DG-HCA: DUAL-GRANULARITY HIERARCHICAL CREDIT ASSIGNMENT FOR GENERATIVE RECOMMENDATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Generative recommendation with explicit reasoning typically relies on sequence-level reinforcement learning, where a single outcome-level advantage is broadcast to all tokens in a rollout. However, for hierarchical Semantic IDs (SIDs), such coarse-grained supervision can induce conflicting credit signals: incorrect suffix decisions may be reinforced by positive advantages, while correct prefix decisions may be suppressed by negative ones. To address this issue, we propose **DG-HCA**, a Dual-Granularity Hierarchical Credit Assignment framework for reasoning-based generative recommendation. DG-HCA grounds policy optimization in retrieval feedback from Constrained Beam Search and decomposes credit at two complementary granularities. At the reasoning-trajectory level, Ranking-Driven Reasoning Credit evaluates trajectories according to their retrieval quality. At the SID-action level, Target-Path SID Credit performs relative credit assignment over hierarchical decisions along the ground-truth path, while Target-Path Recovery provides additional supervision when Beam Search deviates from the target SID. Extensive experiments on three public Amazon datasets demonstrate that DG-HCA achieves competitive recommendation performance and obtains the best NDCG@10 on all three datasets. Ablation studies further verify the effectiveness of SID-level credit assignment and Target-Path Recovery. These results highlight the importance of aligning reinforcement-learning credit with the structured generation process in reasoning-based generative recommendation.

1 INTRODUCTION

In recent years, with the continual improvement in the semantic understanding and reasoning capabilities (Zhou et al., 2025; Liu et al., 2025b; He et al., 2026; Team et al., 2026; Wei et al., 2022) of small-scale foundation models, Generative Recommendation (Wang et al., 2023; Bao et al., 2023b) has emerged as an important research direction in recommender systems. Its core idea is to quantize the semantic information of items into compact sequences of discrete identifiers, referred to as Semantic IDs (SIDs) (Hou et al., 2025), and to fine-tune language models to capture the semantics encoded by these discrete tokens (Rajput et al., 2023; Wang et al., 2024a). In this way, conventional recommendation tasks can be unified under an autoregressive generation framework over the SID space. Compared with traditional retrieval-and-ranking paradigms (Tang & Wang, 2018a; Hidasi et al., 2015; Kang & McAuley, 2018), Generative Recommendation provides a more direct mechanism for leveraging the semantic knowledge and generalization capabilities already acquired by foundation models.

Existing approaches (Zhou et al., 2025; Yang et al., 2026; Tang et al., 2026; Zheng et al., 2024; Wang et al., 2024b) can be broadly categorized into two paradigms according to their inference processes. The first directly generates the SID of a candidate item, whereas the second first produces a reasoning trajectory conditioned on the user's historical interactions and subsequently generates the target SID. The latter can be further divided into **Explicit Textual Reasoning** and **Latent Reasoning**. Explicit Textual Reasoning provides better interpretability and controllability, but typically incurs a substantially longer generation process. In contrast, Latent Reasoning performs reasoning within the hidden representation space (Zheng et al., 2026a; Zhang et al., 2026b), resulting in more compact reasoning trajectories, while making the reasoning process difficult to interpret or explicitly

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

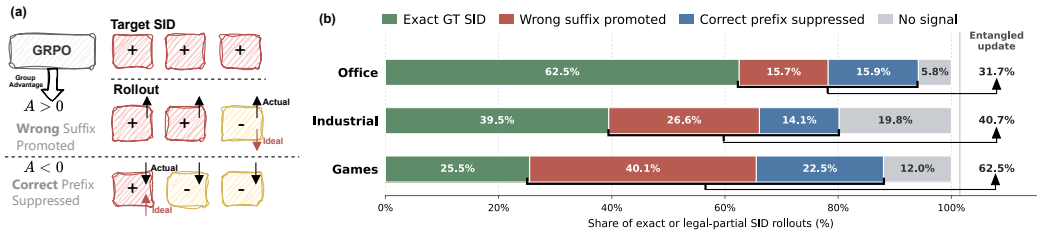


Figure 1: **(a)** A detailed illustration of the Credit Assignment Conflicts phenomenon. **(b)** The prevalence of Credit Assignment Conflicts across three open-source datasets. Notably, Office exhibits a lower incidence of Credit Assignment Conflicts, and this trend is consistent with the model's relative recommendation performance across the three datasets reported in the experimental results.

control. In this work, we primarily focus on Generative Recommendation under the Explicit Textual Reasoning paradigm.

Current Generative Recommendation methods (He et al., 2026; Kong et al., 2025) typically adopt a two-stage post-training pipeline. In the first stage, Supervised Fine-Tuning (SFT) is employed to align the SID space with the natural-language representation space while distilling knowledge from a teacher model. In the second stage, Reinforcement Learning (RL) is further applied to optimize recommendation performance, among which Group Relative Policy Optimization (GRPO) (Shao et al., 2024) has been widely adopted. Although GRPO has demonstrated strong performance in tasks such as mathematical reasoning (Ahn et al., 2024) and code generation (Jiang et al., 2026b; Wang & Chen, 2023), its application to Generative Recommendation still faces two prominent challenges (Zheng et al., 2026b; Yu et al., 2026). First, recommendation rewards are typically sparse, causing a large fraction of rollouts within the same group to receive identical rewards or even zero-valued advantages, which can lead to unstable optimization, particularly in the later stages of training. Second, GRPO broadcasts an outcome-level advantage uniformly to the entire generated sequence, making it unable to distinguish the actual contribution of individual tokens to the final recommendation outcome.

This limitation becomes particularly pronounced for hierarchical SIDs. Consider a target SID represented as $(+, +, +)$. Suppose that, within the same group, a wrong-suffix sample $(+, +, -)$ receives a positive advantage, whereas a correct-prefix sample $(+, -, -)$ receives a negative advantage. Under sequence-level credit assignment, these two cases induce two types of conflicting optimization signals. For the wrong-suffix sample, the erroneous token at the final hierarchy level is incorrectly reinforced by the positive advantage. Conversely, for the correct-prefix sample, the correctly generated token at the first hierarchy level is undesirably suppressed by the negative advantage. To quantify this phenomenon, we randomly sample 5k training instances from each of three Amazon public datasets (Ni et al., 2019)—Video Games, Office Products, and Industrial and Scientific—and analyze the advantages observed during training. Our empirical results reveal a substantial number of such **Credit Assignment Conflicts**, suggesting that sequence-level advantages provide inaccurate supervision for token-level decisions in hierarchical SID generation.

2 Related Work

Semantic ID-based Generative Recommendation. Generative recommendation reformulates the conventional candidate retrieval and ranking pipeline as an autoregressive sequence generation problem. In this paradigm, Semantic IDs represent items as compact, hierarchical sequences of discrete tokens, providing an effective interface that bridges the semantic understanding capabilities of language models and the discrete item space. TIGER (Rajput et al., 2023) systematically introduced hierarchical SIDs into generative retrieval, while subsequent methods such as LETTER (Wang et al., 2024a) and ETEGRec (Liu et al., 2025a) further enhanced the suitability of SIDs for recommendation by improving semantic–collaborative information integration and enabling end-to-end item tokenization. OneRec (Deng et al., 2025) unifies retrieval and ranking within a single generative model and further improves generation quality through preference alignment. MiniOneRec (Kong et al., 2025), in turn, establishes an open-source training framework that covers SID construction, su-

pervised fine-tuning, and recommendation-oriented reinforcement learning (RL). Building on these advances, OneRec-Think (Liu et al., 2025b), SIDReasoner (He et al., 2026), and related approaches further incorporate explicit reasoning into SID generation. Through semantic alignment, reasoning activation, and reinforcement learning, these methods enable the model to explicitly model user interests and preferences before generating the target item.

Reinforcement Learning for Generative Recommendation. As generative recommendation advances from supervised learning to the post-training stage, GRPO and its variants have increasingly been adopted to optimize recommendation objectives that are difficult to capture directly through maximum-likelihood estimation. MiniOneRec combines constrained decoding with hybrid rewards for recommendation-oriented reinforcement learning (RL), while SIDReasoner employs outcome-driven GRPO to optimize recommendation reasoning trajectories based on the final SID prediction outcomes. To better accommodate the structured outputs inherent to recommendation tasks, Rank-GRPO (Zhu et al., 2026) treats individual ranks within a recommendation list as optimization units, whereas OneRec-Think introduces beam-based feedback to make reinforcement learning signals more closely aligned with Top-K recommendation performance. More recent studies have begun to explicitly account for the hierarchical structure of SIDs. SAPO (Zheng et al., 2026b) groups each reasoning block with its corresponding SID token into a single optimization step; Sibling-GRPO (Jiang et al., 2026a) in V-STAR performs relative optimization over sibling actions that share the same parent node; and HCGRec (Zhang et al., 2026a) alleviates zero-advantage groups on challenging samples through target-prefix prompting. Overall, reinforcement learning for recommendation is progressively shifting from sequence-level outcome optimization toward fine-grained credit assignment tailored to ranked recommendation lists, hierarchical SID structures, and the underlying search process. Following this line of research, we further investigate credit alignment among reasoning trajectories, hierarchical SID decisions, and feedback from actual retrieval outcomes.

