

HCGRec: Hint-Conditioned Generative Recommendation with Semantic IDs

Kangning Zhang^{*}
zhangkangning@sjtu.edu.cn
Shanghai Jiao Tong University
Shanghai, China

Haotian Fang^{*}
fanghaotian@sjtu.edu.cn
Shanghai Jiao Tong University
Shanghai, China

Xukun Luo
luoxukun@meituan.com
Mei Tuan
Beijing, China

Hao Yin
yinhao15@meituan.com
Mei Tuan
Beijing, China

Yang Gao
gaoyang94@meituan.com
Mei Tuan
Beijing, China

Peng Yan
yanpeng04@meituan.com
Mei Tuan
Beijing, China

Weiwen Liu[✉]
wwliu@sjtu.edu.cn
Shanghai Jiao Tong University
Shanghai, China

Weinan Zhang[✉]
wnzhang@sjtu.edu.cn
Shanghai Jiao Tong University
Shanghai, China

Yong Yu[✉]
yyu@apex.sjtu.edu.cn
Shanghai Jiao Tong University
Shanghai, China

Abstract

Semantic-ID generative recommenders represent each item as a short sequence of discrete semantic tokens and predict the next item by autoregressively generating this token sequence. This paradigm enables a unified generation interface for item IDs, histories, and item text, but it also creates a structured optimization bottleneck during reward-based post-training: when an early semantic token enters the wrong branch of the item-token space, finite rollout groups rarely reach the ground-truth item, so group-relative optimization receives identical zero rewards and produces no useful advantage. We propose *Hint-Conditioned Generative Recommendation* (HCGRec), a semantic-ID generative recommendation framework that recovers learning signal for such hard training instances. HCGRec diagnoses each instance with checkpoint rollouts and supplies a minimal target-prefix hint only when the current generator cannot reach the correct item. The model then generates the unhinted suffix under the hinted semantic branch, turning zero-reward groups into informative comparisons over item-token completions. Hinting also changes token identity: hinted prefix tokens are oracle-provided item context, while unhinted suffix tokens are sampled generation actions. We therefore introduce *hint-aware credit decomposition*, using supervised learning to preserve item-semantic and prefix-structure alignment for hinted tokens and GRPO to optimize the sampled suffix. Experiments on sequential recommendation benchmarks show that HCGRec substantially improves over supervised fine-tuning and vanilla reward-based post-training, while reducing zero-advantage training samples from over 70% to below 20%. The code is accessible at <https://github.com/WncFht/GRec>.

CCS Concepts

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

Keywords

Generative Recommendation, Sequential Recommendation, Semantic ID, Reward-Based Post-Training

ACM Reference Format:

Kangning Zhang, Haotian Fang, Xukun Luo, Hao Yin, Yang Gao, Peng Yan, Weiwen Liu, Weinan Zhang, and Yong Yu. 2026. HCGRec: Hint-Conditioned Generative Recommendation with Semantic IDs. In *Proceedings of The 35th ACM International Conference on Information and Knowledge Management (CIKM '26)*. ACM, New York, NY, USA, 12 pages.

1 Introduction

Generative recommendation turns item retrieval into language-model-style generation: instead of scoring every candidate item, the model generates the discrete identifier of the next item a user is likely to consume. This formulation is attractive for large-scale sequential recommendation because item identifiers can be organized as short Semantic ID token sequences, often learned from item text or multimodal representations through clustering or residual quantization, and a Transformer can then predict the next item by autoregressively decoding these tokens [19]. It also creates a natural interface with large language models: interaction histories, item titles, descriptions, and other textual fields can be used to align item identifiers with the model’s semantic space [7, 39].

Recent semantic-ID generative recommenders usually follow a two-stage recipe: supervised semantic alignment followed by reward-based post-training. A model is first supervised on several semantic-alignment tasks, such as predicting the next item’s Semantic ID from historical Semantic ID sequences, from historical title sequences, or from item-side text. It is then optimized with recommendation rewards to align generated items with user preference and business objectives [2, 39, 40, 43]. Group Relative Policy Optimization (GRPO) is a common optimizer for this stage because

^{*}These authors contributed equally to this work.
✉Corresponding authors.

it compares multiple generated item-token sequences from the same prompt and avoids training a separate value model [21]. In principle, this converts recommendation feedback into a learning signal over generated Semantic ID tokens.

However, the central failure mode is that many semantic-ID training examples become *finite-rollout unreachable* under reward-based generative training. Item-level reward is sparse, but the more important issue is that this sparsity is organized by the Semantic ID prefix tree. A generated sequence receives useful signal only when it reaches the ground-truth item or a sufficiently relevant item; if an early token chooses the wrong semantic branch, later tokens are generated under an already incorrect prefix and are unlikely to recover the target item. As a result, under the current generator and a fixed rollout budget, all sampled completions can stay outside the reward-distinguishable region, all rewards are identical, and GRPO produces zero relative advantage. In our logged unhinted post-training runs, a large portion of rollout groups remains inactive even late in training, meaning that substantial training data no longer contributes useful group-relative updates.

This paper argues that the obstacle is not merely sparse reward, but *reward reachability within the item-token space*. Reward shaping can provide denser values after a generated sequence is evaluated, and correction-based decoding can repair trajectories after the model has already produced a draft [29]. These strategies are useful, but they do not directly address the training-time question: **how can a semantic-ID generator learn from an example when its current rollout distribution almost never enters the part of the item-token tree where reward becomes distinguishable?** The structure of semantic item identifiers gives a domain-specific answer: the target Semantic ID prefix itself can serve as a minimal scaffold that moves an unreachable training example back into the correct item branch.

We propose **Hint-Conditioned Generative Recommendation** (HCGRec) to recover reachability for hard semantic-ID training examples. Before reward-based post-training, HCGRec runs the current checkpoint on each training instance to estimate whether the ground-truth item is reachable under the current generator. If the model can already generate the target item, the instance is trained without modification. If the instance is unreachable, HCGRec exposes a short target-prefix hint, such as the first Semantic ID token, and asks the model to generate the remaining suffix. The hint is used only as a training-time scaffold for generative recommendation: it does not add side information at inference time, and it is applied only to examples whose unhinted generations fail to produce learning signal.

The key insight is that hinting is **prefix-tree reachability control**, not answer leakage. A four-token item identifier defines a coarse-to-fine path in the item space. When the generator repeatedly chooses the wrong first token, the suffix-generation problem is ill-posed because all suffixes are conditioned on the wrong branch. Revealing the first token for such hard cases relocates generation to the correct coarse branch, where the remaining tokens still need to be generated by the model and where group-relative rewards can distinguish better and worse item completions. This converts previously inactive examples into active generative training groups while preserving on-policy learning for the unhinted suffix.

HCGRec also requires **hint-aware credit decomposition** inside a hinted semantic-ID generation. Once part of the target Semantic ID is inserted into the input, those tokens are no longer sampled actions from the recommender; they are oracle-provided item context. Treating the full Semantic ID sequence as a generated action would therefore mix two different sources of information: the hinted prefix did not result from model exploration and should not receive GRPO credit, whereas the unhinted suffix is the model’s actual generation and should be optimized by GRPO. At the same time, the hinted prefix cannot simply be ignored, because it carries item semantics and the coarse-to-fine prefix structure used to locate the correct item branch. We therefore use supervised learning to provide semantic anchoring for the hinted prefix, while GRPO optimizes the sampled suffix under the hinted context. This differs from global supervised-plus-reward regularization [39, 43]: the learning signal is assigned according to whether each token is oracle-provided item context or a model-generated item-token action.

Empirically, HCGRec improves semantic-ID generation by turning inactive hard examples into active generative training groups. We evaluate HCGRec on sequential recommendation benchmarks under the same semantic-index setting as the base generative recommender. The main result is that reachability-aware hinting is competitive with SFT-only training and reward-based generative baselines and improves many key recommendation metrics, especially on deeper ranking cutoffs where stable semantic-branch generation matters. More importantly, it changes the training dynamics by substantially reducing inactive zero-gradient rollout groups during post-training. Ablations further separate the effect of the hinting policy, the training-task scope, and hint-aware credit decomposition, demonstrating that the gains come from targeted reachability recovery rather than simply adding more supervised tokens.

Our contributions are summarized as follows:

- We identify **finite-rollout unreachable reward groups** as a major bottleneck in semantic-ID generative recommendation, where multi-token item identifiers make early-token errors collapse most reward-based training groups to zero advantage.
- We introduce **HCGRec**, a hint-conditioned generative recommendation framework that diagnoses unreachable training instances with checkpoint rollouts and supplies minimal target-prefix hints only for hard examples, recovering informative suffix-generation comparisons without changing inference.
- We propose **hint-aware credit decomposition**, which treats hinted prefix tokens as oracle-provided item context and sampled suffix tokens as generated item-token actions, using supervised learning to preserve prefix semantics and GRPO to optimize suffix generation.
- We show empirically that HCGRec improves many key generative recommendation metrics and substantially reduces the inactive-sample ratio on logged post-training runs, providing direct evidence that reachability recovery is critical for semantic-ID generative recommendation.

