Kim, Minchul Yang, Kwangjin Oh, Julian McAuley, and Chanyoung Park. 2025. [Lost in sequence: Do large language models understand sequential recommendation?](#) *Preprint*, arXiv:2502.13909.
- Xiaoyu Kong, Jinguang Jiang, Bin Liu, Ziru Xu, Han Zhu, Jian Xu, Bo Zheng, Jiancan Wu, and Xiang Wang. 2026. Think before recommendation: Autonomous reasoning-enhanced recommender. *Advances in Neural Information Processing Systems*, 38:141209–141232.
- Sunkyoung Lee, Minjin Choi, Eunseong Choi, Hye-young Kim, and Jongwuk Lee. 2025. Gram: Generative recommendation via semantic-aware multi-granular late fusion. *arXiv preprint arXiv:2506.01673*.
- Lei Li, Yongfeng Zhang, and Li Chen. 2023. Prompt distillation for efficient llm-based recommendation. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pages 1348–1357.
- Xianming Li and Jing Li. 2024. [AoE: Angle-optimized embeddings for semantic textual similarity](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1825–1839, Bangkok, Thailand. Association for Computational Linguistics.
- Jiacheng Lin, Tian Wang, and Kun Qian. 2025a. Rec-r1: Bridging generative large language models and user-centric recommendation systems via reinforcement learning. *arXiv preprint arXiv:2503.24289*.
- Jianghao Lin, Xinyi Dai, Yunjia Xi, Weiwen Liu, Bo Chen, Hao Zhang, Yong Liu, Chuhan Wu, Xiangyang Li, Chenxu Zhu, Hui Feng Guo, Yong Yu, Ruiming Tang, and Weinan Zhang. 2025b. [How can recommender systems benefit from large language models: A survey](#). *ACM Trans. Inf. Syst.*, 43(2).
- Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the middle: How language models use long contexts. *Transactions of the association for computational linguistics*, 12:157–173.
- Qidong Liu, Xian Wu, Wanyu Wang, Yejing Wang, Yuanshao Zhu, Xiangyu Zhao, Feng Tian, and Yefeng Zheng. 2025. Llmemb: Large language model can be a good embedding generator for sequential recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 12183–12191.
- Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. [Pytorch: An imperative style, high-performance deep learning library](#). *CoRR*, abs/1912.01703.
- Shashank Rajput, Nikhil Mehta, Anima Singh, Raghunandan Hulikal Keshavan, Trung Vu, Lukasz Heldt, Lichan Hong, Yi Tay, Vinh Tran, Jonah Samost, et al. 2023. Recommender systems with generative retrieval. *Advances in Neural Information Processing Systems*, 36:10299–10315.
- Mathieu Ravaut, Aixin Sun, Nancy Chen, and Shafiq Joty. 2024. On context utilization in summarization with large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2764–2781.
- Leheng Sheng, An Zhang, Yi Zhang, Yuxin Chen, Xiang Wang, and Tat-Seng Chua. 2025. Language representations can be what recommenders need: Findings and potentials. In *ICLR*.
- Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H Chi, Nathanael Schärli, and Denny Zhou. 2023. Large language models can be easily distracted by irrelevant context. In *International Conference on Machine Learning*, pages 31210–31227. PMLR.
- Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pages 1441–1450.
- Zhongxiang Sun, Zihua Si, Xiaoxue Zang, Kai Zheng, Yang Song, Xiao Zhang, and Jun Xu. 2024. [Large language models enhanced collaborative filtering](#). In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, CIKM ’24, page 21782188, New York, NY, USA. Association for Computing Machinery.

- Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, and Alexandre Ramé et al. 2025. [Gemma 3 technical report](#). Preprint, arXiv:2503.19786.
- Qingyuan Tian, Hanlun Zhu, Lei Wang, Yang Li, and Yunshi Lan. 2023. R3 prompting: Review, rephrase and resolve for chain-of-thought reasoning in large language models under noisy context. *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 1670–1685.
- Alicia Tsai, Adam Kraft, Long Jin, Chenwei Cai, Anahita Hosseini, Taibai Xu, Zemin Zhang, Lichan Hong, Ed H Chi, and Xinyang Yi. 2024. Leveraging llm reasoning enhances personalized recommender systems. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 13176–13188.
- Miles Turpin, Julian Michael, Ethan Perez, and Samuel Bowman. 2023. Language models don’t always say what they think: Unfaithful explanations in chain-of-thought prompting. *Advances in Neural Information Processing Systems*, 36:74952–74965.
- Kremena Valkanova and Pencho Yordanov. 2024. Irrelevant alternatives bias large language model hiring decisions. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 6899–6912.
- Shijie Wang, Wenqi Fan, Yue Feng, Lin Shanru, Xinyu Ma, Shuaiqiang Wang, and Dawei Yin. 2025. Knowledge graph retrieval-augmented generation for llm-based recommendation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 27152–27168.
- Yuling Wang, Changxin Tian, Binbin Hu, Yanhua Yu, Ziqi Liu, Zhiqiang Zhang, Jun Zhou, Liang Pang, and Xiao Wang. 2024. [Can small language models be good reasoners for sequential recommendation?](#) In *Proceedings of the ACM Web Conference 2024*, WWW ’24, page 38763887, New York, NY, USA. Association for Computing Machinery.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. [Huggingface’s transformers: State-of-the-art natural language processing](#). *CoRR*, abs/1910.03771.
- Yongliang Wu, Yizhou Zhou, Zhou Ziheng, Yingzhe Peng, Xinyu Ye, Xinting Hu, Wenbo Zhu, Lu Qi, Ming-Hsuan Yang, and Xu Yang. 2025. On the generalization of sft: A reinforcement learning perspective with reward rectification. *arXiv preprint arXiv:2508.05629*.
- Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. 2023. [C-pack: Packaged resources to advance general chinese embedding](#). Preprint, arXiv:2309.07597.
- Haozhe Xu, Xiaohua Wang, Changze Lv, and Xiaoqing Zheng. 2025a. Beyond single labels: Improving conversational recommendation through llm-powered data augmentation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15573–15590.
- Wujiang Xu, Qitian Wu, Zujie Liang, Jiaojiao Han, Xuying Ning, Yunxiao Shi, Wenfang Lin, and Yongfeng Zhang. 2025b. Slmrec: Distilling large language models into small for sequential recommendation. In *ICLR*.
- Jingfeng Yang, Hongye Jin, Ruixiang Tang, Xiaotian Han, Qizhang Feng, Haoming Jiang, Shaochen Zhong, Bing Yin, and Xia Hu. 2024. Harnessing the power of llms in practice: A survey on chatgpt and beyond. *ACM Transactions on Knowledge Discovery from Data*, 18(6):1–32.
- Ruihan Yang, Yikai Zhang, Aili Chen, Xintao Wang, Siyu Yuan, Jiangjie Chen, Deqing Yang, and Yanghua Xiao. 2025. Aria: Training language agents with intention-driven reward aggregation. *arXiv preprint arXiv:2506.00539*.
- Tingting Zhang, Pengpeng Zhao, Yanchi Liu, Victor S Sheng, Jiajie Xu, Deqing Wang, Guanfeng Liu, Xiaofang Zhou, et al. 2019. Feature-level deeper self-attention network for sequential recommendation. In *IJCAI*, pages 4320–4326.
- Keyu Zhao, Fengli Xu, and Yong Li. 2025. Reason-to-recommend: Using interaction-of-thought reasoning to enhance llm recommendation. *arXiv preprint arXiv:2506.05069*.
- Kun Zhou, Hui Wang, Wayne Xin Zhao, Yutao Zhu, Sirui Wang, Fuzheng Zhang, Zhongyuan Wang, and Ji-Rong Wen. 2020. S3-rec: Self-supervised learning for sequential recommendation with mutual information maximization. In *Proceedings of the 29th ACM international conference on information & knowledge management*, pages 1893–1902.