3 METHODOLOGY

3.1 FRAMEWORK OVERVIEW

Our training framework consists of three stages, with the optimization objective progressively transitioning from SID–Language Alignment and Recommendation Reasoning-Mode Guidance to Fine-Grained Hierarchical Credit Assignment. Let an SID consist of D hierarchical tokens, denoted as $y = (y_1, \ldots, y_D)$. The method described below is applicable to SIDs of arbitrary depth $D \geq 3$.

Stage 1: SID–Language Alignment. We employ RQ-VAE (Lee et al., 2022) to discretize item semantic representations into hierarchical SIDs. We then establish fine-grained alignment between SID tokens and the language space through recommendation-oriented multi-task supervised fine-tuning (SFT) and teacher-driven knowledge augmentation.

Stage 2: Reasoning-Mode Guidance. After SID–language alignment, we apply GRPO to initially reinforce the recommendation paradigm of reasoning before generating the target SID, thereby encouraging the model to develop stable and coherent recommendation reasoning trajectories.

Stage 3: Dual-Granularity Hierarchical Credit Assignment (DG-HCA). We employ constrained Beam Search consistent with the inference procedure used at test time to obtain retrieval-grounded feedback, and perform credit assignment at two complementary granularities: the reasoning trajectory and hierarchical SID actions.

3.2 SID-LANGUAGE ALIGNMENT

Given the textual metadata t_i of item i, we first encode it into a continuous semantic representation $\mathbf{z}_i \in \mathbb{R}^{d_{\text{emb}}}$ using a pre-trained text encoder. RQ-VAE (Lee et al., 2022) then performs hierarchical residual quantization with D codebooks, where $\mathcal{C}^{(d)} = \{\mathbf{e}_m^{(d)}\}_{m=1}^{M}$ denotes the codebook at level d. Let $\mathbf{r}_i^{(0)} = \mathbf{z}_i$. The code assignment and residual update at level d are given by

$$c_i^{(d)} = \arg\min_m \left\| \mathbf{r}_i^{(d-1)} - \mathbf{e}_m^{(d)} \right\|_2^2, \qquad \mathbf{r}_i^{(d)} = \mathbf{r}_i^{(d-1)} - \mathbf{e}_{c_i^{(d)}}^{(d)}. \tag{1}$$

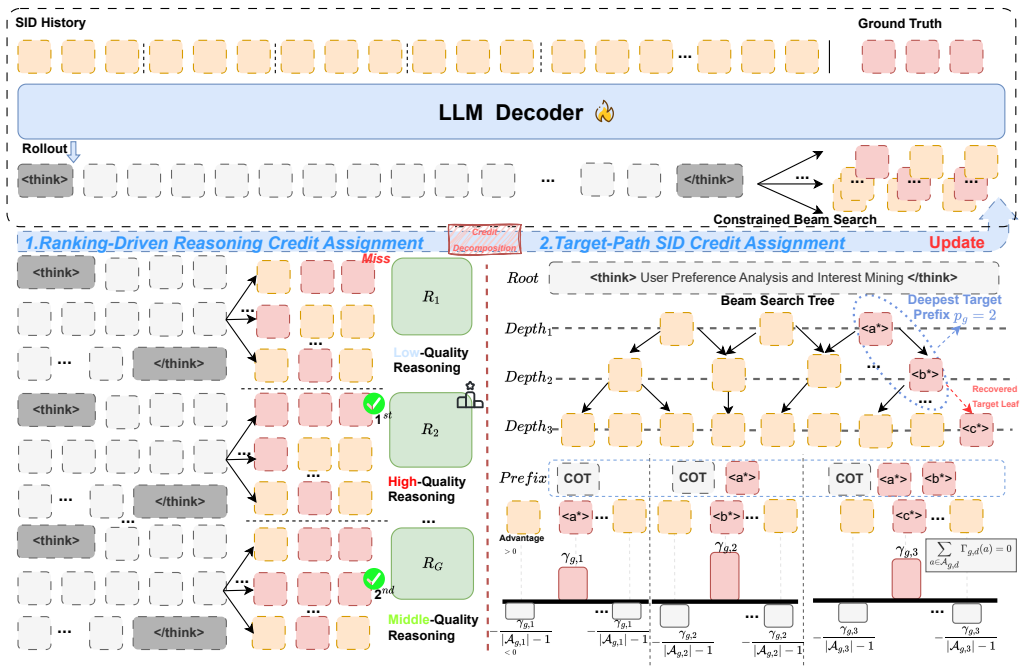


Figure 2: Overview of **DG-HCA**. Retrieval feedback from Constrained Beam Search is decomposed into Ranking-Driven Reasoning Credit for reasoning trajectories and Target-Path SID Credit for hierarchical SID decisions. Target-Path Recovery further provides supervision when Beam Search deviates from the ground-truth SID path.

Each code index $c_i^{(d)}$ is mapped to a level-specific SID token $y_i^{(d)}$, yielding $\mathrm{SID}(i) = (y_i^{(1)}, \ldots, y_i^{(D)})$, with reconstruction $\hat{\mathbf{z}}_i = \sum_{d=1}^{D} \mathbf{e}_{c_i^{(d)}}^{(d)} = \mathbf{z}_i - \mathbf{r}_i^{(D)}$. RQ-VAE is optimized by

$$\mathcal{L}_{\mathrm{RQ}} = \|\mathbf{z}_i - \hat{\mathbf{z}}_i\|_2^2 + \sum_{d=1}^{D} \Big[\| \mathrm{sg}[\mathbf{r}_i^{(d-1)}] - \mathbf{e}_{c_i^{(d)}}^{(d)} \|_2^2 + \beta_{\mathrm{com}} \|\mathbf{r}_i^{(d-1)} - \mathrm{sg}[\mathbf{e}_{c_i^{(d)}}^{(d)}]\|_2^2 \Big]. \tag{2}$$

where $\mathrm{sg}[\cdot]$ denotes the stop-gradient operator and $\beta_{\mathrm{com}} > 0$ controls the commitment loss. After training, the resulting SID tokens are added to the item-token vocabulary of the generative model, followed by recommendation-oriented multi-task supervised fine-tuning (SFT) and teacher-guided knowledge augmentation to further align the SID and language spaces.

3.3 Reasoning-Mode Guidance

Given a user history $\mathcal{H}$, the model first generates a reasoning trajectory τ and then predicts the target SID $y = (y_1, \ldots, y_D)$. For the same user history, the old policy $\pi_{\theta_{\mathrm{old}}}$ samples G rollouts, each denoted as $o_g = \tau_g \circ y_g$. Each rollout receives an outcome-level reward based on the ground-truth SID y^*, from which the group-normalized advantage $\hat{A}_g$ is computed.

The objective of Stage 2 is to establish a stable recommendation reasoning pattern through online outcome feedback. Accordingly, all tokens within the same rollout share the sequence-level advantage $\hat{A}_g$. The detailed reward formulation and GRPO objective are provided in Appendix D. However, this coarse-grained credit assignment cannot distinguish the contributions of a correct prefix from those of an incorrect suffix, motivating the hierarchical credit assignment introduced in Stage 3.

3.4 Dual-Granularity Hierarchical Credit Assignment

Let the recommendation training set be $\mathcal{D}_{\text{rec}} = \{(\mathcal{H}_n, y_n^*)\}_{n=1}^N$, where $y_n^* = (y_{n,1}^*, \ldots, y_{n,D}^*)$ denotes the target SID. We focus on a single training instance and omit the sample index n for simplicity. For the g-th reasoning trajectory sampled from the old policy, we perform Beam Search constrained to valid SIDs:

$$\tau_g \sim \pi_{\theta_{\text{old}}}(\cdot \mid \mathcal{H}), \qquad \mathcal{B}_g = \text{Beam}_B(\pi_{\theta_{\text{old}}}; \mathcal{H}, \tau_g) = (\tilde{y}_{g,1}, \ldots, \tilde{y}_{g,B_g}), \quad B_g \le B. \tag{3}$$

Here, $g \in \{1, \ldots, G\}$ indexes the sampled reasoning trajectories, while $j \in \{1, \ldots, B_g\}$ indexes the ranked Beam candidates. DG-HCA decomposes retrieval feedback into two complementary levels of credit: CoT Credit measures the retrieval quality associated with different reasoning trajectories, whereas SID Credit attributes credit to individual hierarchical decisions along the target SID path.

3.4.1 Ranking-Driven Reasoning Credit Assignment

Define the rank discount as $\omega(r) = \frac{1}{\log_2(r+1)}$. For a candidate $\tilde{y}$, let $m(\tilde{y}, y^*) = \max\{h \in \{0, \ldots, D\} : \tilde{y}_{1:h} = y_{1:h}^*\}$ denote the length of its longest common prefix with the target SID. For a non-exact match, we define the hierarchical prefix score as $\psi(\tilde{y}, y^*) = \mathbf{1}[m(\tilde{y}, y^*) > 0] 2^{m(\tilde{y}, y^*) - D}$.

If the target SID y^* appears in $\mathcal{B}_g$ at rank r_g, the reward for the g-th reasoning trajectory is

$$R_g = \begin{cases} R_{\text{inv}}, & \tau_g \text{ is invalid}, \\ \omega(r_g), & y^* \in \mathcal{B}_g, \\ \omega(B) \max\limits_{1 \le j \le B_g} \left[\psi(\tilde{y}_{g,j}, y^*)\omega(j)\right], & y^* \notin \mathcal{B}_g, \end{cases} \tag{4}$$

where $R_{\text{inv}} < 0$. Since any non-exact match satisfies $\psi(\tilde{y}, y^*) \le 1/2$, its prefix-based reward is strictly lower than that of an exact Top-B hit and therefore serves only as an auxiliary signal to alleviate reward sparsity. For the G trajectories sampled for the same user history, let μ_R and σ_R denote the mean and standard deviation of their rewards. The CoT advantage is then $A_g = \frac{(R_g - \mu_R)}{(\sigma_R + \varepsilon_R)}$ and is shared by all CoT tokens in τ_g.