2 Related Work

2.1 Sequential and Semantic-ID Generative Recommendation

Sequential recommendation traditionally formulates next-item prediction as discriminative ranking over a catalog. GRU4Rec [8] models interaction histories with recurrent neural networks, Caser [25] uses convolutional sequence embeddings to capture local sequential patterns, and SASRec [12] and BERT4Rec [23] establish self-attention as a strong backbone for user behavior modeling. Generative recommendation changes the output interface by generating an item identifier rather than scoring every item. TIGER is a representative semantic-ID framework [19]: item content representations are quantized into short token sequences, and a Transformer predicts the next item’s Semantic ID by autoregressive decoding. Follow-up studies extend this idea with contrastively learned tree-structured identifiers [22], unified generative-dense retrieval [30], and practical semantic-ID pipelines [11].

Recent work further studies how semantic identifiers should be constructed and used. CoST [41] improves semantic tokenization with contrastive quantization, MTGRec [38] augments pre-training with multiple identifiers per item, HiD-VAE [4] learns hierarchical and disentangled identifiers, and DIGER [5] makes semantic-ID learning differentiable. Variable-length semantic IDs [13] and long semantic IDs [27] further revisit the fixed-length bottleneck of residual quantization. Other generative-recommendation studies address broader settings, such as joint generative search and recommendation [17], cross-task search-recommendation transfer [18], contextualized generative recommendation with SynerGen [6], codebook rebalancing with CRAB [3], and non-uniform quantization with CarD [26], and diffusion-based cold-start item modeling [34] and ID-free multimodal token representations [31]. These works primarily improve the item-token space itself. HCGRec is orthogonal: given a semantic-ID space, it studies why reward-based post-training cannot learn from many hard instances and uses target-prefix hints to restore reachability inside the existing item-token tree.

2.2 Supervised Semantic Alignment for Recommendation Generation

Language-model-based recommenders often rely on supervised fine-tuning to align user histories, item text, and recommendation outputs in a shared generation space. P5 [7] formulates recommendation as a family of text-to-text tasks over interactions, metadata, and reviews. GenRec [9] and TALLRec [1] show that large language models can be adapted to recommendation through instruction-style or tuning-based supervision. LC-Rec [37] injects collaborative semantics into large language models and is closely related to the item-text and sequence fusion tasks used in our supervised stage. In semantic-ID systems, this alignment stage usually includes predicting the next Semantic ID from historical Semantic IDs, predicting identifiers from textual histories, and mapping item-side text to item identifiers. Multimodal recommendation likewise studies alignment between content and ID-based representations [15].

Industrial and large-scale generative recommenders adopt a similar SFT-first recipe. OneRec [2] unifies retrieval and ranking with

a generative recommender and iterative preference alignment. The OneRec technical report [39] and GenRec [43] combine semantic-ID generation with supervised objectives before reward-based or preference-oriented post-training. HCGRec assumes this supervised alignment has already produced a reasonable semantic-ID generator. The problem we target appears after SFT: under teacher forcing, the model can learn the target token sequence, but under finite on-policy rollouts it may still fail to enter the correct semantic branch. Therefore, our contribution is not a new SFT task or a replacement for semantic alignment. Instead, HCGRec uses the SFT checkpoint to diagnose which instances are rollout-unreachable and then applies minimal target-prefix hints only to recover reward-based learning signal for those hard cases.

2.3 Reward-Based Post-Training for Generative Recommendation

Reward-based post-training aligns generated recommendations with objectives beyond next-token likelihood. GRPO provides a practical group-relative policy optimization objective by comparing multiple rollouts from the same prompt and normalizing their rewards within the group [21]. OneRec [2] applies iterative preference alignment, and OneRec-V2 [40] further incorporates real user feedback and duration-aware reward shaping in production recommendation. GenRec [43] proposes GRPO-SR, combining GRPO with NLL regularization and hybrid rewards for preference-oriented large-scale generative recommendation. MiniOneRec [14] provides an open-source framework for scaling generative recommendation experiments, while LoopTool [33] and Harness-R1 [20] extend reinforcement learning to model-aware data evolution and executable agent harness editing.

Several recent methods study how to make reward-optimized generative recommendation more effective. ReRe [24] introduces reinforced preference optimization with constrained sampling and auxiliary ranking rewards, GRC [29] adds a generation-reflection-correction trajectory, V-STAR [10] uses value-guided structured sampling and Sibling-GRPO, and Rank-GRPO [42] changes the credit unit for conversational recommendation from a whole sequence to rank positions. These methods recognize that naive sequence-level reward optimization can produce weak, noisy, or poorly allocated learning signals. HCGRec addresses a different but complementary bottleneck: before correction, ranking, or value guidance can help, many semantic-ID training instances produce rollout groups with identical rewards because early tokens enter the wrong item branch. Our method focuses on making such hard instances reachable under the rollout budget and then assigning credit only to the actually sampled suffix actions.

2.4 Hints, Scaffolds, and Credit Assignment under Sparse Rewards

Our method is also related to hint- or scaffold-based learning under sparse verifiable rewards. HiLL observes that GRPO can suffer from advantage collapse when all rollouts in a group receive the same reward, and learns hints to recover informative groups for hard reasoning tasks [28]. Scaf-GRPO [36] similarly studies scaffolded group-relative optimization for reasoning models. These works show that external structure can make sparse-reward optimization

more effective, but they are designed for general reasoning rather than semantic-ID item generation. MMSkills [35] provides reusable multimodal procedural scaffolds for visual agents.

HCGRec differs in three ways. First, the hint is not a free-form reasoning scaffold or an inference-time assistant; it is a target-prefix token sequence defined by the semantic-ID prefix tree and used only during training. Second, the hint is selected by an offline reachability diagnosis: if an instance is already reachable without hints, no target token is exposed. Third, hinting creates a token-source-specific credit assignment problem [32]. The hinted prefix is oracle-provided item context and should be semantically anchored by supervised loss, whereas the suffix is the model’s sampled recommendation action and should receive group-relative policy credit. This token-source-aware decomposition is the key distinction between HCGRec and the more general practice of mixing supervised and reward-based losses in generative recommendation.

3 Method

This section formalizes HCGRec for semantic-ID generative recommendation. We first introduce the standard semantic-ID generation objective, then show why finite rollout groups can become reward-unreachable, and finally present reachability-aware hinting with hint-aware credit decomposition. Figure 1 provides an overview of the three central components of our method.

3.1 Semantic-ID Generative Recommendation

Let $\mathcal{I}$ denote the item catalog. Each item $v \in \mathcal{I}$ is represented by a semantic identifier $\mathbf{y}(v) = (y_1, \dots, y_M)$, where $y_t \in \mathcal{V}_{\text{SID}}$ is a discrete semantic token and M is the identifier length. In the common residual-quantization setting, M is small; for example, $M = 4$ in our implementation. Given a user context x , such as a sequence of historical items, titles, or item-side textual fields, a generative recommender defines an autoregressive distribution over the next item’s Semantic ID:

$$\pi_\theta(\mathbf{y} | x) = \prod_{t=1}^M \pi_\theta(y_t | x, y_{<t}), \quad (1)$$

where θ denotes the model parameters.

The supervised alignment stage trains the model to connect user histories, item text, and item identifiers in the same generation space. For a training set $\mathcal{D} = \{(x_i, \mathbf{y}_i^*)\}$, the basic teacher-forced loss is

$$\mathcal{L}_{\text{SFT}}(\theta) = -\mathbb{E}_{(x, \mathbf{y}^*) \sim \mathcal{D}} \sum_{t=1}^M \log \pi_\theta(y_t^* | x, y_{<t}^*). \quad (2)$$

In practice, x can be instantiated by multiple semantic-alignment tasks: historical Semantic ID sequences, title histories, or item-side text. This stage provides a reasonable generator, but it does not guarantee that on-policy rollouts can **reach the target Semantic ID path under a finite sampling budget**.

3.2 Reward-Based Post-Training and Unreachable Groups

After SFT, reward-based post-training assigns a scalar recommendation reward to generated item-token sequences. For a prompt x , we sample a group of G candidate identifiers $\{\hat{\mathbf{y}}_j\}_{j=1}^G \sim \pi_{\theta_{\text{old}}}(\cdot | x)$

and evaluate each completion with a reward $r_j = R(x, \hat{\mathbf{y}}_j, \mathbf{y}^*)$. The reward function can include exact target match, valid-item constraints, or other recommendation feedback; HCGRec only requires that higher reward means a better generated item. GRPO converts the group rewards into relative advantages:

$$A_j = \frac{r_j - \mu_r}{\sigma_r + \epsilon}, \quad \mu_r = \frac{1}{G} \sum_{j=1}^G r_j, \quad \sigma_r^2 = \frac{1}{G} \sum_{j=1}^G (r_j - \mu_r)^2, \quad (3)$$

where ϵ is a small constant. With the token-level probability ratio

$$\rho_{j,t}(\theta) = \frac{\pi_\theta(\hat{y}_{j,t} | x, \hat{y}_{j,<t})}{\pi_{\theta_{\text{old}}}(\hat{y}_{j,t} | x, \hat{y}_{j,<t})}, \quad (4)$$

and the clipped surrogate

$$\ell_{\text{clip}}(\rho, A) = \min(\rho A, \text{clip}(\rho, 1 - \epsilon, 1 + \epsilon)A). \quad (5)$$

the standard group-relative objective is

$$\mathcal{L}_{\text{GRPO}} = -\frac{1}{G} \sum_{j=1}^G \frac{1}{M} \sum_{t=1}^M \ell_{\text{clip}}(\rho_{j,t}(\theta), A_j). \quad (6)$$

The difficulty in semantic-ID recommendation is that the reward is structured by the **prefix tree of item tokens**. If early tokens choose the wrong coarse semantic branch, all later tokens are decoded under a wrong prefix and the generated identifier is unlikely to reach the ground-truth item. We define a training instance as **finite-rollout unreachable** when its sampled group does not produce reward-distinguishable completions:

$$U_\theta(x, \mathbf{y}^*) = \mathbb{I} \left[\text{Var} \left(\{R(x, \hat{\mathbf{y}}_j, \mathbf{y}^*)\}_{j=1}^{G_d} \right) \leq \tau \right], \quad \hat{\mathbf{y}}_j \sim \pi_\theta(\cdot | x), \quad (7)$$

where G_d is the diagnostic rollout budget and τ is a small threshold. For exact-match rewards, $\tau = 0$ identifies groups where every completion receives the same reward, usually all zero. In this case $A_j = 0$ for all rollouts, and Eq. (6) contributes no useful update for that training instance.