A Experiment Details

A.1 Preliminary Study and Empirical Motivation

In this section, we provide the empirical evidence that motivated the development of GEARec, focusing on the limitations of existing evaluation setups and the presence of *Context-Utilization Bias* in LLM-based recommendations.

A.1.1 Gap Between Rating Prediction and Sequential Recommendation

Existing studies on LLM-based alignment (e.g., SFT data sampling, RL) often rely on rating prediction tasks (Tsai et al., 2024; Kong et al., 2026; Fang et al., 2025). In such setups, the action space is restricted to a narrow range (e.g., 1 to 5), and the target item is explicitly provided in the prompt, which artificially eliminates the reward sparsity problem. However, we argue that this setup is disconnected from real-world recommendation scenarios where the model must identify a target from thousands of candidates. To verify this, we compared the trajectory success rate between a point-wise rating task and a retrieval-based sequential recommendation task. As shown in Figure 6, while the rating task consistently yields high optimization signals, the sequential recommendation task suffers from a catastrophic drop in successful reasoning paths. This confirms that autonomous exploration in a massive item pool is inherently difficult without a grounded guidance mechanism.

A.1.2 Impact of Context-Utilization Bias

LLMs exhibit a strong *Context-Utilization Bias*, where they tend to over-interpret the entire provided history rather than identifying the most salient signals. To investigate how this bias limits the diversity of recommendation reasoning, we conducted a “Best-of- N ” ($N = 8$) experiment under two different settings: **(1) Unconstrained Setting:** The LLM is allowed to autonomously select an anchor item from the history to generate 8 different reasoning trajectories. **(2) Constrained Setting (Anchor-Forced):** We explicitly force the LLM to use 8 distinct items from the user’s recent history as anchors for each trajectory.

Our results reveal that the Constrained Setting outperforms the Unconstrained Setting by nearly two times in terms of hit rate. In the Unconstrained Setting, the LLM attempts to holistically

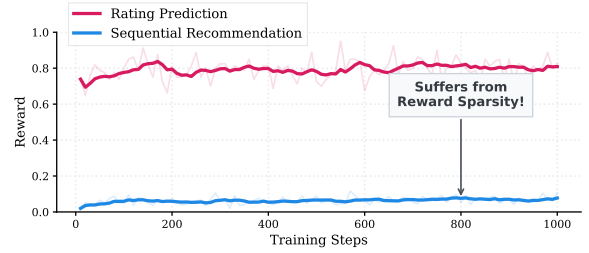


Figure 6: Comparison of RL training rewards between rating prediction and sequential recommendation tasks.

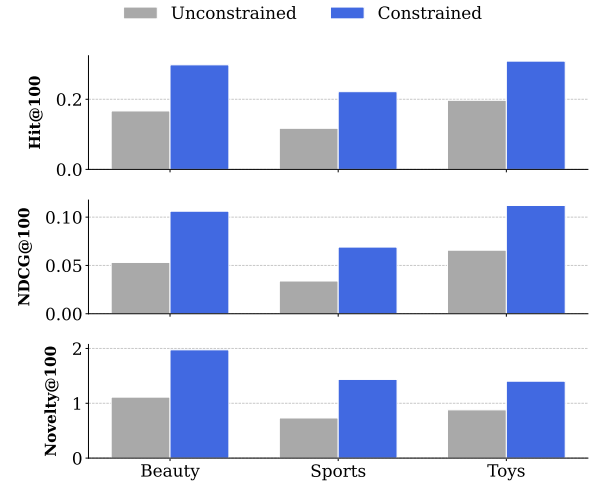


Figure 7: Performance comparison of Best-of- N reasoning generation under Unconstrained and Constrained (Anchor-Forced) settings.

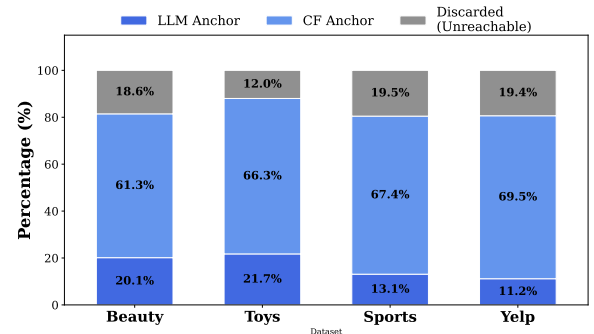


Figure 8: Statistics of reachability-aware trajectory routing. The figure illustrates the discard rates and the final proportion of LLM and CF-guided trajectories across four benchmark datasets.

connect all past interactions, which hinders its ability to explore differentiated trajectories and ultimately causing exploration collapse. This empirical finding demonstrates that Context-Utilization Bias acts as a significant bottleneck, and providing explicit diversity via anchor-guided routing is essential for effective data curation.