3.4.2 Target-Path SID Credit Assignment

CoT Credit captures the overall quality of different reasoning trajectories but cannot localize hierarchical errors within an SID. To enable fine-grained credit assignment, we construct local action groups along the target path and perform relative optimization within each group. For the g-th trajectory, we define the deepest target-prefix depth covered by Beam Search as $p_g = \max\{h \in \{0, \ldots, D\} : \exists j, \tilde{y}_{g,j,1:h} = y_{1:h}^*\}$. For each $d \le p_g$, we construct a sibling action set only under the target parent state $y_{<d}^*$:

$$\mathcal{A}_{g,d} = \text{Unique}\left\{\tilde{y}_{g,j,d} : \tilde{y}_{g,j,<d} = y_{<d}^*\right\}, \qquad a_{g,d}^* = y_d^*. \tag{5}$$

Local groups containing fewer than two unique actions are excluded from relative optimization.

Target-Path Recovery. When $p_g < D$, Beam Search first departs from the target path at depth $d = p_g + 1$. We introduce the remaining target suffix as a recovery action $a_{g,d}^* = y_{d:D}^*$ and group it with the suffixes explored by Beam Search under the same target parent state:

$$\mathcal{A}_{g,d} = \{y_{d:D}^*\} \cup \text{Unique}\left\{\tilde{y}_{g,j,d:D} : \tilde{y}_{g,j,<d} = y_{<d}^*\right\}. \tag{6}$$

The recovery action is used only for policy optimization and is neither inserted back into the Beam nor allowed to alter the original retrieval results. If the resulting local group contains only the target action, the group is skipped.

Prefix Support. Beam evidence is used to scale the credit assigned to each action, while the sign of the credit is determined independently. For a target action already covered by the Beam, i.e., $d \le p_g$, let $r_{g,d} = \min\{j : \tilde{y}_{g,j,1:d} = y_{1:d}^*\}$ denote the highest-ranked occurrence of the target prefix $y_{1:d}^*$. We use $\omega(r_{g,d})$ as the corresponding action evidence. For recovery actions, we additionally quantify the aggregate Beam support for a target prefix. For any $1 \le h \le D - 1$, define the support

set $\mathcal{S}_{g,h} = \{j : \tilde{y}_{g,j,1:h} = y^*_{1:h}\}$ and its cardinality $n_{g,h} = |\mathcal{S}_{g,h}|$. The corresponding prefix-support score is

$$q_{g,h} = \begin{cases} \mathrm{clip}_{[0,1]}\left[\dfrac{1 - e^{-\frac{n_{g,h}}{\kappa}}}{1 - e^{-\frac{B}{\kappa}}} \cdot \dfrac{\sum_{j \in \mathcal{S}_{g,h}} \omega(j)}{n_{g,h}}\right], & n_{g,h} > 0, \\ 0, & n_{g,h} = 0. \end{cases} \tag{7}$$

Here, $\kappa > 0$ controls the saturation rate with respect to the support count. The first factor captures the coverage of the target prefix within the Beam, while the second measures the ranking quality of the supporting candidates. The maximum Beam width B calibrates the evidence scale consistently across different rollouts.

Accordingly, the Beam evidence for a target action already covered by the Beam is defined as $e_{g,d} = \omega(r_{g,d})$ for $d \le p_g$. For a recovery action at depth $d = p_g + 1$, when $p_g = 0$, the full-target recovery receives no Beam evidence and we set $e_{g,1} = 0$. When $1 \le p_g \le D - 2$, the recovery action uses the rank evidence of its deepest covered target prefix, and we set $e_{g,p_g+1} = \omega(r_{g,p_g})$. When $p_g = D - 1$, only the final SID level remains to be recovered, and we use the aggregate support of the $(D-1)$-level target prefix by setting $e_{g,D} = q_{g,D-1}$. When $p_g = D$, the target SID has already been fully covered by the Beam and no recovery action is constructed.

We further define the credit magnitude as $\gamma_{g,d} = A_{\max}[w_{\min} + (w_{\max} - w_{\min})e_{g,d}]$, where $A_{\max} > 0$ and $0 \le w_{\min} \le w_{\max} \le 1$. The local SID Credit is defined as

$$\Gamma_{g,d}(a) = \begin{cases} \gamma_{g,d}, & a = a^*_{g,d}, \\ -\dfrac{\gamma_{g,d}}{|\mathcal{A}_{g,d}| - 1}, & a \ne a^*_{g,d}, \end{cases} \qquad \sum_{a \in \mathcal{A}_{g,d}} \Gamma_{g,d}(a) = 0. \tag{8}$$

The target action receives positive credit, while the competing actions equally share a total negative credit of the same magnitude. This zero-sum design normalizes the credit budget across local action groups, preventing groups with more competing actions from receiving disproportionately large negative supervision.

3.4.3 Joint Optimization

For CoT tokens, we use the standard importance ratio $\rho_{g,t} = \frac{\pi_\theta(\tau_{g,t} | \mathcal{H}, \tau_{g,<t})}{\pi_{\theta_{\text{old}}}(\tau_{g,t} | \mathcal{H}, \tau_{g,<t})}$. For an SID action starting at depth d, let $x_{g,d} = (\mathcal{H}, \tau_g, y^*_{<d})$ denote its conditioning context. Since a recovery action may span multiple SID tokens, we define the length-normalized action log-probability as $\ell_\theta(a \mid x) = \frac{\log \pi_\theta(a|x)}{|a|}$, with the corresponding policy ratio $\rho_{g,d}(a) = \exp[\ell_\theta(a \mid x_{g,d}) - \ell_{\text{old}}(a \mid x_{g,d})]$.

For both positive and negative credit, we adopt the following clipped surrogate:

$$\Phi(\rho, A) = \min(\rho A, \mathrm{clip}(\rho, 1 - \epsilon, 1 + \epsilon)A)\,. \tag{9}$$

This objective optimizes credits of different signs under a unified clipping formulation, thereby maintaining a consistent form of policy update.

Let $\mathcal{V}_g$ denote the set of starting depths of valid SID action groups in the g-th trajectory, and let $\lambda_d \ge 0$ be the base weight for depth d. For a standard single-level action, we set $\lambda_{g,d} = \lambda_d$; for a recovery suffix starting at depth d, we instead set $\lambda_{g,d} = \sum_{r=d}^{D} \lambda_r$. The optimization objective is

$$\begin{aligned} \mathcal{L}_{\text{DG-HCA}} = &- \mathbb{E}_{g,t}\left[\Phi(\rho_{g,t}, A_g)\right] - \eta\, \mathbb{E}_g\left[\sum_{d \in \mathcal{V}_g} \lambda_{g,d} \sum_{a \in \mathcal{A}_{g,d}} \Phi(\rho_{g,d}(a), \Gamma_{g,d}(a))\right] \\ &+ \beta_{\text{KL}} D_{\text{KL}}(\pi_\theta \| \pi_{\text{ref}}). \end{aligned} \tag{10}$$

Here, η controls the relative weight of the SID branch with respect to the CoT branch, while β_{KL} controls KL regularization toward the reference policy. A_g provides reasoning-level credit, whereas $\Gamma_{g,d}(a)$ provides hierarchical SID-level credit. DG-HCA leaves the Beam Search procedure at inference time unchanged; instead, during training, it attributes the final retrieval feedback separately to the reasoning trajectory and to hierarchical decisions along the target SID path.

4 EXPERIMENTS

We report training and evaluation results on public datasets, experimental settings, and findings. Thinking mode is used unless otherwise specified.

Datasets. Following SIDReasoner's construction protocol (He et al., 2026), we use three public Amazon datasets (Ni et al., 2019): Video Games (Games), Office Products (Office), and Industrial and Scientific (Industrial). We remove training samples with fewer than five historical user interactions and order interaction histories chronologically. For fair comparison, initial fine-tuning uses SIDReasoner's SID–Language multi-task alignment and general reasoning knowledge data, with identical training, validation, and test splits.

Evaluation and Metrics. Following prior work (Bao et al., 2023a; 2024; Chen et al., 2024), the model generates reasoning trajectories before retrieving candidate SIDs via Constrained Beam Search. Samples whose reasoning does not terminate within the maximum generation length count as ground-truth item retrieval failures. We report Recall@K and NDCG@K, where $K \in \{5, 10\}$.

Baselines. We compare against three categories: (1) conventional discriminative sequential recommenders: Caser (Tang & Wang, 2018b), GRU4Rec (Hidasi et al., 2015), and SASRec (Kang & McAuley, 2018); (2) generative recommenders: TIGER (Rajput et al., 2023), HSTU (Zhai et al., 2024), LETTER (Wang et al., 2024a), and LC-Rec (Zheng et al., 2024); (3) generative recommenders with explicit or implicit reasoning: ReaRec (Tang et al., 2026), R^2ec (You et al., 2025), SIDReasoner (He et al., 2026), OneRec-Think (Liu et al., 2025b), and SAPO (Zheng et al., 2026b).