3.3 Reachability-Aware Hint Conditioning

HCGRec recovers learning signal by conditioning hard instances on the **shortest target-prefix hint** that makes the target item reachable in diagnostic rollouts. Let π_{sft} be the checkpoint after supervised semantic alignment. We use this checkpoint once before reward-based post-training to decide how much prefix information each training instance needs.

For a target identifier $\mathbf{y}^* = (y_1^*, \dots, y_M^*)$ and a candidate hint length h , define

$$\mathbf{p}_h^* = (y_1^*, \dots, y_h^*), \quad x_h = [x; \mathbf{p}_h^*], \quad (8)$$

where $\mathbf{p}_0^*$ is empty and $x_0 = x$. Under x_h , the model only needs to generate the remaining suffix. We draw a diagnostic group

$$\tilde{\mathbf{s}}_j^{(h)} \sim \pi_{\text{sft}}(\cdot | x_h), \quad \tilde{\mathbf{y}}_j^{(h)} = \mathbf{p}_h^* \oplus \tilde{\mathbf{s}}_j^{(h)}, \quad j = 1, \dots, G_d, \quad (9)$$

where $\oplus$ denotes concatenation and G_d is the diagnostic rollout budget. We say that length h passes the reachability test if at least one diagnostic completion exactly recovers the target identifier:

$$B_h(x, \mathbf{y}^*) = \mathbb{I} \left[\exists j \in \{1, \dots, G_d\} \text{ s.t. } \tilde{\mathbf{y}}_j^{(h)} = \mathbf{y}^* \right]. \quad (10)$$

Let

$$C(x, \mathbf{y}^*) = \{h \in \{0, \dots, H_{\text{max}}\} : B_h(x, \mathbf{y}^*) = 1\} \quad (11)$$

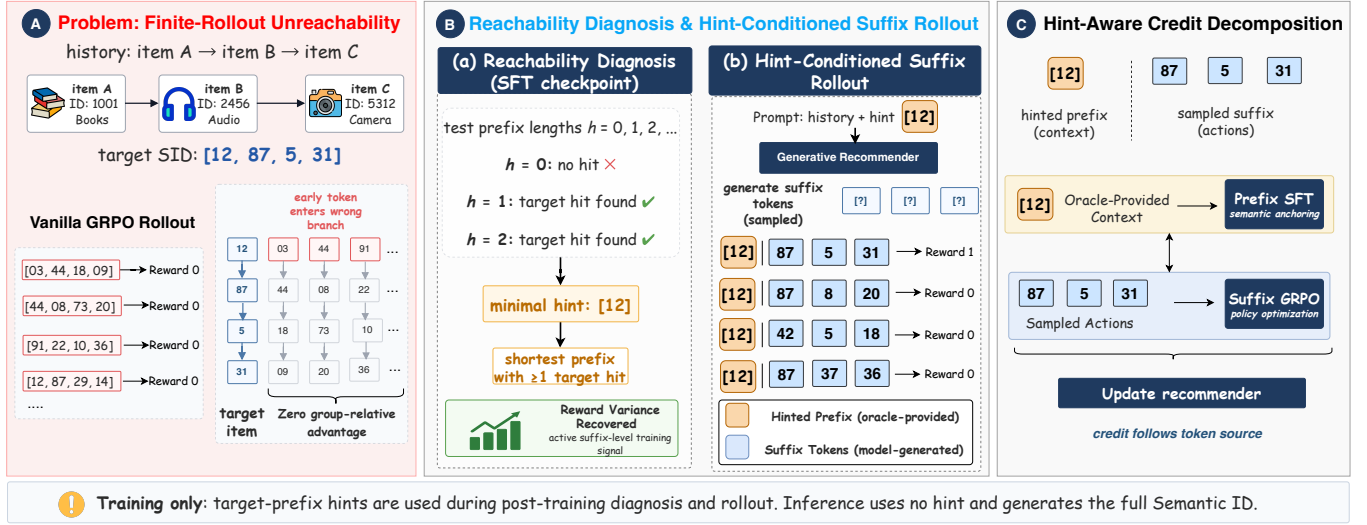


Figure 1: Overview of HCGRec. (A) In semantic-ID generative recommendation, unhinted rollouts may enter wrong prefix branches and produce identical rewards, leaving the instance inactive for group-relative optimization. (B) HCGRec diagnoses each instance with an SFT checkpoint, selects the shortest target-prefix hint that makes the target reachable within the rollout budget, and performs suffix rollouts under the hinted semantic branch. (C) Hint-aware credit decomposition treats the hinted prefix as oracle-provided context anchored by SFT, while the sampled suffix remains the model action optimized by GRPO.

be the set of prefix lengths that make the diagnostic group contain the target item. The hint used by HCGRec is the shortest prefix in this set:

$$h^*(x, y^*) = \min C(x, y^*). \quad (12)$$

This definition makes the **hint policy explicit**. If $h^* = 0$, the SFT checkpoint can already produce the target item without a hint, and the example is trained in the original form. If $h^* > 0$, the example is hard under unhinted rollout, and HCGRec reveals the **minimum target prefix** that makes the diagnostic rollout group contain a correct answer. The guarantee is therefore finite-budget and checkpoint-specific: the selected prompt has already produced at least one target hit under π_{sft} and the diagnostic budget. We choose $H_{\text{max}} < M$ so that at least one semantic token remains to be generated. If $C(x, y^*)$ is empty under this budget, the instance is recorded as unresolved in the active-sample diagnostics; a selected hint is accepted only when it satisfies $B_{h^*}(x, y^*) = 1$.

Given $h = h^*(x, y^*)$, the hinted prompt is x_h . During post-training, the generator samples suffixes

$$\hat{s}_{j,h} \sim \pi_{\theta_{\text{old}}}(\cdot | x_h), \quad \hat{y}_j^{(h)} = \mathbf{p}_h^* \oplus \hat{s}_{j,h}, \quad (13)$$

using the same recommendation reward as the unhinted task. When $h = 0$, Eq. (13) reduces to the original unhinted generation. When $h > 0$, the hinted prefix relocates the rollout to the correct coarse branch of the item-token tree, but the remaining suffix is still generated by the model and evaluated by reward. Thus, the hint is a training-time **reachability scaffold**, not inference-time side information.

3.4 Hint-Aware Credit Decomposition

Hinting changes which tokens should receive which learning signal. In a hinted instance, prefix tokens $\mathbf{p}_h^*$ are oracle-provided item

context; they are not sampled actions from the recommender. The suffix tokens $\hat{s}_{j,h}$, in contrast, are the actual generated item-token actions. HCGRec therefore **decomposes credit by token source**.

For the sampled suffix, we compute group rewards on the completed identifiers $\hat{y}_j^{(h)}$ and form advantages $A_j^{(h)}$ as in Eq. (3). The suffix probability ratio is

$$\rho_{j,t}^{(h)}(\theta) = \frac{\pi_{\theta}(\hat{y}_{j,t}^{(h)} | x_h, \hat{y}_{j,h+1:t-1}^{(h)})}{\pi_{\theta_{\text{old}}}(\hat{y}_{j,t}^{(h)} | x_h, \hat{y}_{j,h+1:t-1}^{(h)})}, \quad t > h. \quad (14)$$

The suffix-only group-relative loss is

$$\mathcal{L}_{\text{suffix}}^{\text{GRPO}} = -\frac{1}{G} \sum_{j=1}^G \frac{1}{M-h} \sum_{t=h+1}^M \ell_{\text{clip}}(\rho_{j,t}^{(h)}(\theta), A_j^{(h)}). \quad (15)$$

No GRPO term is applied to the **hinted prefix**, because those tokens were not generated by the policy.

The hinted prefix is nevertheless important: it carries item semantics and the coarse-to-fine structure that identifies the correct item branch. Ignoring it would weaken the semantic alignment that the SFT stage originally learned. We therefore add a **prefix anchoring loss**

$$\mathcal{L}_{\text{prefix}}^{\text{SFT}} = -\mathbb{I}[h > 0] \sum_{t=1}^h \log \pi_{\theta}(y_t^* | x, y_{<t}^*). \quad (16)$$

This loss asks the model to keep recognizing the hinted semantic branch from the original recommendation context, while the suffix loss optimizes the model's actual generation under that branch.