A.1.3 Trajectory Routing Statistics

In our implementation, we set the number of candidate trajectories to $N = 8$. The reachability

thresholds for the LLM and CF-guided trajectories were set to $K = 20$ and $M = 100$, respectively. As a result of our reachability-aware routing process, we observed that a significant portion of initially generated trajectories were filtered out, as illustrated in Figure 8. Across the four benchmark datasets, the discard rate was approximately 17.35% (ranging from 11.98% in *Toys* to 19.51% in *Sports*). Among the retained trajectories, the semantic (target-masked) view supplies the majority of supervision: 81.3% on Beauty, 69.5% on Yelp, 67.4% on Sports, and 68.3% on Toys, with the remainder coming from the gated CF fallback. The semantic view is therefore the default supervision source, not the CF fallback, and the routing step actively removes the unreachable fraction. These discarded instances represent “unreachable” reasoning paths that could not successfully retrieve the target item within the Top- K candidates, ensuring that our final SFT dataset $\mathcal{D}_{SFT}$ remains highly grounded.

A.2 Datasets

Our experiments are conducted on three subsets of the Amazon review dataset: *Sports*, *Beauty*, and *Toys*. Each record in these datasets comprises a user-item transaction, including user and item identifiers, numeric ratings, textual reviews, and chronological timestamps. Detailed statistics for each domain are summarized in Table 5. These datasets are recognized as standard benchmarks in the field and have been frequently utilized in recent recommendation studies (Geng et al., 2022; Rajput et al., 2023; Lee et al., 2025; Sun et al., 2024) to validate model performance over the past few years.

To evaluate our model, we employ the *leave-one-out* protocol, which is a widely established evaluation strategy in sequential recommendation research. Following the setup used in numerous prior works (Kang and McAuley, 2018; Sun et al., 2019; Zhou et al., 2020; Li et al., 2023; Lee et al., 2025; Sun et al., 2024), we use the most recent interaction for testing, the second most recent for validation, and the remainder for training. This consistent experimental setup ensures a fair and direct comparison between GEARec and the baseline methods.

Table 5: Statistics of the datasets.

Dataset	#Users	#Items	#Reviews	Density (%)
Sports	35,598	18,357	296,337	0.0453
Beauty	22,363	12,101	198,502	0.0734
Toys	19,412	11,924	167,597	0.0724
Yelp	30,431	20,033	316,354	0.0519

A.3 Baselines

To rigorously assess the performance of GEARec, we compared it against a comprehensive set of baselines, which can be categorized into three distinct paradigms.

A.3.1 Conventional Recommendation Methods

This category includes standard sequential recommendation architectures that focus on modeling user interaction patterns:

- **GRU4Rec** (Hidasi, 2015): A recurrent neural network-based model that utilizes Gated Recurrent Units to capture sequential dependencies with a pairwise ranking objective.
- **SASRec** (Kang and McAuley, 2018): A self-attention-based framework designed to model long-term user behavior using unidirectional Transformer blocks.
- **BERT4Rec** (Sun et al., 2019): Employs bidirectional self-attention to learn deep sequential representations for item prediction.
- **FDSA** (Zhang et al., 2019): Integrates feature-level and item-level self-attention to incorporate heterogeneous item attributes into the recommendation process.
- **S³-Rec** (Zhou et al., 2020): Leverages self-supervised learning to capture intrinsic correlations between items and their associated metadata.

A.3.2 LLM Embedding-based Recommendation Methods

These methods primarily utilize LLMs as advanced feature extractors to generate semantically rich item or user representations:

- **AlphaRec** (Sheng et al., 2025): Challenges traditional ID-based embedding paradigms by demonstrating that LLM latent spaces implicitly encode collaborative signals. We evaluated both the MLP and LightGCN variants for a robust comparison.

- **LEARN** (Jia et al., 2025): Synergizes open-world knowledge from pre-trained LLMs with collaborative filtering using a twin-tower architecture, ensuring semantic alignment without catastrophic forgetting.
- **SLMRec** (Xu et al., 2025b): Focuses on inference efficiency by distilling knowledge from intermediate layers of a large teacher model into a compact student model, integrated with SASRec-based embeddings.
- **DLLM2Rec** (Cui et al., 2024): A distillation strategy that transfers LLM-based ranking knowledge to lightweight models using an importance-aware mechanism to bridge the capacity gap.
- **LLMEmb** (Liu et al., 2025): Utilizes contrastive fine-tuning to align LLM-generated semantic embeddings with collaborative signals, specifically targeting the long-tail item problem.
- **LLM-SRec** (Kim et al., 2025): Integrates semantic priors from LLMs with collaborative filtering signals by inputting pre-trained sequential representations directly into the language model.

A.3.3 LLM Generation-based Recommendation Methods

This group encompasses generative frameworks where LLMs are fine-tuned to produce recommendations or retrieval queries:

- **BIGRec** (Bao et al., 2025): Proposes a bi-step grounding paradigm where the LLM generates descriptive tokens that are subsequently mapped to the full item pool for practical retrieval.
- **SLIM** (Wang et al., 2024): Implements reasoning distillation by fine-tuning a student model on trajectories generated by a larger teacher (e.g., Gemma-3-12B in our setup).
- **Rec-R1** (Lin et al., 2025a): Introduces a reinforcement learning framework that optimizes LLM generation through closed-loop feedback from a black-box recommender, preserving general reasoning capabilities while enhancing performance.
- **RecZero** (Kong et al., 2026): A reinforcement learning framework that optimizes LLM reasoning for recommendation. The original implementation is for pairwise rating prediction, but we adapted it for sequential recommendation by treating the final recommendation text as query.

A.4 Implementation Details

We utilized the vLLM³ framework for high-efficiency text generation and inference. To extract dense textual features, we employed the SentenceTransformers⁴ library. The entire system was developed using PyTorch⁵ and executed in a Linux environment on an NVIDIA A6000 GPU with 48GB of VRAM. To ensure the reliability and reproducibility of our results, experiments were conducted across three independent random seeds (22, 42, and 62). Furthermore, we performed statistical significance tests to verify our improvements, confirming that GEARec consistently outperforms the strongest baseline with a p -value less than 0.05.

To ensure the reproducibility of our experiments, we detail the hyperparameters used during the optimization phase. The Dynamic Fine-Tuning (DFT) and the semantic-to-collaborative alignment were conducted with the following configurations, carefully chosen to balance alignment stability and computational efficiency.

Table 6: Hyperparameters for DFT.

Hyperparameter	Value
Batch Size	4
Learning Rate	1e-6
Training Epochs	1
Max Steps	3,000

Table 7: Hyperparameters for projection layer.