Implementation Details. All experiments involving LLMs use *Qwen3-1.7B* (Yang et al., 2025) as the backbone, extending its vocabulary with randomly initialized SID tokens. SID depth is $D = 3$. Stage 1 uses alignment data for 10 epochs; Stage 2 applies GRPO to the full training set for 2 epochs with 16 rollouts per instance; Stage 3 applies DG-HCA for 2 epochs, using 6 rollouts per instance and beam width $B = 10$ due to computational constraints.

4.1 PERFORMANCE COMPARISON ON RECOMMENDATION TASK

Table 1: Performance comparison on three public datasets. The best and second-best results are highlighted in **bold** and underline. † denotes results directly reported in SAPO (Zheng et al., 2026b).

Method	Industrial				Games				Office			
	R@5	N@5	R@10	N@10	R@5	N@5	R@10	N@10	R@5	N@5	R@10	N@10
Traditional Sequential Recommenders												
Caser	0.0672	0.0541	0.0890	0.0611	0.0354	0.0201	0.0681	0.0316	0.0872	0.0663	0.1108	0.0738
GRU4Rec	0.0784	0.0593	0.1011	0.0663	0.0356	0.0184	0.0579	0.0256	0.0729	0.0521	0.0987	0.0609
SASRec	0.0843	0.0720	0.1024	0.0776	0.0462	0.0301	0.0735	0.0397	0.0981	0.0812	0.1147	0.0868
Generative Recommenders without Reasoning												
TIGER	0.1006	0.0818	0.1324	0.0917	0.0537	0.0351	0.0778	0.0425	0.1209	0.0979	0.1419	0.1052
HSTU	0.1035	0.0861	0.1158	0.0903	0.0546	0.0408	0.0729	0.0465	0.1224	0.1071	0.1392	0.1122
LETTER	0.1107	0.0883	0.1369	0.0966	0.0383	0.0261	0.0647	0.0341	0.1306	0.1079	0.1555	0.1153
LC-Rec	0.0863	0.0547	0.1332	0.0712	0.0409	0.0260	0.0896	0.0432	0.1028	0.0757	0.1440	0.0887
Reasoning-based Generative Recommenders												
ReaRec	0.0919	0.0804	0.1197	0.0891	0.0529	0.0408	0.0821	0.0509	0.1152	0.1033	0.1378	0.1103
R^2ec	0.0899	0.0817	0.1247	0.0921	0.0636	0.0372	0.0941	0.0478	0.1114	0.0916	0.1509	0.1041
SIDReasoner	0.1074	0.0890	0.1465	0.1016	0.0568	0.0371	0.0917	0.0482	0.1320	0.1082	0.1627	0.1188
OneRec-Think†	0.0464	0.0398	0.0520	0.0416	0.0413	0.0296	0.0575	0.0349	0.1353	0.1107	0.1506	0.1166
SAPO†	**0.1176**	**0.0931**	0.1416	0.1012	0.0620	0.0413	0.0986	0.0529	0.1362	0.1141	0.1643	0.1245
DG-HCA(**Ours**)	0.1125	0.0899	**0.1524**	**0.1026**	**0.0741**	**0.0458**	**0.1097**	**0.0572**	**0.1377**	**0.1153**	**0.1685**	**0.1257**

We compare our method against 12 baseline methods on three datasets, with the results reported in Table 1. Overall, generative recommendation methods consistently outperform conventional discriminative recommenders, demonstrating the effectiveness of the generative paradigm for recommendation. Moreover, our method outperforms existing baselines on the majority of evaluation metrics and achieves the best NDCG@10 performance across all three public datasets. These results indicate that DG-HCA can effectively exploit retrieval feedback through semi-online fine-tuning, enabling the model to further improve its retrieval capability beyond that achieved by standard post-training objectives.

Table 2: Performance comparison across different training stages and ablation variants of DG-HCA.

Method	Industrial				Games				Office			
	R@5	N@5	R@10	N@10	R@5	N@5	R@10	N@10	R@5	N@5	R@10	N@10
Training Stages												
Stage 1	0.1006	0.0801	0.1319	0.0902	0.0464	0.0305	0.0775	0.0405	0.1130	0.0929	0.1379	0.1010
Stage 2	0.1021	0.0810	0.1291	0.0896	0.0562	0.0361	0.0847	0.0453	0.1286	0.1049	0.1521	0.1125
Stage 3	0.1125	0.0899	**0.1524**	**0.1026**	**0.0741**	**0.0458**	**0.1097**	**0.0572**	**0.1377**	**0.1153**	**0.1685**	**0.1257**
DG-HCA Ablations												
A1	0.1094	0.0899	0.1474	0.1020	0.0589	0.0395	0.0936	0.0506	0.1348	0.1096	0.1671	0.1200
A2	0.1101	0.0914	0.1469	0.1021	0.0621	0.0416	0.0949	0.0510	0.1356	0.1097	0.1656	0.1193
A3	**0.1127**	**0.0922**	0.1458	0.1023	0.0651	0.0412	0.1008	0.0526	0.1363	0.1108	0.1675	0.1209

4.2 Ablation Study

To evaluate the effectiveness of each component, we report the test-set performance after all three training stages and conduct a dedicated ablation study on the components of DG-HCA in Stage 3. We consider three variants. **A1** performs credit assignment only for CoT while removing SID-level credit assignment. **A2** retains credit assignment for both CoT and SID decisions but disables both Target-Path Recovery and Beam-evidence modulation, i.e., setting $\gamma_{g,d} = A_{\max}$. **A3** disables only Target-Path Recovery.

The results in Table 2 show that removing SID-level credit assignment, as in A1, leads to a substantial performance degradation, demonstrating the importance of explicitly attributing credit to hierarchical SID decisions. Reintroducing SID-level credit assignment without Target-Path Recovery improves performance over A1, but still falls short of the complete DG-HCA framework. We further compare model performance across the three training stages and observe consistent improvements across most evaluation metrics as training progresses. Notably, even the ablated variants of DG-HCA consistently outperform the Stage 2 model, indicating that fine-grained hierarchical credit assignment provides complementary gains beyond reasoning-mode guidance.

5 Analysis

5.1 Dataset Bucketing Analysis

To better characterize the recommendation gains introduced by Stage 3, we conduct bucket-wise evaluations on the Office and Industrial test sets. Target items are grouped by whether their first-level SID token appears in the interaction history and by item popularity. As shown in Figure 3, Stage 3 achieves relatively consistent gains over Stage 2 across the U-T, U-M, and U-H buckets, suggesting that the Target-Path Recovery mechanism in DG-HCA improves the retrieval of previously unseen items. Nevertheless, performance remains constrained by item popularity: clear improvement among tail items is mainly observed in S-T, while the particularly challenging U-T bucket exhibits only limited gains.

5.2 Non-thinking Mode Performance

We further evaluate the checkpoints from all three training stages under the non-thinking mode. Specifically, we disable the thinking mode of *Qwen3-1.7B* by prepending the `<think></think>` field and subsequently perform Constrained Beam Search. Across all three datasets, we observe that, despite the absence of dedicated training for the non-thinking mode and the resulting train–inference mismatch, recommendation performance continues to improve from Stage 1 to Stage 2 and Stage 3. Moreover, disabling the thinking mode yields better performance on most metrics than enabling it, which is consistent with the phenomenon reported by OneReason (Team et al., 2026). We attribute this observation to the fact that *Qwen3-1.7B* is neither pretrained to understand the semantics of recommendation-specific SIDs nor equipped with the extensive multi-stage, fine-grained semantic alignment objectives adopted by OneReason. Consequently, semantic deviations accumulated during explicit reasoning may adversely affect recommendation accuracy.

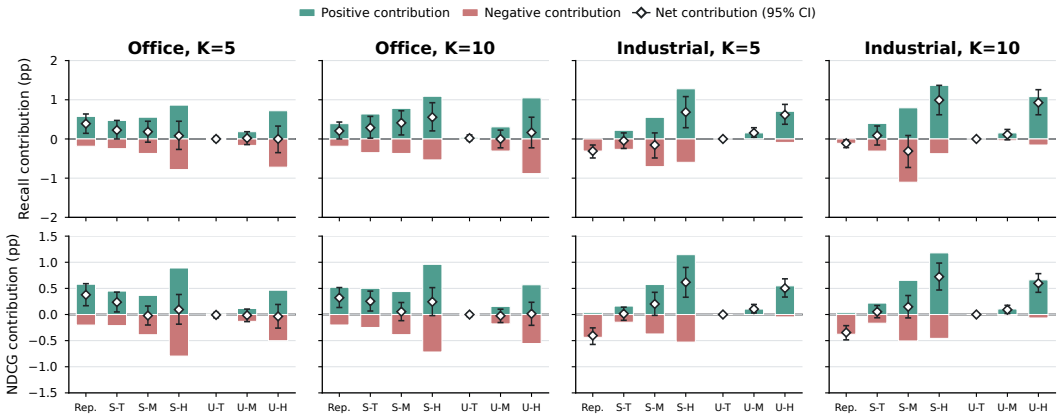


Figure 3: Comparison of test-set hit outcomes between Stage 3 and Stage 2. Values denote performance changes in percentage points; green and red indicate positive and negative contributions, respectively. S/U denote Seen/Unseen target SIDs, T/M/H denote Tail/Middle/Head items, and Repeat(Rep.) indicates that the target item appears in the interaction history.