The final HCGRec objective for a training instance is

$$\mathcal{L}_{\text{HCGRec}} = \mathcal{L}_{\text{suffix}}^{\text{GRPO}} + \lambda \mathcal{L}_{\text{prefix}}^{\text{SFT}}, \quad (17)$$

Algorithm 1 HCGRec Training

Require: SFT checkpoint π_{sft} , training set $\mathcal{D}$, reward R , diagnostic budget G_d , training group size G , maximum hint length $H_{\text{max}} < M$, anchoring weight λ

- 1: Initialize $\pi_\theta \leftarrow \pi_{\text{sft}}$
- Reachability diagnosis**
- 2: **for** each $(x, y^*) \in \mathcal{D}$ **do**
- 3: **for** $h = 0, \dots, H_{\text{max}}$ **do**
- 4: Build $x_h = [x; \mathbf{p}_h^*]$ and sample G_d diagnostic suffixes from $\pi_{\text{sft}}(\cdot \mid x_h)$
- 5: Compute $B_h(x, y^*)$ by Eq. (10)
- 6: **end for**
- 7: **if** $C(x, y^*) \neq \emptyset$ **then**
- 8: Store $h^*(x, y^*) = \min C(x, y^*)$ by Eq. (12)
- 9: **else**
- 10: Mark the instance as unresolved for active-sample diagnostics
- 11: **end if**
- 12: **end for**
- Reward-based post-training**
- 13: **for** each minibatch $\mathcal{B} \subset \mathcal{D}$ **do**
- 14: **for** each resolved $(x, y^*) \in \mathcal{B}$ **do**
- 15: Let $h = h^*(x, y^*)$ and build $x_h = [x; \mathbf{p}_h^*]$
- 16: Sample G suffixes from $\pi_{\theta_{\text{old}}}(\cdot \mid x_h)$
- 17: Complete each identifier, compute rewards and group advantages
- 18: Compute $\mathcal{L}_{\text{suffix}}^{\text{GRPO}}$ by Eq. (15) and $\mathcal{L}_{\text{prefix}}^{\text{SFT}}$ by Eq. (16)
- 19: **end for**
- 20: Update θ with the minibatch average of $\mathcal{L}_{\text{HCGRec}}$ in Eq. (17)
- 21: **end for**

where λ controls the strength of semantic anchoring. This objective is different from applying a global supervised regularizer to all tokens. The **prefix receives supervised credit** because it is oracle-provided context, and the **suffix receives group-relative credit** because it is sampled behavior from the recommender.

3.5 Training and Inference

Algorithm 1 summarizes the full procedure. HCGRec starts from an SFT checkpoint, performs an offline reachability diagnosis with that checkpoint, and stores one hint length for each resolved training instance. The same reward function is used for unhinted and hinted rollouts; the only change is where generation starts in the semantic-ID prefix tree. At inference time, no target prefix is available or used. The recommender generates the complete Semantic ID from the user context exactly as in Eq. (1), so HCGRec **changes training but not the deployment interface**.

4 Experiments

We organize the experiments around four research questions. These questions examine the overall effectiveness of HCGRec, clarify how prefix hints recover useful optimization signal, and identify the contributions of training-task design and hint-aware credit decomposition.

- **RQ1: Overall performance.** Does HCGRec improve semantic-ID generative recommendation over supervised training and vanilla reward-based post-training?
- **RQ2: Deep analysis for prefix hint.** Does prefix hinting recover effective training signal, and how does the proposed offline minimal hint differ from dynamic hinting?
- **RQ3: Ablation study.** How do training tasks and loss decomposition affect HCGRec?
- **RQ4: Hint-aware credit decomposition weight.** How does the prefix anchoring weight λ in Eq. (17) affect recommendation performance across datasets?

4.1 Experimental Setup

4.1.1 Datasets. We evaluate the method on three public real-world benchmarks from the Amazon Product Reviews dataset [16]: *Musical Instruments*, *Arts, Crafts and Sewing*, and *Video Games*. For brevity, we refer to these three datasets as *Instruments*, *Arts*, and *Games*, respectively. Following the GenRec pipeline [9, 43], each item is represented by a four-token Semantic ID, and all variants share the same semantic index, backbone, and evaluation pipeline. User behavior sequences are constructed in chronological order, and the maximum history length is set to 50 for all three datasets.

Table 1 summarizes both the raw corpus scale and the resulting task-wise sample counts for the three domains.

4.1.2 Data Split and Evaluation Metrics. We adopt the leave-one-out strategy for evaluation. The most recent item in each user sequence is used for testing, the second most recent item is used for validation, and the remaining interactions are used for training. We report top- K Hit Ratio (HR@ K) and Normalized Discounted Cumulative Gain (NDCG@ K) with $K \in \{5, 10, 50\}$. Evaluation is conducted with full ranking over the **entire item set** rather than sampled negatives. For the generative methods, the beam size is fixed at 50. For each variant, one checkpoint is selected by NDCG@10 from all synchronized checkpoints of that variant, and all reported metrics are read from that same checkpoint.

4.1.3 Training Tasks. The data pipeline instantiates five task families. *SeqRec* is the basic sequential recommendation task: given the historical interaction sequence, the model predicts the next item Semantic ID. *item2index+index2item* is an item-side semantic alignment task that maps between item titles and semantic identifiers in both directions. *fusion-seqrec* conditions on the historical Semantic ID sequence but predicts the title of the next item instead of its semantic identifier. *title-seqrec* uses the title sequence of historical interacted items to predict the next item Semantic ID. *title/desc2index* is an item identification task that predicts the target Semantic ID from either an item title or an item description.

The supervised stage uses *SeqRec*, *item2index+index2item*, and *fusion-seqrec*. Reward-based post-training starts from the same SFT checkpoint in all comparisons, and its training pool is built from *SeqRec*, *title-seqrec*, and *title/desc2index*. In other words, RL keeps the sequential next-Semantic ID objective from *SeqRec*, adds title-sequence-based next-Semantic ID prediction through *title-seqrec*, and includes title/description-to-Semantic ID grounding through *title/desc2index*, while *item2index+index2item* and *fusion-seqrec* remain SFT-only objectives.

Table 1: Statistics of the three evaluation domains and the resulting sample counts for the training tasks. *SeqRec* reports train/valid/test counts; *item2index+index2item*, *fusion-seqrec*, *title-seqrec*, and *title/desc2index* report train-only counts. The last two columns summarize the total numbers of SFT and RL training samples, respectively. Sparsity is computed as $1 - \#Interactions / (\#Users \times \#Items)$.

Dataset	Raw Data					Training Samples						
	#Users	#Items	#Interactions	Sparsity	Avg. len.	seqrec tr/va/te	item2index +index2item	fusion-seqrec	title-seqrec	title/desc 2index	SFT total	RL total
Instruments	17,112	6,250	136,226	99.87%	7.96	84,890 / 17,112 / 17,112	12,465	84,890	10,000	11,551	182,245	106,441
Arts	22,171	9,416	174,079	99.92%	7.85	107,566 / 22,171 / 22,171	18,743	107,566	10,000	17,061	233,875	134,627
Games	42,259	13,839	373,514	99.94%	8.84	246,737 / 42,259 / 42,259	27,287	246,737	10,000	23,373	520,761	280,110

4.1.4 Reward Function. For reward-based post-training, we use an **Exact Target Match Reward**. Given a target semantic identifier $\mathbf{y}^* = (y_1^*, \dots, y_L^*)$ and a generated semantic identifier $\hat{\mathbf{y}} = (\hat{y}_1, \dots, \hat{y}_L)$, the rollout reward is

$$r(\hat{\mathbf{y}}, \mathbf{y}^*) = \mathbb{I}[\hat{\mathbf{y}} = \mathbf{y}^*], \quad (18)$$

where the reward is 1 only when the generated Semantic ID exactly matches the ground-truth target Semantic ID, and 0 otherwise. For hinted rollouts, the hinted prefix is treated as oracle-provided context, and the exact-match check is applied to the complete target Semantic ID after concatenating the hint and the generated suffix.

4.1.5 Implementation Details. We initialize the generative recommender from Qwen2.5-3B-Instruct and perform full-parameter supervised fine-tuning. Unless otherwise noted, supervised training uses a maximum sequence length of 512, bf16 precision, a cosine learning-rate schedule, learning rate 3×10^{-4} , 8 GPUs, per-device batch size 32, and gradient accumulation over 8 steps. Reward-based post-training uses 8 processes with the exact target match reward, per-device train/eval batch size 64, gradient accumulation over 2 steps, learning rate 10^{-5} , 2 training epochs, temperature 1.0, maximum completion length 128, and 16 sampled completions per prompt. The offline diagnostic pass uses beam size 16, maximum hint depth 3, and default unsolved depth 3. The prefix-anchoring coefficient is set to 0.005 unless otherwise stated.

4.2 RQ1: Overall Recommendation Performance

We compare HCGRec with representative sequential recommendation baselines, including Caser [25], GRU4Rec [8], BERT4Rec [23], SASRec [12], TIGER [19], and LC-Rec [37]. We also include three reward-based generative baselines, namely *GRPO Rule-only*, *MiniOneRec* [14], and *HCGRec (offline hint)*. For a controlled post-training comparison, *GRPO Rule-only*, *MiniOneRec*, *HCGRec (offline hint)*, and HCGRec all start from the same SFT checkpoint and use the same semantic index, rollout group size, reward function, decoding constraints, training budget, and checkpoint selection rule. Results are reported in Table 2. The compared methods are summarized below.