Hyperparameter	Value
Batch Size	128
Hidden Dimension	1024
Learning Rate	1e-4
Training Epochs	20
Number of Negative Samples	99

More detailed information, including code and hyperparameters for reproducibility verification, is

³<https://docs.vllm.ai/>

⁴<https://sbert.net/>

⁵<https://pytorch.org/>

Main Prompt Template

You are an intelligent shopping assistant that helps predict what users may want to purchase next. Below is a list of items a user has purchased recently. Analyze the user’s purchase history and infer their NEXT purchase intent driven by a specific past item. Below is the user purchase history: {user_history}

First, identify the single key item from the history that most strongly motivates the user’s next purchase. Then, generate a detailed, paragraph-length intent (3-5 sentences) describing the user’s underlying preferences. This paragraph should clearly explain WHY the key item influenced them, WHAT specific attributes or features they value, and HOW this leads to their next purchase.

Output Format:
Strictly follow this exact tag format:
<key_item>[Key Item Name]</key_item>
<query>[A detailed paragraph describing the user’s preferences and future intent]</query>

GT-Guided Key Item Selection and Generation Prompt Template

You are an intelligent shopping assistant that helps predict what users may want to purchase next. Below is a list of items a user has purchased recently. Analyze the user’s purchase history and infer their NEXT purchase intent driven by a specific past item. Below is the user purchase history: {user_history}

The user’s NEXT purchase has been identified as: “{gt_item_title}”. Generate a detailed, paragraph-length preference description (3-5 sentences) that explains:

- WHY “{gt_item_title}” is a natural continuation of the user’s purchase history,
- WHAT specific attributes or features in their history suggest this interest,
- HOW their past purchases connect to and motivate this next purchase.

Output Format:
Strictly follow this exact tag format:
<key_item>[Key Item Name]</key_item>
<query>[A detailed paragraph describing the user’s preferences and motivation for purchasing “{gt_item_title}”]</query>

publicly available in our online anonymous repository⁶.

A.5 Prompt Template

The following prompt is used to elicit the reasoning process for identifying the anchor item and generating the subsequent purchase intent. The placeholder {user_history} is dynamically replaced with the item titles from the user’s interaction history.

A.5.1 Prefix-Enforced LLM Anchor Extraction

To rigorously assess the LLM’s autonomous reasoning and enforce it to ground its deduction on a specific candidate anchor (c_k), we apply a prefix-enforcement decoding strategy. We utilize the exact same base prompt designed for key item selection (as shown in Appendix A.5), but instead of allowing free-form generation, we forcefully pre-fill the LLM’s initial output sequence with the designated candidate item.

Specifically, the generation is strictly constrained to begin with: <key_item>[c_k]</key_item>. This hard prefix ensures that the LLM must construct its

reasoning trajectory conditional on c_k without ever exposing the actual Ground Truth (GT) target item in the input prompt.

A.5.2 GT-Guided Reasoning Generation

When the semantic reachability check fails, we fall back to the Collaborative Filtering (CF) perspective. In this phase, both the user history and the target item are provided to the LLM to articulate the causal link. The placeholder {gt_item_title} is dynamically replaced with the actual target item.

A.6 Computational Cost and Efficiency

Despite the inherent computational overhead of utilizing Large Language Models for sequential recommendation, GEARec is designed to maintain high practical efficiency. A summary table of the three phases is provided in the main text (Table 4).

Data Curation Efficiency. By leveraging the vLLM framework for high-throughput text generation, the offline data generation process is highly accelerated. Constructing the reasoning trajectories for 10k user interactions requires less than 15 minutes on a single NVIDIA A6000 GPU (48GB

⁶<https://anonymous.4open.science/r/GEARec>

VRAM).

Training Efficiency. The Dynamic Fine-Tuning (DFT) process operates smoothly within a maximum of 3,000 steps. On average, the entire training phase for the DFT alignment takes less than 30 minutes on the same A6000. This demonstrates that our framework can curate high-quality data and align the model without the prohibitive computational costs typically associated with extensive RL approaches.

Inference Efficiency. At test time, GEARec executes a single LLM pass followed by dense retrieval over precomputed item embeddings (no Best-of- N , no CF fallback, no routing). We measured per-user inference latency on the entire Beauty test split (22,363 users) on one A6000 with gemma-3-1b-it served via vLLM. The average input length is 521 tokens and the average output length is 161 tokens (sampling: temperature= 0.1, max_tokens= 512, min_tokens= 64, max_num_seqs= 32). The resulting average per-user generation latency is **44.1 ms**, corresponding to a throughput of 22.7 users/s (11,807 input tokens/s and 3,645 output tokens/s); the full test split completes in 16.5 minutes wall-clock time. This places GEARec within the deployable range for real-time per-user serving.

A.7 Query Construction

To prepare the input for the dense retriever, we perform a query refinement process. We strip all structural tags (e.g., <key_item>, <query>) used during the LLM’s generation phase to ensure the input contains only pure semantic information. Subsequently, the verbalized intent is concatenated with the title of the user’s most recent interaction.

A.8 Enriched Item Index

To enhance the representational quality of item embeddings during the offline item index enrichment phase, we leverage a LLM to generate expanded descriptions that capture latent user preferences. These LLM-generated descriptions are systematically integrated with the original item metadata to form a comprehensive textual profile. Finally, this augmented textual information is processed through a pre-trained text encoder to extract enriched item embeddings, ensuring a more nuanced and semantically dense representation of each item.

Prompt for Item Description Generation

You are an intelligent assistant that analyzes item metadata and user feedback to generate personalized item recommendations.

Task:

Analyze the provided metadata and reviews to determine what kinds of users are most likely to prefer the target item.

Your response should start with:

****Users who prefer [common themes/preferences] would find this item suitable.**** Incorporate patterns from the reviews, such as favored features, usage scenarios, or functional benefits.

Target Item Metadata:

- ****Target Item Title**:** {Title}
- ****Brand**:** {Brand}
- ****Original Description**:** {Description}

Requirements:

- Begin with the sentence: “Users who prefer ...”
- Respond with a concise and information-dense single paragraph, strictly limited to {max_words} words. Avoid redundancy and ensure clarity.

Response:

A.9 LLM-as-a-Judge Evaluation Details

To rigorously assess the qualitative superiority of the generated reasoning, we employed an LLM-as-a-Judge framework. We randomly sampled 100 user-item interaction pairs and their corresponding generated reasoning trajectories from the *Beauty* dataset. For the primary evaluation, we utilized gemini-3.1-flash-lite as the blind evaluator to compare GEARec against each baseline.