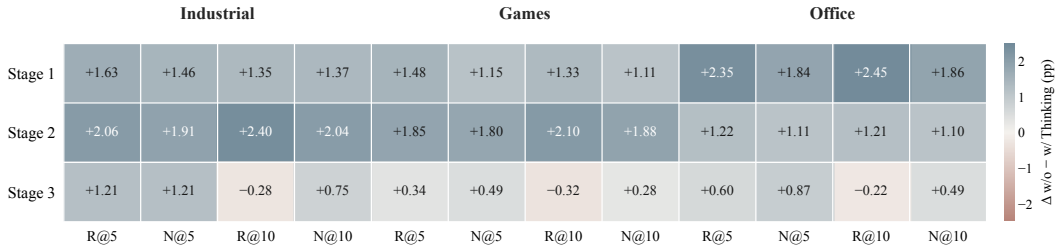


Figure 4: Effect of removing the thinking mode across different training stages. Each cell reports the performance difference between w/o Thinking and w/ Thinking in percentage points. Positive values indicate gains from removing Thinking, while negative values favor Thinking.

Nevertheless, the thinking mode is not intrinsically ineffective. Examining Recall@K with $K = 10$ reveals a more nuanced pattern. Under Stage 2, where GRPO is used for reasoning-mode guidance, the thinking mode is outperformed by the non-thinking mode on nearly all metrics. In contrast, Stage 3 with DG-HCA improves recall at larger K through semi-online fine-tuning with retrieval feedback, and the Stage 3 model under the thinking mode achieves competitive Recall@10 performance. These results suggest that GRPO alone is insufficient to fully exploit the benefits of explicit reasoning for recommendation. Instead, GRPO primarily serves to establish the desired reasoning behavior, while subsequent fine-grained optimization is necessary to effectively align reasoning trajectories with recommendation outcomes.

6 Conclusion

This work addresses the limitation of sequence-level credit assignment in capturing the contributions of hierarchical SID decisions in generative recommendation. We propose DG-HCA, which enables finer-grained policy optimization by decomposing credit at two complementary granularities—the reasoning-trajectory level and the SID-action level—while grounding the supervision in retrieval feedback from Beam Search. Experiments on the Games, Office, and Industrial datasets demonstrate that DG-HCA achieves competitive recommendation performance and attains the best NDCG@10 on all three datasets. Ablation studies further validate the effectiveness of SID-level credit assignment and Target-Path Recovery. Our results highlight that fine-grained credit assignment aligned with the structured generation process is an important direction for improving explicit-reasoning-based generative recommendation.

AI USE STATEMENT

In this work, we used generative AI tools to assist with language polishing and literature discovery for the Introduction. All AI-assisted content was manually reviewed and verified by the authors to ensure the correctness of experimental results and citations and to avoid inappropriate reuse of existing text. The authors take full responsibility for the final content of this paper.

ETHICS STATEMENT

All datasets used in this work are either publicly available or synthetically constructed from public data. User-related information was anonymized during dataset release, and our study does not involve access to personally identifiable information. Model evaluation is conducted entirely through automated procedures and does not involve human or crowdsourced annotators.

REPRODUCIBILITY STATEMENT

We provide the materials required to reproduce our experiments in the supplementary material and an anonymous repository https://anonymous.4open.science/r/DG-HCA-F444. We will also release the trained model checkpoints to facilitate reproducibility. Detailed training configurations, hyperparameters, and experimental procedures are documented in the appendix.

REFERENCES

Janice Ahn, Rishu Verma, Renze Lou, Di Liu, Rui Zhang, and Wenpeng Yin. Large language models for mathematical reasoning: Progresses and challenges. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop*, pp. 225–237, 2024.

Keqin Bao, Jizhi Zhang, Wenjie Wang, Yang Zhang, Zhengyi Yang, Yancheng Luo, Chong Chen, Fuli Feng, and Qi Tian. A bi-step grounding paradigm for large language models in recommendation systems, 2023a. URL https://arxiv.org/abs/2308.08434.

Keqin Bao, Jizhi Zhang, Yang Zhang, Wenjie Wang, Fuli Feng, and Xiangnan He. Tallrec: An effective and efficient tuning framework to align large language model with recommendation. In *Proceedings of the 17th ACM Conference on Recommender Systems*, RecSys '23, pp. 1007–1014. ACM, 2023b. doi: 10.1145/3604915.3608857. URL http://dx.doi.org/10.1145/3604915.3608857.

Keqin Bao, Jizhi Zhang, Yang Zhang, Xinyue Huo, Chong Chen, and Fuli Feng. Decoding matters: Addressing amplification bias and homogeneity issue in recommendations for large language models. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 10540–10552, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.589. URL https://aclanthology.org/2024.emnlp-main.589/.

Yuxin Chen, Junfei Tan, An Zhang, Zhengyi Yang, Leheng Sheng, Enzhi Zhang, Xiang Wang, and Tat-Seng Chua. On softmax direct preference optimization for recommendation, 2024. URL https://arxiv.org/abs/2406.09215.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Jiaxin Deng, Shiyao Wang, Kuo Cai, Lejian Ren, Qigen Hu, Weifeng Ding, Qiang Luo, and Guorui Zhou. Onerec: Unifying retrieve and rank with generative recommender and iterative preference alignment. *arXiv preprint arXiv:2502.18965*, 2025.

Yingzhi He, Yan Sun, Junfei Tan, Yuxin Chen, Xiaoyu Kong, Chunxu Shen, Xiang Wang, An Zhang, and Tat-Seng Chua. Reasoning over semantic ids enhances generative recommendation. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 1638–1649, 2026.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939*, 2015.

Yupeng Hou, Jiacheng Li, Ashley Shin, Jinsung Jeon, Abhishek Santhanam, Wei Shao, Kaveh Hassani, Ning Yao, and Julian McAuley. Generating long semantic ids in parallel for recommendation. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 956–966, 2025.

Jie Jiang, Yangru Huang, Zeyu Wang, Changping Wang, Yuling Xiong, Jun Zhang, and Huan Yu. Spend search where it pays: Value-guided structured sampling and optimization for generative recommendation. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 2098–2109, 2026a.

Juyong Jiang, Fan Wang, Jiasi Shen, Sungju Kim, and Sunghun Kim. A survey on large language models for code generation. *ACM Transactions on Software Engineering and Methodology*, 35 (2):1–72, 2026b.

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

Xiaoyu Kong, Leheng Sheng, Junfei Tan, Yuxin Chen, Jiancan Wu, An Zhang, Xiang Wang, and Xiangnan He. Minionerec: An open-source framework for scaling generative recommendation. *arXiv preprint arXiv:2510.24431*, 2025.

Doyup Lee, Chiheon Kim, Saehoon Kim, Minsu Cho, and Wook-Shin Han. Autoregressive image generation using residual quantization, 2022. URL https://arxiv.org/abs/2203.01941.

Enze Liu, Bowen Zheng, Cheng Ling, Lantao Hu, Han Li, and Wayne Xin Zhao. Generative recommender with end-to-end learnable item tokenization. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '25, pp. 729–739, New York, NY, USA, 2025a. Association for Computing Machinery. ISBN 9798400715921. doi: 10.1145/3726302.3729989. URL https://doi.org/10.1145/3726302.3729989.

Jian Liu, Leyang Cui, Hanmeng Liu, Dandan Huang, Yile Wang, and Yue Zhang. Logiqa: A challenge dataset for machine reading comprehension with logical reasoning. *arXiv preprint arXiv:2007.08124*, 2020.

Zhanyu Liu, Shiyao Wang, Xingmei Wang, Rongzhou Zhang, Jiaxin Deng, Honghui Bao, Jinghao Zhang, Wuchao Li, Pengfei Zheng, Xiangyu Wu, et al. Onerec-think: In-text reasoning for generative recommendation. *arXiv preprint arXiv:2510.11639*, 2025b.

Jianmo Ni, Jiacheng Li, and Julian McAuley. 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)*, pp. 188–197, 2019.

Shashank Rajput, Nikhil Mehta, Anima Singh, Raghunandan Hulikal Keshavan, Trung Vu, Lukasz Heldt, Lichan Hong, Yi Tay, Vinh Tran, Jonah Samost, et al. Recommender systems with generative retrieval. *Advances in Neural Information Processing Systems*, 36:10299–10315, 2023.

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

Jiakai Tang, Sunhao Dai, Teng Shi, Jun Xu, Xu Chen, Wen Chen, Jian Wu, and Yuning Jiang. Think before recommend: Unleashing the latent reasoning power for sequential recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 2026.

Jiaxi Tang and Ke Wang. Personalized top-n sequential recommendation via convolutional sequence embedding. In *Proceedings of the eleventh ACM international conference on web search and data mining*, pp. 565–573, 2018a.

Jiaxi Tang and Ke Wang. Personalized top-n sequential recommendation via convolutional sequence embedding, 2018b. URL https://arxiv.org/abs/1809.07426.

OneRec Team, Biao Yang, Boyang Ding, Chenglong Chu, Dunju Zang, Fei Pan, Han Li, Hao Jiang, Honghui Bao, Huanjie Wang, et al. Onereason technical report. *arXiv preprint arXiv:2606.06260*, 2026.

Jianxun Wang and Yixiang Chen. A review on code generation with llms: Application and evaluation. In *2023 IEEE International Conference on Medical Artificial Intelligence (MedAI)*, pp. 284–289. IEEE, 2023.

Wenjie Wang, Xinyu Lin, Fuli Feng, Xiangnan He, and Tat-Seng Chua. Generative recommendation: Towards next-generation recommender paradigm. *arXiv preprint arXiv:2304.03516*, 2023.