- **Caser** [25] is a convolutional sequential recommender that models local patterns in recent user interactions with horizontal and vertical convolutional filters.
- **GRU4Rec** [8] is a recurrent sequential recommendation baseline that encodes the interaction history with gated recurrent units and scores candidate next items from the hidden state.

- **BERT4Rec** [23] uses bidirectional Transformer encoding and masked item prediction, providing a strong non-generative sequential modeling baseline.
- **SASRec** [12] is a self-attentive sequential recommender that models long-range dependencies with causal attention and predicts the next item by item scoring.
- **TIGER** [19] represents the semantic-ID generative retrieval paradigm, in which items are represented as discrete semantic tokens and recommendation is formulated as identifier generation.
- **LC-Rec** [37] adapts language-model-style supervision to collaborative semantics and serves as the supervised semantic-alignment checkpoint in the present pipeline.
- **GRPO Rule-only** is a rule-reward variant of generative recommendation that directly optimizes full generated semantic identifiers with the exact target match reward, without prefix hinting or hint-aware credit decomposition.
- **MiniOneRec** [14] is a reward-based generative recommendation baseline that follows the OneRec-style post-training recipe with semantic identifiers.
- **HCGRec (offline hint)** is the offline minimal-hint post-training variant that reuses the offline diagnosed minimal reachable prefix during reward-based optimization, but does not apply the hint-aware credit decomposition used by the full method.
- **HCGRec** is our method, which combines reachability-aware prefix hinting with hint-aware credit decomposition during post-training.

Based on these comprehensive results, we make the following observations.

HCGRec delivers its clearest gains where semantic reachability matters most. Table 2 shows that HCGRec is not uniformly best on every dataset and cutoff. On Instruments, it gives the best HR@50 (0.1985) and NDCG@50 (0.1118), while the rule-reward and offline-hint variants remain stronger on several top-rank metrics. On Arts and Games, HCGRec is strongest or tied for strongest on the headline HR@5/HR@10 and NDCG@10 metrics, but TIGER still leads on HR@50. The consistent pattern is therefore more specific than a blanket win: HCGRec improves many key metrics while leaving a few extreme operating points to specialized baselines.

The comparison with HCGRec (offline hint) separates the effect of reachability correction from the effect of credit decomposition. Once an offline reachable prefix is introduced, performance already becomes competitive, which confirms that reachability correction is the primary intervention. The full method further

Table 2: Overall results on Instruments, Arts, and Games. Bold and underline mark the best and second-best results; highlight marks the full HCGRec.

Dataset	Metric	Sequential Rec.				Generative SFT		Reward-based Generative Rec.			
		Caser	GRU4Rec	BERT4Rec	SASRec	TIGER	LC-Rec	GRPO Rule-only	MiniOneRec	HCGRec (offline hint)	HCGRec
<i>Gen.</i>		–	–	–	–	✓	✓	✓	✓	✓	✓
<i>SID</i>		–	–	–	–	✓	✓	✓	✓	✓	✓
<i>RL</i>		–	–	–	–	–	–	✓	✓	✓	✓
Instruments	HR@5	0.0645	0.0793	0.0790	0.0684	0.0872	0.0932	0.1027	0.1002	<u>0.1014</u>	0.1009
	HR@10	0.0825	0.0880	0.0937	0.0869	0.1044	0.1094	0.1179	0.1145	0.1189	<u>0.1180</u>
	HR@50	0.1593	0.1291	0.1541	0.1657	0.1763	0.1844	0.1681	0.1696	<u>0.1941</u>	0.1985
	NDCG@5	0.0533	0.0735	0.0689	0.0568	0.0771	0.0771	0.0911	<u>0.0906</u>	0.0875	0.0890
	NDCG@10	0.0591	0.0763	0.0735	0.0628	0.0825	0.0823	0.0960	<u>0.0952</u>	0.0931	0.0945
	NDCG@50	0.0757	0.0850	0.0865	0.0799	0.0980	0.0985	0.1070	0.1071	<u>0.1094</u>	0.1118
Arts	HR@5	0.0421	0.0682	0.0563	0.0593	0.0905	0.0949	0.1031	0.1018	<u>0.1041</u>	0.1048
	HR@10	0.0547	0.0812	0.0734	0.0814	0.1154	0.1164	0.1215	0.1189	<u>0.1248</u>	0.1257
	HR@50	0.1103	0.1428	0.1431	0.1627	0.2076	0.1941	0.1906	0.1839	<u>0.2007</u>	0.2006
	NDCG@5	0.0347	0.0598	0.0461	0.0426	0.0725	0.0801	0.0893	0.0876	<u>0.0890</u>	0.0889
	NDCG@10	0.0387	0.0640	0.0516	0.0498	0.0805	0.0870	<u>0.0952</u>	0.0931	0.0956	0.0956
	NDCG@50	0.0508	0.0772	0.0666	0.0673	0.1006	0.1038	0.1102	0.1071	0.1120	<u>0.1118</u>
Games	HR@5	0.0334	0.0358	0.0465	0.0340	0.0489	0.0503	0.0544	0.0541	<u>0.0555</u>	0.0558
	HR@10	0.0546	0.0574	0.0723	0.0556	0.0806	0.0804	<u>0.0825</u>	0.0802	0.0857	0.0857
	HR@50	0.1517	0.1621	0.1875	0.1571	0.2118	0.1998	0.1815	0.1804	0.1972	0.2012
	NDCG@5	0.0218	0.0232	0.0304	0.0218	0.0318	0.0336	0.0377	0.0374	<u>0.0383</u>	0.0385
	NDCG@10	0.0286	0.0301	0.0387	0.0287	0.0419	0.0433	<u>0.0467</u>	0.0458	0.0480	0.0480
	NDCG@50	0.0496	0.0526	0.0636	0.0505	0.0704	0.0691	0.0683	0.0676	<u>0.0723</u>	0.0732

improves over the offline-hint-only variant on several metrics, including HR@50 and NDCG@50 on Instruments, HR@5/HR@10 on Arts, and HR@5/NDCG@50 on Games. At the same time, offline hinting remains equal or better on some other cutoffs. This mixed but structured pattern supports the credit-decomposition argument without overstating it: token-source-aware optimization is helpful, but it is not a universal improvement on every metric.

The remaining weaknesses are structurally consistent with the design of the method. The cases in which HCGRec does not dominate are concentrated in two settings: very early ranks on Instruments, where the rule-reward variant remains slightly more aggressive, and the deepest cutoff on Arts and Games, where TIGER still performs best. These residual gaps do not contradict the method. Instead, they indicate that reachability-aware hinting mainly improves branch stability and suffix optimization under the correct semantic prefix, which naturally favors overall ranking quality and broad coverage more than every extreme operating point. This is why the empirical profile of HCGRec is most convincing as a robust generative recommender rather than as a specialized optimizer for a single cutoff.

4.3 RQ2: Deep Analysis for Prefix Hint

RQ2 addresses two questions about prefix hints. The first question is whether hinting genuinely recovers useful training signal in group-relative post-training. The second question is whether the proposed offline minimal-hint policy is preferable to dynamic hinting, which computes hint depth during training.

We quantify the first question with the **zero-gradient group ratio**, that is, the fraction of training instances whose sampled rollout group satisfies $\text{Var}(\{r_j\}_{j=1}^G) = 0$, where G denotes the rollout-group size. Such instances yield identical group rewards and therefore do not provide useful group-relative learning signal. As shown in Figure 2, the unhinted post-training baseline leaves a large fraction of rollout groups in this inactive regime: the smoothed ratio ends around 0.55 on Arts and 0.63 on Instruments. Prefix hinting reduces the corresponding end-of-training ratios to about 0.12 and 0.17, respectively, by moving hard instances onto a reachable semantic branch before suffix rollout. This observation directly supports the central claim that prefix hints are not a superficial input modification. They convert many otherwise inactive samples into active reward-bearing training cases.

For the second question, we compare the proposed strategy with dynamic hinting using the cross-dataset visualization in Figure 4. **Ours** denotes the reachability-aware minimal prefix hint from Section 3.3: the hint length is selected once by SFT-checkpoint diagnosis and then fixed during reward-based post-training. Dynamic Hint instead recomputes the hint depth online during training.

The task-level depth distributions explain why prefix hinting does not affect every RL task equally. Figure 3 shows that the offline hint policy is distributed very differently across tasks. In both datasets, *title-seqrec* places much more mass on deeper hints than *SeqRec*, whereas *title/desc2index* is dominated by shallow cases. This means that the gain from prefix hinting is not only dataset-dependent but also task-dependent: some RL tasks already become

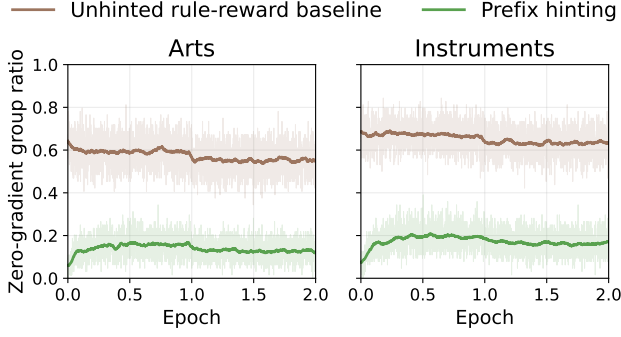


Figure 2: Fraction of rollout groups without gradient signal during training. The y-axis reports the proportion of sampled groups with zero reward variance, where lower values indicate fewer inactive samples.