The evaluator was tasked with performing a head-to-head comparison between GEARec and a baseline across four key dimensions: (1) Resistance to Context-Utilization Bias, (2) Causal Grounding, (3) Logical Plausibility, and (4) Faithfulness (Hallucination Check). To ensure objectivity, the model names were anonymized as “Model A” and “Model B” during the inference. The specific prompt template used for this evaluation is provided.

Cross-vendor judge replication. To verify that the qualitative dominance reported in the main text is not specific to the Gemini family, we replicated the entire protocol with a different-vendor judge, gpt-5.4-mini. The sample (Beauty, $n = 100$),

LLM-as-a-Judge: Evaluation Prompt Template

You are an expert evaluator for Recommender Systems and Large Language Models. Your task is to objectively compare two “Reasoning Trajectories” (Model A and Model B) generated by LLMs.

Both models were tasked with explaining WHY a user would interact with the [Target Item] based on their [Historical Interactions].

Real-world user history is extremely noisy. A high-quality reasoning must NOT fall into “Context-Utilization Bias” (the flaw of forcing unnatural logical connections using every single item in the history). Instead, it must extract the true causal anchor, build a logically sound path, and remain strictly faithful to the actual facts without hallucination.

[Input Data]

```
- User      Historical      Interactions:
{User_History}
- Target Item (Ground Truth): {Target_Item}
- Reasoning by Model A: {Reasoning_A}
- Reasoning by Model B: {Reasoning_B}
```

[Evaluation Criteria]

1. Resistance to Context-Utilization Bias: Which model better ignores irrelevant historical noise?
2. Causal Grounding: Which model more clearly identifies a specific “Key Item” (Anchor) as the causal root?
3. Logical Plausibility: Which explanation is more convincing and naturally sound?
4. Faithfulness: Which model is more strictly faithful to the actual item attributes without hallucination?

[Output Format]

Output your evaluation in the following JSON format strictly.

```
{
  "Evaluation": {
    "Bias_Resistance_Winner":
      "Model A" | "Model B" | "Tie",
    "Causal_Grounding_Winner":
      "Model A" | "Model B" | "Tie",
    "Logical_Plausibility_Winner":
      "Model A" | "Model B" | "Tie",
    "Faithfulness_Winner":
      "Model A" | "Model B" | "Tie",
    "Overall_Winner":
      "Model A" | "Model B" | "Tie",
    "Justification":
      "<A concise 3-4 sentence explanation.>"
  }
}
```

prompt template, anonymization scheme, and pair-order randomization are identical to the primary run; only the evaluator model changes. The resulting win-rates are summarized in Table 2 in the main text: GEARec wins 95.1% (excl. ties) against the Rec-R1 RL baseline and 55.2% against the GT-Based oracle constructed with target access. Because the reachability criterion is defined on the dense retriever and not on any language model’s preference, this judge-agnostic outcome is consistent with the framework’s design.

B Discussion on Practical Applicability and Core Contributions

While previous studies have explored the integration of LLMs in recommendation systems, many of these approaches predominantly focus on constrained tasks such as rating prediction or reranking small candidate set (Bao et al., 2023; Bai et al., 2024; Tsai et al., 2024; Kong et al., 2026; Zhao et al., 2025). In these setups, the target candidate item is explicitly provided to the LLM within the prompt. We argue that this formulation artificially simplifies the problem, effectively bypassing the fundamental difficulties of recommendation. Providing a single candidate or a heavily narrowed-down list acts as an informational shortcut, allowing the model to merely evaluate a given user-item pair rather than actively discovering the target. Another important issue is that the model may propagate errors from data generation to the training process. Our reachability-aware routing process is designed to prevent this error propagation by filtering out unreachable trajectories. Consequently, solutions optimized for these restricted environments are often impractical for deployment in real-world systems.

The the primary challenge of real-world sequential recommendation lies in two critical aspects:

1. **Find intent from noisy histories:** Real users exhibit diverse and often disjointed purchase behaviors. Identifying the true causal motivation from a long, noise-injected sequence of past interactions is highly complex.
2. **Navigating a massive search space:** The model must retrieve the correct target from an unconstrained candidate pool encompassing thousands or millions of items, rather than choosing from a pre-selected handful.

3. **Preventing error propagation:** The model may propagate errors from data generation to the training process. Our reachability-aware routing process is designed to prevent this error propagation by filtering out unreachable trajectories.

Conventional LLM alignment techniques struggle in this realistic environment due to context-utilization bias and severe reward sparsity, as demonstrated in our preliminary studies. Therefore, the core contribution of our work is fundamentally bridging the gap between LLM reasoning and practical, large-scale retrieval. By introducing the **reachability-aware routing** process and anchor-guided intent generation, our framework successfully isolates causal signals from noisy histories and establishes a grounded reasoning trajectory. This ensures that the reasoning capabilities of LLMs are practically utilized to overcome the vast search space, setting a realistic, scalable, and noise-robust foundation for future LLM-based recommendation systems.

C Analysis

Due to space limitations in the main text, we provide additional in-depth analyses in this section to further validate the effectiveness and robustness of our proposed framework. Specifically, we focus on evaluating the semantic quality and logical coherence of the generated reasoning paths, as well as on probing the generator scale and the routing-based filtering criterion.

C.1 In-Depth Quantitative Analysis of Context-Utilization Bias

To directly address the conceptual concern regarding the mechanism of context-utilization bias, we provide a rigorous, retrieval-objective quantitative analysis within the recommendation embedding space. While qualitative case studies and LLM-as-a-judge evaluations demonstrate semantic plausibility, this analysis evaluates exactly where the model’s predictions land geometrically relative to the user’s history and the ground-truth target.

C.1.1 Measurement Framework

We formulate a geometric measurement framework to quantify context-utilization bias. For each test user u , we extract three vectors mapped into the shared 128-dimensional CF latent space of a pre-trained SASRec model:

- $\mathbf{p}(u) \in \mathbb{R}^{128}$: The SASRec embedding of the *top-1 item retrieved* using the model’s generated reasoning trajectory as a dense text query.
- $\mathbf{h}_{\text{mean}}(u) = \frac{1}{|H_u|} \sum_{i \in H_u} \mathbf{e}_i$: The centroid embedding of the user’s past interaction history H_u , representing the geometric proxy of context-utilization bias (i.e., indiscriminately averaging all context).
- $\mathbf{t}(u) \in \mathbb{R}^{128}$: The embedding of the ground-truth target item i^+ .