Wenjie Wang, Honghui Bao, Xinyu Lin, Jizhi Zhang, Yongqi Li, Fuli Feng, See-Kiong Ng, and Tat-Seng Chua. Learnable item tokenization for generative recommendation. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pp. 2400–2409, 2024a.

Ye Wang, Jiahao Xun, Minjie Hong, Jieming Zhu, Tao Jin, Wang Lin, Haoyuan Li, Linjun Li, Yan Xia, Zhou Zhao, and Zhenhua Dong. Eager: Two-stream generative recommender with behavior-semantic collaboration, 2024b. URL https://arxiv.org/abs/2406.14017.

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

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Shangxin Yang, Min Gao, Zongwei Wang, and Junliang Yu. Where reasoning matters: Rethinking latent reasoning in semantic id-based generative recommendation. *arXiv preprint arXiv:2607.12425*, 2026.

Runyang You, Yongqi Li, Xinyu Lin, Xin Zhang, Wenjie Wang, Wenjie Li, and Liqiang Nie. R^2ec: Towards large recommender models with reasoning, 2025. URL https://arxiv.org/abs/2505.16994.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *Advances in Neural Information Processing Systems*, 38:113222–113244, 2026.

Jiaqi Zhai, Lucy Liao, Xing Liu, Yueming Wang, Rui Li, Xuan Cao, Leon Gao, Zhaojie Gong, Fangda Gu, Michael He, Yinghai Lu, and Yu Shi. Actions speak louder than words: Trillion-parameter sequential transducers for generative recommendations, 2024. URL https://arxiv.org/abs/2402.17152.

Kangning Zhang, Haotian Fang, Xukun Luo, Hao Yin, Yang Gao, Peng Yan, Weiwen Liu, Weinan Zhang, and Yong Yu. Hcgrec: Hint-conditioned generative recommendation with semantic ids. *arXiv preprint arXiv:2608.11980*, 2026a.

Yang Zhang, Wenxin Xu, Xiaoyan Zhao, Wenjie Wang, Fuli Feng, Xiangnan He, and Tat-Seng Chua. Reinforced latent reasoning for llm-based recommendation. In *International Conference on Learning Representations*, volume 2026, pp. 128449–128470, 2026b.

Bowen Zheng, Yupeng Hou, Hongyu Lu, Yu Chen, Wayne Xin Zhao, Ming Chen, and Ji-Rong Wen. Adapting large language models by integrating collaborative semantics for recommendation, 2024. URL https://arxiv.org/abs/2311.09049.

Bowen Zheng, Chao Yi, Dian Chen, Gaoyang Guo, Han Zhu, Jiakai Tang, Jian Wu, Mao Zhang, Wen Chen, Yifan Lu, et al. Recgpt-v3 technical report. *arXiv preprint arXiv:2607.15591*, 2026a.

Zaiyi Zheng, Liang Wu, Guanghui Min, Yaochen Zhu, Liangjie Hong, Chen Chen, and Jundong Li. Sapo: Step-aligned policy optimization for reasoning-based generative recommendation. *arXiv preprint arXiv:2605.17648*, 2026b.

Guorui Zhou, Jiaxin Deng, Jinghao Zhang, Kuo Cai, Lejian Ren, Qiang Luo, Qianqian Wang, Qigen Hu, Rui Huang, Shiyao Wang, et al. Onerec technical report. *arXiv preprint arXiv:2506.13695*, 2025.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*, 2023.

Yaochen Zhu, Harald Steck, Dawen Liang, Yinhan He, Vito Ostuni, Jundong Li, and Nathan Kallus. Rank-grpo: Training llm-based conversational recommender systems with reinforcement learning. In *International Conference on Learning Representations*, volume 2026, pp. 155483–155512, 2026.

APPENDIX

CONTENTS

A Summary of Model Performance **15**

B Limitations **15**

C Implementation Details for Training **15**

D GRPO for Reasoning-Mode Guidance **15**

E More Analysis **17**

E.1 General Capability Evaluation 17

E.2 Training Dynamics 18

E.3 More Ablation Studies 19

F Theoretical Analysis **19**

F.1 Reward-Level Bounded Refinement of the Retrieval Objective 20

F.2 Hierarchical Informativeness under Sparse Group Feedback 21

F.3 Coordinate-Wise Sign Stability of the Clipped SID Surrogate 22

F.4 Local Contrastive Geometry of SID Credit 23

F.5 Stability of Length-Normalized Recovery Policy Ratios 24

G Discussions **26**

G.1 Why Does Beam Evidence Switch at $p_g = D - 1$? 26

G.2 Why Does SID Credit Use a Zero-Sum Assignment? 27

G.3 How Does Skipping Stage 2 Reasoning GRPO Affect Model Performance? 27

H Notation **28**

I Case Study **28**

A Summary of Model Performance

We use TK to denote that the model operates in thinking mode. Table 3 summarizes the performance of different training stages and baselines under both Thinking and non-Thinking settings.

Table 3: Performance comparison on three public datasets. The best and second-best results are highlighted in **bold** and underline, respectively. † denotes results directly reported in SAPO (Zheng et al., 2026b).

Method	Industrial				Games				Office			
	R@5	N@5	R@10	N@10	R@5	N@5	R@10	N@10	R@5	N@5	R@10	N@10
Traditional Sequential Recommenders												
Caser	0.0672	0.0541	0.0890	0.0611	0.0354	0.0201	0.0681	0.0316	0.0872	0.0663	0.1108	0.0738
GRU4Rec	0.0784	0.0593	0.1011	0.0663	0.0356	0.0184	0.0579	0.0256	0.0729	0.0521	0.0987	0.0609
SASRec	0.0843	0.0720	0.1024	0.0776	0.0462	0.0301	0.0735	0.0397	0.0981	0.0812	0.1147	0.0868
Generative Recommenders without Reasoning												
TIGER	0.1006	0.0818	0.1324	0.0917	0.0537	0.0351	0.0778	0.0425	0.1209	0.0979	0.1419	0.1052
HSTU	0.1035	0.0861	0.1158	0.0903	0.0546	0.0408	0.0729	0.0465	0.1224	0.1071	0.1392	0.1122
LETTER	0.1107	0.0883	0.1369	0.0966	0.0383	0.0261	0.0647	0.0341	0.1306	0.1079	0.1555	0.1153
LC-Rec	0.0863	0.0547	0.1332	0.0712	0.0409	0.0260	0.0896	0.0432	0.1028	0.0757	0.1440	0.0887
Reasoning-based Generative Recommenders												
ReaRec	0.0919	0.0804	0.1197	0.0891	0.0529	0.0408	0.0821	0.0509	0.1152	0.1033	0.1378	0.1103
R^2ec	0.0899	0.0817	0.1247	0.0921	0.0636	0.0372	0.0941	0.0478	0.1114	0.0916	0.1509	0.1041
SIDReasoner	0.1074	0.0890	0.1465	0.1016	0.0568	0.0371	0.0917	0.0482	0.1320	0.1082	0.1627	0.1188
OneRec-Think†	0.0464	0.0398	0.0520	0.0416	0.0413	0.0296	0.0575	0.0349	0.1353	0.1107	0.1506	0.1166
SAPO†	0.1176	0.0931	0.1416	0.1012	0.0620	0.0413	0.0986	0.0529	0.1362	0.1141	0.1643	0.1245
Thinking-Mode Ablation												
Stage 1 w/ TK	0.1006	0.0801	0.1319	0.0902	0.0464	0.0305	0.0775	0.0405	0.1130	0.0929	0.1379	0.1010
Stage 2 w/ TK	0.1021	0.0810	0.1291	0.0896	0.0562	0.0361	0.0847	0.0453	0.1286	0.1049	0.1521	0.1125
Stage 3 w/ TK (Ours)	0.1125	0.0899	0.1524	0.1026	0.0741	0.0458	**0.1097**	0.0572	0.1377	0.1153	**0.1685**	0.1257
Stage 1 w/o TK	0.1169	0.0947	0.1454	0.1039	0.0612	0.0420	0.0908	0.0516	0.1365	0.1113	0.1624	0.1196
Stage 2 w/o TK	0.1227	0.1001	**0.1531**	0.1100	0.0747	**0.0541**	0.1057	**0.0641**	0.1408	0.1160	0.1642	0.1235
Stage 3 w/o TK (Ours)	**0.1246**	**0.1020**	0.1496	**0.1101**	**0.0775**	0.0507	0.1065	0.0600	**0.1437**	**0.1240**	0.1663	**0.1306**

B Limitations

Although DG-HCA performs dual-granularity credit assignment at the reasoning-trajectory and SID-action levels, all reasoning tokens within the same trajectory still share a common trajectory-level credit signal. This limitation mainly stems from the lack of sufficiently dense supervision for user-preference reasoning in existing recommendation data, as well as the absence of mature process reward models tailored to the recommendation domain. In future work, we aim to develop finer-grained credit assignment mechanisms for recommendation reasoning trajectories, enabling more precise attribution of individual reasoning steps to final recommendation outcomes.

C Implementation Details for Training

We summarize the training-related hyperparameters in Table 4 and Table 5. For **Stage 1** and **Stage 2**, we follow the training configurations and procedures of SIDReasoner (He et al., 2026). Specifically, in Stage 1, the RQ-VAE Tokenization directly uses the precomputed SID indices generated by the SIDReasoner preprocessing protocol, without retraining the RQ-VAE tokenizer. In actual experiments, the SFT procedure in Stage 1 consists of two consecutive sub-stages, namely SID–Language Alignment and Reasoning Activation.