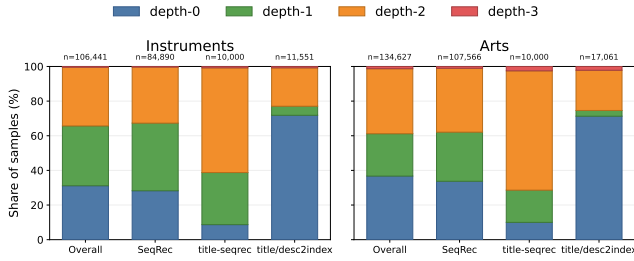


Figure 3: Task-wise fixed-hint depth distributions on Instruments and Arts. Each bar reports the relative depth composition of one RL task and of the overall RL mixture.

reachable after shallow correction, while others rely much more heavily on deeper semantic anchoring before suffix optimization becomes informative.

The main advantage of the offline minimal-hint policy is that it keeps optimization focused on a stable target. Figure 4 shows that dynamic hinting can remain competitive and even slightly better on a few top-heavy metrics, but the offline hint strategy is stronger on most displayed metrics once the comparison is extended to broader cutoffs and multiple datasets. A natural interpretation is that dynamic hinting keeps redirecting training toward whichever hard cases appear most difficult at the current stage. This can preserve isolated short-range gains, but it also makes the optimization objective drift with the model and increases the risk of over-adapting to unstable hard examples. By contrast, the offline policy fixes the semantic decision once and then optimizes the suffix under that fixed branch, which produces a more consistent training signal at lower rollout cost.

4.4 RQ3: Ablation Study

We further examine task scope and loss decomposition on Instruments.

Under the offline minimal-hint setting with prefix anchoring, the full RL task setting becomes the strongest option.

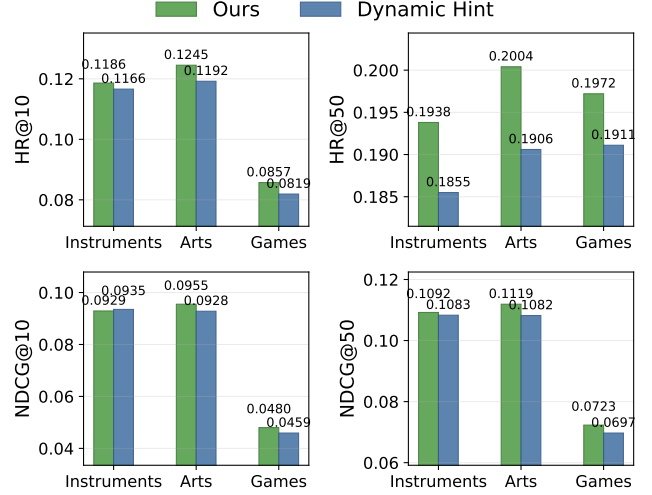


Figure 4: Hint-strategy comparison on Instruments, Arts, and Games. Each panel corresponds to one metric, and each panel reports grouped bars for the three datasets under the same readout rule.

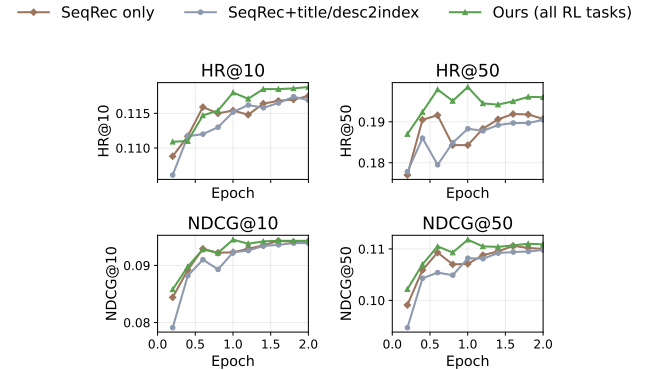


Figure 5: Task-scope trajectories on Instruments under the offline minimal-hint setting with prefix anchoring.

The full task setting, which combines *SeqRec*, *title-seqrec*, and *title+desc2index*, achieves the best HR@10 (0.1180), HR@50 (0.1985), NDCG@10 (0.0945), and NDCG@50 (0.1118), while remaining competitive on the other reported metrics. The two-task variant, *SeqRec + title+desc2index*, is generally weaker than the other two settings, with only a negligible advantage over SeqRec-only on HR@10.

Figure 5 confirms the same training pattern. Across HR@10, HR@50, NDCG@10, and NDCG@50, the full-task line reaches the strongest final checkpoint values, whereas the two-task variant remains the weakest trajectory for most of the run.

The task-scope ablation suggests that the full RL task setting provides better generalization than narrower task subsets. The key point is not only that the full-task setting gives the strongest endpoint results, but also that the additional task coverage improves how consistently the model uses the hinted semantic

branch. Once hinting restores reachability, broader supervision helps the model preserve that benefit across different recommendation conditions instead of specializing too strongly to a single task view. In this sense, the main contribution of the full-task setting is better generalization rather than a narrow gain on one isolated metric.

The task-scope curves reinforce the same interpretation. The advantage of the full-task setting is not a single late checkpoint effect; it remains visible across the training trajectory on the headline metrics. This behavior indicates that the extra tasks do not merely add optimization noise or task complexity. Instead, they act as a useful regularizer that improves the generalization of the hinted model and makes the resulting training dynamics more stable.

For loss decomposition, we compare suffix-only GRPO, full-sequence SFT + GRPO, and our prefix-SFT + suffix-only GRPO objective under the same checkpoint-selection rule.

Ours (prefix SFT + suffix-only GRPO) remains the strongest loss design. It achieves the best HR@5 (0.1009), HR@50 (0.1985), NDCG@5 (0.0890), and NDCG@50 (0.1118), which indicates that a limited amount of supervision on the hinted prefix is sufficient to improve semantic alignment before reward-based optimization acts on the suffix.

The comparison suggests that too much SFT is not beneficial. Full-sequence SFT + GRPO does not improve over the prefix-only design, which means that extending supervised pressure to the entire sequence does not help once the key alignment problem has already been resolved at the prefix level. The more effective strategy is to apply SFT only to the hinted prefix, where semantic alignment is needed, and then let suffix-level policy optimization focus on the sampled continuation.

The main gain therefore comes from aligning only the part of the sequence that actually needs alignment. The hinted prefix serves as semantic context and should be stabilized by supervised learning, whereas the suffix should remain the target of policy optimization. The strongest result comes precisely from preserving this division, rather than from increasing the amount of supervision everywhere in the sequence.

4.5 RQ4: Effect of Hint-Aware Credit Decomposition Weight

We finally study the weight λ of hint-aware credit decomposition in Eq. (17) on Instruments and Arts. This coefficient controls how strongly the hinted prefix is anchored with supervised semantic credit relative to suffix-only GRPO optimization.

RQ4 shows that hint-aware credit decomposition is useful, but only as a calibrated bias. Figure 6 shows that increasing λ does not monotonically improve performance. Moderate values around 0.001–0.01 produce the strongest or near-strongest results on most metrics, while the overly large setting $\lambda = 0.1$ consistently weakens recommendation quality. This behavior is exactly the tradeoff implied by Eq. (17): the hinted prefix should receive enough supervised anchoring pressure to preserve semantic stability, but not so much that suffix-level reward optimization is overwhelmed.

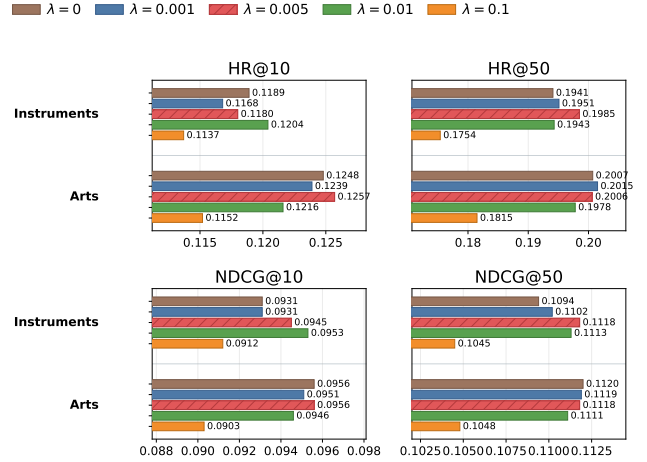


Figure 6: RQ4 credit-weight comparison on Instruments and Arts. Each panel reports one ranking metric, and each dataset group compares $\lambda = 0, 0.001, 0.005, 0.01$, and 0.1 . The hatched bar indicates the default HCGRec setting used in the main comparison.

This makes λ a calibration knob rather than a scaling knob. The goal is not to maximize supervised prefix anchoring in isolation, but to calibrate the balance between semantic stability and suffix discrimination. The pattern in Figure 6 therefore supports the broader claim of the paper: what matters is not simply to add more supervision after hinting, but to add the right amount in a way that preserves the division of labor between prefix and suffix optimization.