Using the cosine distance $d(\mathbf{a}, \mathbf{b}) = 1 - \cos(\mathbf{a}, \mathbf{b})$, we define two primary directional distances: (1) *History Pull* $d_{\mathbf{p-h}}(u) = d(\mathbf{p}, \mathbf{h}_{\text{mean}})$, which measures how closely the prediction is pulled toward the historical noise centroid, and (2) *Target Distance* $d_{\mathbf{p-t}}(u) = d(\mathbf{p}, \mathbf{t})$, which measures proximity to the true next item. To normalize these metrics across users, we introduce the **Bias-Escape Score (BES)**:

$$\text{BES}(u) = 1 - \frac{d_{\mathbf{p-h}}(u)}{d_{\mathbf{p-t}}(u)} \quad (9)$$

A higher BES indicates that the prediction successfully escapes the historical context mean and aligns precisely with the target item trajectory ($\text{BES} \rightarrow 1$), while a $\text{BES} \leq 0$ implies the model remains trapped within the diffuse historical noise.

To evaluate the model’s behavior under varying task difficulties, we stratify users into **Difficulty Quartiles** based on the intrinsic task distance $d_{\text{diff}}(u) = d(\mathbf{h}_{\text{mean}}, \mathbf{t})$. Users in Q1 (Easy) have targets close to their historical mean, whereas users in Q4 (Hard) exhibit a target item located far from the interaction centroid, making context-utilization bias exceptionally lethal.

C.1.2 Controlled 3-Way Baseline Comparison

To isolate the distinct contribution of our reachability-aware curation from the generic effects of fine-tuning, we perform a controlled evaluation across three models using the *Beauty* and *Sports* datasets:

- **A. Base:** The *gemma-3-1b-it* baseline under zero-shot sequential prompting.
- **B. Rec-R1 (Pure RL):** Optimized directly via GRPO using purely rank-based retrieval rewards (NDCG), without any curation or semantic grounding.

Table 8: Overall context-utilization bias metrics evaluated in the SASRec embedding space.

Dataset	Model	$d_{p-h} \downarrow$	$d_{p-t} \downarrow$	BES $\uparrow$
<i>Beauty</i>	A. Base	0.221	0.322	0.014
	B. Rec-R1 (RL)	0.210	0.311	0.295
	C. GEARec (Ours)	0.198	0.301	0.318
<i>Sports</i>	A. Base	0.238	0.354	0.043
	B. Rec-R1 (RL)	0.184	0.356	0.339
	C. GEARec (Ours)	0.207	0.334	0.391

Table 9: Bias-Escape Score (BES) and Retrieval Hit@10 stratified across Difficulty Quartiles on the *Beauty* dataset.

Metric	Model	Q1 (Easy)	Q2	Q3	Q4 (Hard)
BES	A. Base	0.036	0.014	0.005	0.000
	B. Rec-R1 (RL)	0.152	0.275	0.350	0.402
	C. GEARec (Ours)	0.176	0.302	0.373	0.420
	<i>Gain</i>	+16%	+10%	+7%	+5%
Hit@10	A. Base	0.020	0.011	0.012	0.010
	B. Rec-R1 (RL)	0.108	0.048	0.026	0.019
	C. GEARec (Ours)	0.116	0.052	0.025	0.022
	<i>Gain</i>	+7%	+8%	-4%	+14%

- **C. GEARec (Ours):** Fine-tuned via DFT on our reachability-curated supervision dataset $\mathcal{D}_{SFT}$.

C.1.3 Empirical Findings and Curation Isolation

Table 8 summarizes the aggregate metrics. The base model (A) yields a BES near zero, serving as empirical verification that unconstrained LLMs natively suffer from intense context-utilization bias, sitting almost exactly at the interaction history mean. When aligned with reinforcement learning (B: Rec-R1), the model opens up a substantial gap in BES, showing that parametric fine-tuning itself forces the model to move away from the centroid. Crucially, transitioning from the pure RL baseline to our reachability-curated supervision (Model C: GEARec) provides an incremental yet consistent improvement, boosting the overall BES and the Hit@10.

The necessity of reachability-aware curation becomes explicitly clear when observing the **Difficulty Quartile breakdown** in Table 9. In Q4 (the hardest cases where the target is significantly distant from the context history mean), the zero-shot model completely succumbs to the bias. While GEARec achieves a better BES and hit@10 over the base LLM. More importantly, this geometric correction translates directly into empirical retrieval accuracy. As shown in the Hit@10 tier of Table 9, GEARec’s relative advantage over the strong Rec-R1 baseline scales monotonically with

Table 10: Specialization over scale (NDCG@10). Comparison between a 1B GEARec and 12B to 80B generators and a frontier API.

Inference LLM	Params	Sports	Beauty	Toys
GPT-5.1	frontier	0.0196	0.0334	0.0354
Qwen3-80B	80B	0.0194	0.0306	0.0358
Gemma3-27B	27B	0.0195	0.0318	0.0365
Gemma3-12B	12B	0.0126	0.0198	0.0297
GEARec (Gemma3-1B)	1B	0.0246	0.0466	0.0540

difficulty, jumping from a **+7%** relative improvement in the easiest quartile (Q1) to a powerful **+14%** advantage in the hardest quartile (Q4). This behavior provides definitive, mechanistic proof for the core thesis of our work: while general fine-tuning and RL alignment resolve broad domain mapping, it is the *anchor-grounded curation quality* that prevents exploration collapse and drives accuracy precisely where context-utilization bias is most destructive.

C.2 Specialization Over Scale

To examine whether the 1B GEARec backbone serves as a performance bottleneck, we maintain an identical projection-layer alignment protocol while swapping only the trajectory generator with progressively larger LLMs, up to an 80B open model and a frontier API. Crucially, in this controlled setup, these larger commercial and open models are not fine-tuned, whereas our compact 1B model is fine-tuned exclusively on our reachability-curated supervision data.

Table 10 reports the NDCG@10 results under full-pool evaluation across three benchmarks. Remarkably, our curated 1B GEARec consistently matches or outperforms the massive 12B–80B generators and the frontier API by a wide margin. This outcome underscores a critical insight: training a compact model on domain-specialized, high-quality curated data is far more decisive for sequential retrieval than merely relying on raw parameter scale or general reasoning capacity. Ultimately, these findings strongly demonstrate the structural superiority and efficacy of our data curation methodology.