D GRPO for Reasoning-Mode Guidance

The reward and advantage notation in this appendix is specific to Stage 2. Given a user history $\mathcal{H}$, the model first generates a reasoning trajectory τ_g and then predicts an SID y_g. Let m_g denote the length of the longest correct prefix shared by y_g and the target SID y^*, $\mathcal{Y}_{\text{catalog}}$ the set of valid SIDs corresponding to catalog items, and $\lambda_{\text{val}} \geq 0$ the validity-reward weight. Stage 2 uses the outcome-level reward $R_g = 2^{m_g - D} + \lambda_{val} \cdot \mathbf{1}[y_g \in \mathcal{Y}_{\text{catalog}}]$, where the first term is a hierarchical

Table 4: Stage 3 training configuration used for DG-HCA.

Category	Parameter	Value
Data	Initialization	Merged Stage 2 GRPO checkpoint
Data	Training / monitoring data	Full training split / fixed validation subset
Data	Prompt collection batch per GPU	8
Data	Consecutive rollout chunks per macro-batch	8
Data	Global rollout prompt batch	$8 \times 8 \times 4 = 256$
Data	Sampled trajectories per prompt G	6
Data	Trajectories per rollout macro-batch	$256 \times 6 = 1{,}536$
Data	PPO prompt minibatch	128 global / 32 per GPU
Data	PPO rollout microbatch per GPU	8 trajectories
Data	Microbatches per full optimizer step	$(32 \times 6)/8 = 24$ per GPU
Data	Maximum prompt length	1,024 tokens
Data	Maximum reasoning-response length	512 tokens
Algorithm	Training objective	DG-HCA
Algorithm	CoT advantage estimator	Group-normalized ranking-driven reward
Algorithm	Group-normalization threshold ϵ_R	10^{-6}; constant groups receive zero advantage
Algorithm	Invalid-trajectory reward R_{inv}	-0.2
Algorithm	Constrained-Beam width B	10
Algorithm	Target-Path Recovery	Enabled
Algorithm	SID depth weights $(\lambda_1, \lambda_2, \lambda_3)$	$(0.2, 0.3, 0.5)$
Algorithm	Prefix-support saturation κ	2.0
Algorithm	Smooth evidence range $(w_{\min}, w_{\max})$	$(0.2, 1.0)$
Algorithm	Maximum target credit $A_{\max}$	2.0
Algorithm	SID-branch coefficient η	0.02
Optimization	Trainable parameters	LoRA adapter
Optimization	Optimizer	AdamW
Optimization	Learning rate	5×10^{-7} (constant; no scheduler)
Optimization	AdamW (β_1, β_2) / epsilon	$(0.9, 0.999)$ / 10^{-8}
Optimization	Weight decay	0.01
Optimization	PPO epochs per rollout macro-batch	1
Optimization	PPO minibatch shuffling	False
Optimization	Ratio clip ϵ	0.2
Optimization	KL estimator / coefficient β_{KL}	Low-variance KL / 0.001
Optimization	Entropy regularization coefficient	0
Optimization	Maximum gradient norm	1.0
Optimization	LoRA rank / alpha / dropout	16 / 32 / 0
Optimization	LoRA bias	None
Optimization	LoRA target projections	`q_proj`, `k_proj`, `v_proj`, `o_proj`, `gate_proj`, `up_proj`, `down_proj`
Rollout/Search	Generation backend	Hugging Face Transformers `model.generate`
Rollout/Search	Rollout mode	Online and synchronous
Rollout/Search	Reasoning generation	Sampling; stop after `</think>`
Rollout/Search	Temperature / top-p	1.0 / 1.0
Rollout/Search	SID retrieval	Deterministic catalog-constrained Beam Search
Rollout/Search	Beam count / returned candidates	10 / 10
Rollout/Search	Beam scoring batch size	100
Rollout/Search	Maximum Beam conditioning length	1,536 tokens
Rollout/Search	Maximum SID decoding length	32 tokens
Rollout/Search	Maximum sequences per log-probability forward pass	8
Rollout/Search	Beam post-processing	Remove malformed, out-of-catalog, and duplicate SIDs
Rollout/Search	Reference policy	Stage 2 base policy with the LoRA adapter disabled
Parallelism/Memory	Distributed strategy / number of GPUs	DDP / 4
Parallelism/Memory	Numerical precision	bfloat16
Parallelism/Memory	Gradient checkpointing	Enabled, non-reentrant
Parallelism/Memory	DDP unused-parameter detection / buffer broadcast	True / False
Parallelism/Memory	KV cache during training / generation	Disabled / enabled
Trainer	Total epochs	2
Trainer	Training and sampling seed	42
Trainer	Training-data shuffling	Enabled with a distributed sampler
Trainer	Checkpoint frequency	Every 0.1 epoch
Trainer	Fixed validation frequency	Every 0.05 epoch
Trainer	Fixed validation size / batch size / seed	100 / 4 / 20260705
Trainer	Metric logging frequency	Every optimizer step
Trainer	Saved checkpoint content	LoRA adapter and tokenizer

Table 5: Hyperparameter settings used in the three-stage training pipeline.

Stage	Hyperparameter	Value
Shared	SID depth D	3
Stage 1	Codebook size M	256
	Commitment weight β_{com}	–
	Embedding dimension d_{emb}	–
Stage 2	Rollouts per history G	16
	Advantage stabilizer ε	10^{-6}
	PPO clipping threshold ϵ_G	0.2
	KL coefficient β_G	0.001
Stage 3	Trajectories per history G	6
	Beam width B	10
	Invalid-trajectory reward R_{inv}	-0.2
	Advantage stabilizer ε_R	10^{-6}
	Prefix-support saturation κ	2.0
	Minimum credit weight $w_{\min}$	0.2
	Maximum credit weight $w_{\max}$	1.0
	Credit scale $A_{\max}$	2.0
	Hierarchy weights $(\lambda_1, \lambda_2, \lambda_3)$	(0.2, 0.3, 0.5)
	SID-branch weight η	0.02
	PPO clipping threshold ϵ	0.2
	KL coefficient β_{KL}	0.001

prefix reward and the second encourages valid SID generation. The group-normalized advantage is $\hat{A}_g = \frac{R_g - \mu}{\sigma + \varepsilon}$, where μ and σ are the group-wise reward mean and standard deviation, respectively, and $\varepsilon > 0$ ensures numerical stability.

For the complete rollout $o_g = \tau_g \circ y_g$, the token-level importance ratio is $\rho_{g,t} = \frac{\pi_\theta(o_{g,t}|\mathcal{H}, o_{g,<t})}{\pi_{\theta_{\text{old}}}(o_{g,t}|\mathcal{H}, o_{g,<t})}$. GRPO adopts the standard clipped surrogate objective:

$$\begin{aligned}\mathcal{L}_{\text{GRPO}} = &- \mathbb{E}_{g,t}\Big[\min\Big(\rho_{g,t}\hat{A}_g,\ \text{clip}(\rho_{g,t}, 1-\epsilon_{\text{G}}, 1+\epsilon_{\text{G}})\hat{A}_g\Big)\Big] \\ &+ \beta_{\text{G}} D_{\text{KL}}(\pi_\theta \| \pi_{\text{ref}}),\end{aligned} \tag{11}$$

where ϵ_{G} is the clipping threshold and β_{G} controls the KL regularization strength. A key limitation is that all tokens within the same rollout share the same sequence-level advantage, i.e., $\hat{A}_{g,t} = \hat{A}_g$. Stage 2 uses this coarse-grained signal for reasoning-mode guidance, while DG-HCA in Stage 3 further decomposes credit assignment across two granularities: the reasoning trajectory and hierarchical SID decisions.

It is worth noting that SIDReasoner uses GRPO as its main training method and trains for 10 epochs, whereas our method only uses GRPO as a guide for the thinking mode and trains with GRPO for 2 epochs. Our more important innovation is Stage 3 (DG-HCA).

E More Analysis

E.1 General Capability Evaluation

We further evaluate the models obtained after each of the three training stages on general-purpose benchmarks. MMLU (Hendrycks et al., 2020) evaluates multidisciplinary knowledge understanding and reasoning, IFEval (Zhou et al., 2023) measures instruction-following ability under complex instructions and formatting constraints, GSM8K (Cobbe et al., 2021) assesses multi-step reasoning on grade-school mathematical problems, and LogiQA (Liu et al., 2020) focuses on conditional analysis, deductive reasoning, and logical inference.

The results in Table 6 indicate that the three-stage training pipeline largely preserves the model's general capabilities, with no evident performance collapse across these benchmarks. Interestingly, we observe an improvement on MMLU, which may be attributed to the inclusion of general

Table 6: General capability evaluation across different training stages and recommendation datasets.

Stage	MMLU			IFEval			GSM8K			LogiQA		
	Industrial	Office	Games	Industrial	Office	Games	Industrial	Office	Games	Industrial	Office	Games
Qwen3-1.7B	0.2295			0.2292			0.4958			0.3241		
Stage 1	**0.5310**	**0.5351**	0.5126	0.2144	**0.2736**	**0.2403**	0.5459	0.5019	0.3563	**0.3103**	0.3041	0.2980
Stage 2	0.5291	0.5345	**0.5128**	0.2163	**0.2736**	0.1423	**0.5550**	0.5087	**0.5239**	**0.3103**	**0.3088**	0.2965
Stage 3	0.5198	0.5031	0.5077	**0.2236**	0.2311	0.1738	0.5459	**0.5466**	0.5034	0.2903	0.2995	**0.3164**

reasoning-oriented data in the Stage 1 fine-tuning corpus. This suggests that incorporating such data during the alignment stage can help mitigate the degradation of general capabilities while adapting the model to recommendation-specific objectives.