5 Conclusion

This paper studies reward-based post-training for semantic-ID generative recommendation. We identify finite-rollout reward unreachability as a central failure mode: although the target item exists in the semantic-token action space, the current generator often fails to sample any reward-distinguishable completion within a finite rollout group, causing group-relative optimization to receive zero useful advantage. To address this problem, we propose HCGRec, which performs offline reachability diagnosis with an SFT checkpoint and supplies the shortest target-prefix hint that makes hard instances reachable under the diagnostic rollout budget. The model then generates only the suffix under the hinted semantic branch, recovering reward-bearing comparisons without changing the inference interface.

We further show that hinting changes the credit assignment problem. Hinted prefix tokens are oracle-provided item context rather than sampled actions, whereas suffix tokens are the model’s actual generated actions. HCGRec therefore uses hint-aware credit decomposition: supervised prefix anchoring preserves item semantics and coarse-to-fine Semantic ID structure, while suffix-only GRPO optimizes the sampled generation behavior. Experiments on three Amazon domains show that HCGRec improves over conventional sequential recommenders and is competitive with semantic-ID generative baselines on many key recommendation metrics, while

some cutoffs remain stronger for specialized baselines. The analysis further shows that prefix hints substantially reduce inactive zero-variance rollout groups, the offline minimal hint is generally more stable and cheaper than dynamic hinting, and the decomposition weight should remain moderate to balance semantic anchoring and reward-driven suffix optimization.

Several directions remain open. First, the current implementation relies on an offline diagnostic pass, and more efficient reachability estimation may further reduce post-training cost. Second, our experiments focus on a fixed semantic index with short four-token identifiers; extending the analysis to longer or hierarchical identifiers may reveal different hint-budget trade-offs. Finally, combining reachability-aware hinting with richer recommendation rewards, such as calibrated ranking rewards or user-feedback rewards, is a promising path toward more robust generative recommendation post-training.

GenAI Usage Disclosure

Generative AI tools were used to assist with manuscript drafting and language editing. All technical claims, experimental results, citations, and final text are reviewed and verified by the authors.

Acknowledgments

The work is supported by National Natural Science Foundation of China (62502310).

References

- [1] 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. arXiv:2305.00447 [cs.IR] doi:10.1145/3604915.3608857
- [2] Jiaxin Deng, Shiyao Wang, Kuo Cai, Lejian Ren, Qigen Hu, Weifeng Ding, Qiang Luo, and Guorui Zhou. 2025. OneRec: Unifying Retrieve and Rank with Generative Recommender and Iterative Preference Alignment. arXiv:2502.18965 [cs.IR] https://arxiv.org/abs/2502.18965
- [3] Zezhong Fan, Ziheng Chen, Luyi Ma, Jin Huang, Lalitesh Morishetti, Kaushiki Nag, Sushant Kumar, and Kannan Achan. 2026. CRAB: Codebook Rebalancing for Bias Mitigation in Generative Recommendation. arXiv:2604.05113 [cs.IR] https://arxiv.org/abs/2604.05113
- [4] Dengzhao Fang, Jingtong Gao, Chengcheng Zhu, Yu Li, Xiangyu Zhao, and Yi Chang. 2025. HiD-VAE: Interpretable Generative Recommendation via Hierarchical and Disentangled Semantic IDs. arXiv:2508.04618 [cs.IR] https://arxiv.org/abs/2508.04618
- [5] Junchen Fu, Xuri Ge, Alexandros Karatzoglou, Ioannis Arapakis, Suzan Verberne, Joemon M. Jose, and Zhaochun Ren. 2026. Differentiable Semantic ID for Generative Recommendation. arXiv:2601.19711 [cs.IR] https://arxiv.org/abs/2601.19711
- [6] Vianne R. Gao, Chen Xue, Marc Versage, Xie Zhou, Zhongruo Wang, Chao Li, Yeon Seonwoo, Nan Chen, Zhen Ge, Gourab Kundu, Weiqi Zhang, Tian Wang, Qingjun Cui, and Trishul Chilimbi. 2025. SynerGen: Contextualized Generative Recommender for Unified Search and Recommendation. arXiv:2509.21777 [cs.CL] https://arxiv.org/abs/2509.21777
- [7] Shijie Geng, Shuchang Liu, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. 2023. Recommendation as Language Processing (RLP): A Unified Pretrain, Personalized Prompt & Predict Paradigm (P5). arXiv:2203.13366 [cs.IR] https://arxiv.org/abs/2203.13366
- [8] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2016. Session-based Recommendations with Recurrent Neural Networks. arXiv:1511.06939 [cs.LG] https://arxiv.org/abs/1511.06939
- [9] Jianchao Ji, Zelong Li, Shuyuan Xu, Wenyue Hua, Yingqiang Ge, Juntao Tan, and Yongfeng Zhang. 2023. GenRec: Large Language Model for Generative Recommendation. arXiv:2307.00457 [cs.IR] https://arxiv.org/abs/2307.00457
- [10] Jie Jiang, Yangru Huang, Zeyu Wang, Changping Wang, Yuling Xiong, Jun Zhang, and Huan Yu. 2026. Spend Search Where It Pays: Value-Guided Structured Sampling and Optimization for Generative Recommendation. arXiv:2602.10699 [cs.AI] https://arxiv.org/abs/2602.10699
- [11] Clark Mingxuan Ju, Liam Collins, Leonardo Neves, Bhuvish Kumar, Louis Yufeng Wang, Tong Zhao, and Neil Shah. 2025. Generative Recommendation with Semantic IDs: A Practitioner’s Handbook. arXiv:2507.22224 [cs.IR] https://arxiv.org/abs/2507.22224
- [12] Wang-Cheng Kang and Julian McAuley. 2018. Self-Attentive Sequential Recommendation. arXiv:1808.09781 [cs.IR] https://arxiv.org/abs/1808.09781
- [13] Kirill Khrylchenko. 2026. Variable-Length Semantic IDs for Recommender Systems. arXiv:2602.16375 [cs.IR] https://arxiv.org/abs/2602.16375
- [14] Xiaoyu Kong, Leheng Sheng, Junfei Tan, Yuxin Chen, Jiancan Wu, An Zhang, Xiang Wang, and Xiangnan He. 2025. MiniOneRec: An Open-Source Framework for Scaling Generative Recommendation. arXiv:2510.24431 [cs.IR] https://arxiv.org/abs/2510.24431
- [15] Yifan Liu, Kangning Zhang, Xiangyuan Ren, Yanhua Huang, Jiarui Jin, Yingjie Qin, Ruilong Su, Ruiwen Xu, Yong Yu, and Weinan Zhang. 2024. AlignRec: Aligning and Training in Multimodal Recommendations. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management. Association for Computing Machinery, New York, NY, USA, 1503–1512*. doi:10.1145/3627673.3679626
- [16] Jianmo Ni, Jiacheng Li, and Julian McAuley. 2019. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. 188–197.
- [17] Gustavo Penha, Edoardo D’Amico, Marco De Nadai, Enrico Palumbo, Alexandre Tamborrino, Ali Vardasbi, Max Lefarov, Shawn Lin, Timothy Heath, Francesco Fabbri, and Hugues Bouchard. 2025. Semantic IDs for Joint Generative Search and Recommendation. arXiv:2508.10478 [cs.IR] https://arxiv.org/abs/2508.10478
- [18] Gustavo Penha, Ali Vardasbi, Enrico Palumbo, Marco de Nadai, and Hugues Bouchard. 2024. Bridging Search and Recommendation in Generative Retrieval: Does One Task Help the Other? arXiv:2410.16823 [cs.IR] https://arxiv.org/abs/2410.16823
- [19] Shashank Rajput, Nikhil Mehta, Anima Singh, Raghunandan H. Keshavan, Trung Vu, Lukasz Heldt, Lichan Hong, Yi Tay, Vinh Q. Tran, Jonah Samost, Maciej Kula, Ed H. Chi, and Maheswaran Sathiamoorthy. 2023. Recommender Systems with Generative Retrieval. arXiv:2305.05065 [cs.IR] https://arxiv.org/abs/2305.05065
- [20] Shuai Shao, Kangning Zhang, Qingyao Li, Shijian Wang, Hao Wang, Wenxiang Jiao, Yuan Lu, Yi Guo, Weiwen Liu, and Weinan Zhang. 2026. Harness-R1: Learning to Edit Executable Runtime Harnesses from Agent Failure Trajectories. arXiv:2608.02276 [cs.AI] doi:10.48550/arXiv.2608.02276
- [21] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. DeepSeek-Math: Pushing the Limits of Mathematical Reasoning in Open Language Models. arXiv:2402.03300 [cs.CL] https://arxiv.org/abs/2402.03300
- [22] Zihua Si, Zhongxiang Sun, Jiale Chen, Guozhang Chen, Xiaoxue Zang, Kai Zheng, Yang Song, Xiao Zhang, Jun Xu, and Kun Gai. 2024. Generative Retrieval with Semantic Tree-Structured Item Identifiers via Contrastive Learning. arXiv:2309.13375 [cs.IR] https://arxiv.org/abs/2309.13375
- [23] 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. arXiv:1904.06690 [cs.IR] https://arxiv.org/abs/1904.06690
- [24] Junfei Tan, Yuxin Chen, An Zhang, Junguang Jiang, Bin Liu, Ziru Xu, Han Zhu, Jian Xu, Bo Zheng, and Xiang Wang. 2025. Reinforced Preference Optimization for Recommendation. arXiv:2510.12211 [cs.IR] https://arxiv.org/abs/2510.12211
- [25] Jiaxi Tang and Ke Wang. 2018. Personalized Top-N Sequential Recommendation via Convolutional Sequence Embedding. arXiv:1809.07426 [cs.IR] https://arxiv.org/abs/1809.07426
- [26] Yibiao Wei, Jie Zou, Pengfei Zhang, Xiao Ao, Weikang Guo, Zeyu Ma, and Yang Yang. 2026. CARD: Non-Uniform Quantization of Visual Semantic Unit for Generative Recommendation. arXiv:2604.26427 [cs.IR] https://arxiv.org/abs/2604.26427
- [27] Ming Xia, Zhiqin Zhou, Guoxin Ma, and Dongmin Huang. 2026. Unleash the Potential of Long Semantic IDs for Generative Recommendation. arXiv:2602.13573 [cs.IR] https://arxiv.org/abs/2602.13573
- [28] Yu Xia, Canwen Xu, Zhewei Yao, Julian McAuley, and Yuxiong He. 2026. Learning to Hint for Reinforcement Learning. arXiv:2604.00698 [cs.LG] https://arxiv.org/abs/2604.00698
- [29] Haibo Xing, Hao Deng, Lingyu Mu, Jinxin Hu, Yu Zhang, Xiaoyi Zeng, and Jing Zhang. 2026. Learning to Reflect and Correct: Towards Better Decoding Trajectories for Large-Scale Generative Recommendation. arXiv:2602.23639 [cs.IR] https://arxiv.org/abs/2602.23639
- [30] Liu Yang, Fabian Paischer, Kaveh Hassani, Jiacheng Li, Shuai Shao, Zhang Gabriel Li, Yun He, Xue Feng, Nima Noorshams, Sem Park, Bo Long, Robert D Nowak, Xiaoli Gao, and Hamid Eghbalzadeh. 2024. Unifying Generative and Dense Retrieval for Sequential Recommendation. arXiv:2411.18814 [cs.IR] https://arxiv.org/abs/2411.18814
- [31] Kangning Zhang, Jiarui Jin, Yingjie Qin, Ruilong Su, Jianghao Lin, Yong Yu, and Weinan Zhang. 2024. MOTOR: Learning ID-free Item Representation with Token Crossing for Embedding-based Multimodal Recommendation. arXiv:2410.19276 [cs.IR] doi:10.48550/arXiv.2410.19276