C.3 Anchor-Window Robustness

We additionally probe sensitivity to the anchor window N , the number of recent history items considered by the LLM anchor extractor. Table 11 reports NDCG@10 under full-pool evaluation for $N = 8$ (main) and $N = 16$, against the best published baseline on each dataset. Both

Table 11: Anchor window robustness (NDCG@10, full item-pool). Both $N = 8$ and $N = 16$ exceed the best published baseline across all four datasets.

Setting	Sports	Beauty	Toys	Yelp
Best published baseline	0.0210	0.0398	0.0458	0.0437
GEARec ($N = 8$, main)	0.0236	0.0466	0.0525	0.0611
GEARec ($N = 16$)	0.0222	0.0449	0.0522	0.0597
Δ vs. baseline ($N = 16$)	+5.7%	+12.8%	+14.0%	+36.6%

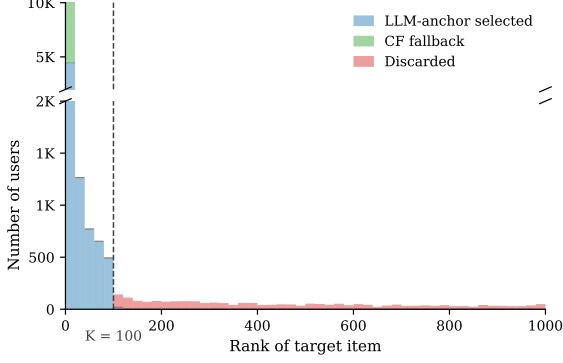


Figure 9: Routing decomposition of $\mathcal{D}_{\text{SFT}}$ on Beauty. Per-user best rank of the target item is split into LLM-anchor selected, CF-fallback selected, and Discarded. The $K=100$ reachability threshold is marked; the y-axis is broken at 2K to expose the long tail.

settings exceed the best baseline across all four domains, indicating that the reachability criterion is not knife-edge in the window choice: the gain comes from anchor-grounded reachable supervision rather than a fine-tuned window length. We do not claim long-term signals are unhelpful in principle; under current context-utilization bias, bounded windows yield more reachable supervision, and a long-term-aware anchor selector is left as future work.

C.4 Routing Composition and Filtering Calibration

To make the routing decision concrete, Figure 9 decomposes the curated training pool $\mathcal{D}_{\text{SFT}}$ for Beauty into the three routing buckets by the rank of the target item under the LLM-anchor proposal: *LLM-anchor selected* (rank $\leq K = 100$), *CF-fallback selected* (LLM rank $> K$ but recovered by the collaborative anchor), and *Discarded* (unreachable on both views). The discarded distribution has long-tail mass beyond rank 1,000, while the kept buckets concentrate near rank 1, which is what we mean by “unreachable supervision is filtered out before training, not learned around.”

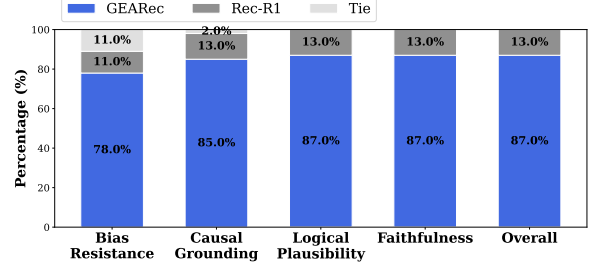


Figure 10: LLM-as-a-judge evaluation against Rec-R1 on Beauty.

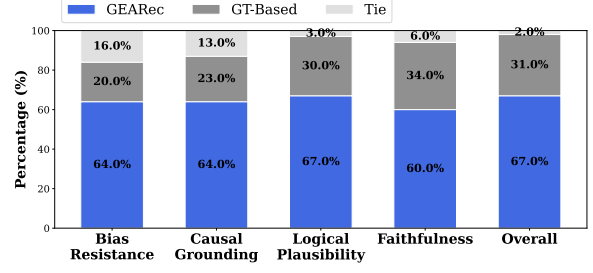


Figure 11: LLM-as-a-judge evaluation against GT-based reasoning on Beauty.

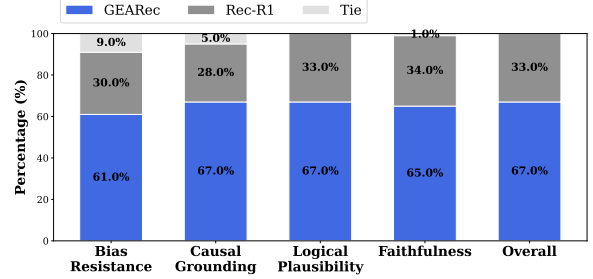


Figure 12: LLM-as-a-judge evaluation against Rec-R1 on Yelp.

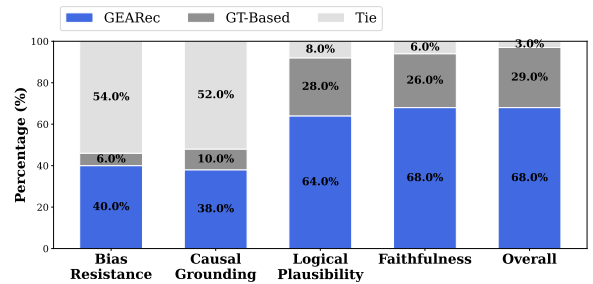


Figure 13: LLM-as-a-judge evaluation against GT-based reasoning on Yelp.

C.5 Reasoning Quality Analysis

To assess the semantic quality of the reasoning paths generated by GEARec, we employ an LLM-as-a-judge evaluation protocol. While standard retrieval metrics (e.g., NDCG, Hit Ratio) effectively measure final recommendation accuracy, they may not fully capture the quality of the reasoning that produces those rankings. Therefore, we use an

LLM as an impartial judge to perform pairwise comparisons of our generated trajectories against baselines on the Beauty and Yelp datasets.

We first compare GEARec against Rec-R1, a representative RL-based approach, on Beauty (Figure 10) and Yelp (Figure 12). Across both domains, our method achieves comparable or superior reasoning quality on most axes. RL-based autonomous exploration in massive item pools suffers from reward sparsity and context-utilization bias, which can degrade the coherence of the generated text. In contrast, our reachability-aware routing explicitly grounds the generation on a valid anchor, preserving fidelity to the user’s intent.