- [32] Kangning Zhang, Yixing Li, Shuai Shao, Qingyao Li, Zhengxi Lu, Zhiyuan Yao, Jianghao Lin, Wenxiang Jiao, Yuan Lu, Weiwen Liu, Weinan Zhang, and Yong Yu. 2026. VAD: Attributing Visual Evidence for Target Reconstruction in Multimodal On-Policy Distillation. arXiv:2607.28590 [cs.CV] doi:10.48550/arXiv.2607.28590
- [33] Kangning Zhang, Weiwen Liu, Wenxiang Jiao, Kounianhua Du, Yuan Lu, Weinan Zhang, and Yong Yu. 2026. LoopTool: Closing the Data-Training Loop for Robust LLM Tool Calls. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, San Diego, California, United States, 21144–21164. doi:10.18653/v1/2026.acl-long.968
- [34] Kangning Zhang, Yingjie Qin, Weinan Zhang, Yong Yu, and Jianghao Lin. 2026. DiffCold: A Diffusion-based Generative Model for Cold-Start Item Recommendation. arXiv:2606.12245 [cs.IR] doi:10.48550/arXiv.2606.12245
- [35] Kangning Zhang, Shuai Shao, Qingyao Li, Jianghao Lin, Lingyue Fu, Shijian Wang, Wenxiang Jiao, Yuan Lu, Weiwen Liu, Weinan Zhang, and Yong Yu. 2026. MMSkills: Towards Multimodal Skills for General Visual Agents. arXiv:2605.13527 [cs.AI] doi:10.48550/arXiv.2605.13527
- [36] Xichen Zhang, Sitong Wu, Yinghao Zhu, Haoru Tan, Shaozuo Yu, Ziyi He, and Jiaya Jia. 2026. Scaf-GRPO: Scaffolded Group Relative Policy Optimization for Enhancing LLM Reasoning. arXiv:2510.19807 [cs.CL] https://arxiv.org/abs/2510.19807
- [37] Bowen Zheng, Yupeng Hou, Hongyu Lu, Yu Chen, Wayne Xin Zhao, Ming Chen, and Ji-Rong Wen. 2023. Adapting Large Language Models by Integrating Collaborative Semantics for Recommendation. arXiv:2311.09049 [cs.IR] https://arxiv.org/abs/2311.09049
- [38] Bowen Zheng, Enze Liu, Zhongfu Chen, Zhongrui Ma, Yue Wang, Wayne Xin Zhao, and Ji-Rong Wen. 2025. Pre-training Generative Recommender with Multi-Identifier Item Tokenization. arXiv:2504.04400 [cs.IR] https://arxiv.org/abs/2504.04400
- [39] Guorui Zhou, Jiaxin Deng, Jinghao Zhang, Kuo Cai, Lejian Ren, Qiang Luo, Qianqian Wang, Qigen Hu, Rui Huang, Shiyao Wang, Weifeng Ding, Wuchao Li, Xinchun Luo, Xingmei Wang, Zexuan Cheng, Zixing Zhang, Bin Zhang, Boxuan Wang, Chaoyi Ma, Chengru Song, Chenhui Wang, Di Wang, Dongxue Meng, Fan Yang, Fangyu Zhang, Feng Jiang, Fuxing Zhang, Gang Wang, Guowang Zhang, Han Li, Hengrui Hu, Hezheng Lin, Hongtao Cheng, Hongyang Cao, Huanjie Wang, Jiaming Huang, Jiapeng Chen, Jiaqiang Liu, Jinghui Jia, Kun Gai, Lantao Hu, Liang Zeng, Liao Yu, Qiang Wang, Qidong Zhou, Shengzhe Wang, Shihui He, Shuang Yang, Shujie Yang, Sui Huang, Tao Wu, Tiantian He, Tingting Gao, Wei Yuan, Xiao Liang, Xiaoxiao Xu, Xugang Liu, Yan Wang, Yi Wang, Yiwu Liu, Yue Song, Yufei Zhang, Yunfan Wu, Yurfeng Zhao, and Zhanyu Liu. 2025. OneRec Technical Report. arXiv:2506.13695 [cs.IR] https://arxiv.org/abs/2506.13695
- [40] Guorui Zhou, Hengrui Hu, Hongtao Cheng, Huanjie Wang, Jiaxin Deng, Jinghao Zhang, Kuo Cai, Lejian Ren, Lu Ren, Liao Yu, Pengfei Zheng, Qiang Luo, Qianqian Wang, Qigen Hu, Rui Huang, Ruiming Tang, Shiyao Wang, Shujie Yang, Tao Wu, Wuchao Li, Xinchun Luo, Xingmei Wang, Yi Su, Yunfan Wu, Zexuan Cheng, Zhanyu Liu, Zixing Zhang, Bin Zhang, Boxuan Wang, Chaoyi Ma, Chengru Song, Chenhui Wang, Chenglong Chu, Di Wang, Dongxue Meng, Dunju Zang, Fan Yang, Fangyu Zhang, Feng Jiang, Fuxing Zhang, Gang Wang, Guowang Zhang, Han Li, Honghui Bao, Hongyang Cao, Jiaming Huang, Jiapeng Chen, Jiaqiang Liu, Jinghui Jia, Kun Gai, Lantao Hu, Liang Zeng, Qiang Wang, Qidong Zhou, Rongzhou Zhang, Shengzhe Wang, Shihui He, Shuang Yang, Siyang Mao, Sui Huang, Tiantian He, Tingting Gao, Wei Yuan, Xiao Liang, Xiaoxiao Xu, Xugang Liu, Yan Wang, Yang Zhou, Yi Wang, Yiwu Liu, Yue Song, Yufei Zhang, Yurfeng Zhao, Zhixin Ling, and Ziming Li. 2025. OneRec-V2 Technical Report. arXiv:2508.20900 [cs.IR] https://arxiv.org/abs/2508.20900
- [41] Jieming Zhu, Mengqun Jin, Qijiong Liu, Zexuan Qiu, Zhenhua Dong, and Xiu Li. 2024. CoST: Contrastive Quantization based Semantic Tokenization for Generative Recommendation. arXiv:2404.14774 [cs.IR] https://arxiv.org/abs/2404.14774
- [42] Yaochen Zhu, Harald Steck, Dawen Liang, Yinhan He, Vito Ostuni, Jundong Li, and Nathan Kallus. 2026. Rank-GRPO: Training LLM-based Conversational Recommender Systems with Reinforcement Learning. arXiv:2510.20150 [cs.IR] https://arxiv.org/abs/2510.20150
- [43] Yanyan Zou, Junbo Qi, Lunsong Huang, Yu Li, Kewei Xu, Jiabao Gao, Binglei Zhao, Xuanhua Yang, Sulong Xu, and Shengjie Li. 2026. GenRec: A Preference-Oriented Generative Framework for Large-Scale Recommendation. arXiv:2604.14878 [cs.IR] https://arxiv.org/abs/2604.14878