Figures 11 and 13 then compare GEARec against a GT-based reasoning baseline, where the model is trained on trajectories generated with explicit access to the target item. Even against this privileged variant, GEARec remains competitive on Beauty and Yelp, indicating that anchor-grounded reachable supervision captures most of what target leakage offers, without the inference-time leakage. While the absence of human evaluation remains a limitation of this study, these LLM-as-a-judge results across two domains underscore the practical viability of our anchor-guided data curation strategy.

C.6 Reasoning Fidelity Analysis

Beyond the aggregate LLM-as-a-judge scores, two case studies make the fidelity argument concrete. Figure 14 (*Noise Filtering*) shows that routing through a verified anchor (Kabuki Brush) keeps the reasoning faithful to the user’s actual intent (makeup tools), while Rec-R1 and the GT-based baseline force-link irrelevant history items. Figure 15 (*Domain Consistency*) shows the complementary case under a sparse, single-brand history: GEARec stays inside the brand-aligned cleansing context, whereas baselines drift into an unrelated category or misread the target item. Without strict causal grounding, baseline models frequently hallucinate non-existent products or misinterpret the user’s core intent. By forcing the LLM to route its reasoning strictly through the verified anchor item, GEARec reduces hallucinatory behavior, keeping the generated reasoning both logically sound and faithful to the factual user history.

As shown in **Case Study: Domain Consistency** demonstrates our framework’s robustness against hallucination. Without strict causal

Case Study: Noise Filtering

[User History]

1. Suave Shampoo
- 2. Exfoliating Bath Towel
- 3. Derma Roller
- 4. **Flat Top Kabuki Brush**
- 5. **Detangling Brush Set**
- 6. Vitamin C Serum

[Target Item]

Beauty Blender Makeup Sponges 4pc Set

[GEARec (Ours)] ✓

The user’s recent purchase demonstrates a strong interest in **makeup tools and application techniques**. They’ve previously shown a preference for **brushes and tools** (Kabuki Brush) that enhance their makeup application...

[The sponge set aligns with their previous purchases of brushes]

[Rec-R1] ✗

1. Makeup (Keywords: Foundation Brush) 2. **Hair Care**: (Rationale: Purchase of **Suave Shampoo** suggests interest in grooming)
3. **Anti-Aging**: (Rationale: **Derma Roller** indicates a desire to address wrinkles)...

[Over-generalized list-based reasoning]

[GT-Based] ✗

Based on skincare, **hair care tools** (Suave), and makeup brushes... The inclusion of the **Derma Roller** suggests a desire for deeper skin treatment...

[Attempts to force-link irrelevant item in the history]

Figure 14: Case study on Beauty: routing through a verified anchor (Kabuki Brush) keeps the reasoning faithful to the user’s actual intent (makeup tools), while Rec-R1 and the GT-based baseline force-link irrelevant history items.

grounding, baseline models frequently hallucinate non-existent products or misinterpret the user’s core intent. By forcing the LLM to route its reasoning strictly through the verified anchor item, GEARec drastically reduces hallucinatory behavior, ensuring that the generated reasoning is both logically sound and strictly faithful to the factual user history.

D Clarification on Training Supervision and Inference Consistency

To clarify the scope of supervision used in GEARec, we further discuss the role of target-aware offline curation during training and the consistency between the training pipeline and test-time inference.

D.1 Target-Aware Offline Curation and Test-Time Inputs

A potential concern in GEARec is the use of the target item i^+ during CF anchor selection (Equation 3) and reasoning generation (Equation 4). We emphasize that this target-aware procedure is used only during the **offline data curation stage**, where

Case Study: Domain Consistency

[User History]

1. Dermalogica Active Moist
→ 2. Dermalogica Clearing Skin Wash

[Target Item]

Dermalogica Clearing Skin Wash (8.4 oz)

[GEARec (Ours)] ✓

The user's recent purchase of the Dermalogica Clearing Skin Wash suggests a desire for deep cleansing... They've previously shown interest in **Dermalogica products**, indicating a **preference for effective cleansing routines within the same brand**.

[Rec-R1] ✗

3. **Nail Care**: (Rationale: The purchase of **OPI Soft Shades nails**) ... Related Product Ideas: **Dermalogica Nail Polish**.

[Misaligned recommendations]

[GT-Based] ✗

The individual is actively seeking solutions for **oily or congested skin and blemishes** ... Brand is known for its **moisturizing properties**.

[Inaccurate context for a Cleanser]

Figure 15: Case study on Beauty: with only two same-brand history items, GEARec stays within the brand-aligned cleansing context, while Rec-R1 drifts into an unrelated category (nail care) and the GT-based baseline misreads the target as a moisturizer.

it serves as a privileged teacher signal for constructing supervised fine-tuning data.

The goal of this generation pipeline is to build a higher-quality filtered dataset ($\mathcal{D}_{SFT}$) that provides more target-relevant reasoning trajectories from the user history to the next item. In large item pools, purely autonomous exploration often yields sparse successful trajectories. Under this setting, incorporating the target item during offline generation provides training-time guidance that helps translate collaborative signals into textual reasoning trajectories.

Importantly, **no target information is used during test-time inference or evaluation**. At inference time, GEARec receives only the user history, under the same input setting as all compared baselines. Therefore, the gains reported in Table 1 do not arise from additional test-time information, but from the effectiveness of the offline curation strategy used to construct training supervision.

D.2 Offline Search for Supervision and Single-Pass Inference

During training data construction, GEARec uses an offline search procedure, including prefix-

enforced generation, Best-of-N sampling, and reachability-aware routing, to identify higher-quality reasoning trajectories. At inference time, in contrast, the fine-tuned LLM generates the anchor and reasoning trajectory in a single pass, without relying on the external search procedure.

This design follows the standard objective of supervised fine-tuning: the offline routing process is used to construct better supervision, and the resulting dataset $\mathcal{D}_{SFT}$ is then used to train the model to approximate the curated reasoning behavior. In this sense, the search procedure is part of supervision construction rather than part of the deployed inference pipeline.

The empirical improvements of GEARec are consistent with this design choice. If the model did not benefit from the curated supervision, its inference-time performance would be less likely to remain competitive in large candidate spaces. The fact that GEARec consistently outperforms strong embedding-based and RL-based baselines at test time suggests that the model successfully learns from the curated teacher trajectories, narrowing the gap between guided offline supervision and autonomous inference.