E.2 Training Dynamics

On the Industrial dataset, we further continue training the Stage 2 checkpoint with GRPO for two additional epochs and compare it against Stage 3 trained with DG-HCA. We track the response length and generation entropy throughout training, as illustrated in the figure. Compared with continued GRPO training, DG-HCA exhibits no evident downward trend in either response length or entropy, and neither quantity collapses during optimization. These training dynamics suggest that DG-HCA can smoothly continue fine-grained optimization after GRPO has established a reasonable reasoning mode, while preserving response diversity and generation behavior and further improving recommendation performance.

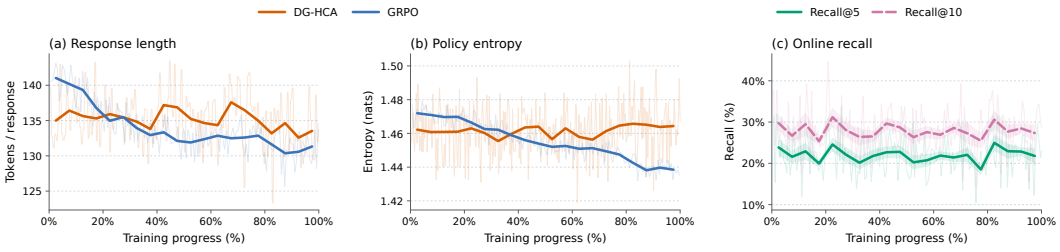


Figure 5: **(a)** Changes in response length during training with DG-HCA and GRPO. **(b)** Changes in policy entropy during training with DG-HCA and GRPO. **(c)** Recall@K curves obtained after Beam Search during DG-HCA training.

Furthermore, we randomly sample a small subset of the validation set to monitor model performance during training and report Recall@K and NDCG@K on this subset. As shown by the performance curves, although Recall@K on the training data exhibits noticeable fluctuations, the model achieves a steady improvement on the held-out validation subset throughout training.

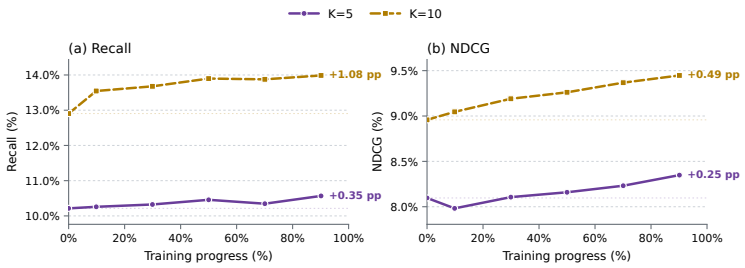


Figure 6: Performance evaluated on the Industrial validation subset during training.

E.3 More Ablation Studies

In this subsection, we conduct additional ablation studies on DG-HCA to better understand how its hyperparameter settings and individual components contribute to model performance. For computational efficiency, we restrict these experiments to the Industrial and Scientific dataset.

We first investigate the effect of the number of beam candidates while keeping all other hyperparameters fixed during training. The results are summarized in Table 7. By default, DG-HCA uses 10 beam candidates. Reducing this number to 5 leads to substantial performance degradation in Recall@5 and NDCG@5. Increasing the number beyond the default yields modest improvements in NDCG@5 and NDCG@10, but provides limited gains on the remaining metrics, while incurring higher computational costs and substantially longer training times. These results suggest that using 10 beam candidates offers a favorable trade-off between model performance and computational efficiency.

Table 7: Effect of the number of Beam Candidates on DG-HCA performance on the Industrial and Scientific dataset. The row shaded in gray reports the results obtained using the default configuration.

Training Beam Candidates	R@5	N@5	R@10	N@10
5	0.1059	0.0855	0.1500	0.0995
10	**0.1125**	0.0899	**0.1524**	0.1026
20	0.1116	**0.0907**	0.1507	**0.1032**

We further investigate the relative contributions of the CoT Branch Loss and the SID Branch Loss by conducting ablation studies on the weighting coefficient η and the inclusion of the CoT Branch Loss. Specifically, when the CoT Branch Loss is disabled, we still follow the same inference procedure of first generating the reasoning trajectory and subsequently performing Beam Search; however, gradients from the CoT Branch Loss are no longer backpropagated during training. The detailed results are reported in Table 8.

As shown in the table, removing the CoT Branch Loss leads to consistent performance degradation across all four evaluation metrics. This observation indicates that relying solely on the SID-level credit signal is insufficient to fully substitute for the direct optimization of the reasoning process. Therefore, we conclude that the CoT Branch Loss provides an independent and consistently positive contribution to the overall performance of DG-HCA. Furthermore, increasing the value of η does not yield consistent performance gains. These results suggest that the SID-level supervision should serve as a weak auxiliary signal rather than dominate the joint optimization objective.

Table 8: Ablation study of CoT Loss and the SID Loss weight η on the Industrial and Scientific dataset. The row shaded in gray reports the results obtained using the default configuration.

CoT Loss	η	R@5	N@5	R@10	N@10
✓	0.02	**0.1125**	**0.0899**	0.1524	**0.1026**
✓	0.10	0.1094	0.0888	0.1522	0.1025
✓	0.50	0.1101	0.0889	**0.1529**	0.1026
✓	1.00	0.1105	0.0883	0.1522	0.1015
✗	0.02	0.1099	0.0888	0.1513	0.1021

F Theoretical Analysis

This appendix provides a theoretical analysis of DG-HCA from four perspectives. We first show that the ranking-driven reasoning reward is a bounded refinement of the per-instance single-target NDCG@B objective. We then study the informativeness of hierarchical SID credit when trajectory-level group-relative advantages vanish. Next, we analyze the clipped SID surrogate through its coordinate-wise sign behavior and local contrastive structure. Finally, we examine the length-normalized policy ratio used for multi-token recovery actions.

Boundary conventions. We assume $B \geq 1$. If a valid reasoning trajectory produces an empty constrained Beam, its valid-miss reward in Eq. 4 is set to 0 and no SID action group is constructed. Invalid trajectories and trajectories without an exact target hit are assigned retrieval score 0.

F.1 Reward-Level Bounded Refinement of the Retrieval Objective

Let $\omega(r) = 1/\log_2(r+1)$ denote the rank discount. For a non-exact candidate $\tilde{y}$, define its longest target-prefix length as

$$m(\tilde{y}, y^*) = \max\{h \in \{0, \dots, D\} : \tilde{y}_{1:h} = y^*_{1:h}\}, \tag{12}$$

and its hierarchical prefix score as

$$\psi(\tilde{y}, y^*) = \mathbf{1}[m(\tilde{y}, y^*) > 0]\, 2^{m(\tilde{y}, y^*) - D}. \tag{13}$$

For a reasoning trajectory and its constrained Beam of width at most B, define the single-target retrieval score by

$$N_B = \begin{cases} \omega(r), & \text{if the trajectory is valid and } y^* \text{ appears at rank } r \leq B, \\ 0, & \text{otherwise.} \end{cases} \tag{14}$$

With a single relevant target item, N_B is exactly the per-instance NDCG@B.

Let H denote the event of a valid exact Top-B hit, M the event of a valid miss, and I the event of an invalid reasoning trajectory. These events form a partition of the trajectory space. Let $\delta_B = \omega(B)/2$ and $\alpha = -R_{\text{inv}} > 0$.

Theorem 1 (Reward-level bounded retrieval refinement)**.** *For every policy π, define $J_R(\pi) = \mathbb{E}_\pi[R_B]$, $J_N(\pi) = \mathbb{E}_\pi[N_B]$, and $p_{\text{inv}}(\pi) = \Pr_\pi(\mathsf{I})$. Then*

$$J_R(\pi) = J_N(\pi) + S(\pi) - \alpha p_{\text{inv}}(\pi), \tag{15}$$

where $S(\pi) = \mathbb{E}_\pi[R_B \mathbf{1}_{\mathsf{M}}]$ and $0 \leq S(\pi) \leq \delta_B \Pr_\pi(\mathsf{M})$. Consequently,

$$J_N(\pi) - \alpha p_{\text{inv}}(\pi) \leq J_R(\pi) \leq \delta_B + (1 - \delta_B) J_N(\pi). \tag{16}$$

For any tolerance $\xi \in [0, 1]$, define $\Pi_\xi = \{\pi \in \Pi : p_{\text{inv}}(\pi) \leq \xi\}$. Assume Π_ξ is non-empty and the relevant maximizers exist. Let $J_N^* = \max_{\pi \in \Pi_\xi} J_N(\pi)$ and $\pi_R \in \arg\max_{\pi \in \Pi_\xi} J_R(\pi)$. Then

$$J_N(\pi_R) \geq \frac{J_N^* - \delta_B - \alpha\xi}{1 - \delta_B}, \tag{17}$$

or equivalently,

$$J_N^* - J_N(\pi_R) \leq \frac{\delta_B(1 - J_N^*) + \alpha\xi}{1 - \delta_B}. \tag{18}$$

Proof. On H, Eq. 4 gives $R_B = \omega(r) = N_B$. On I, we have $R_B = R_{\text{inv}} = -\alpha$ and $N_B = 0$.

On M, an empty constrained Beam gives $R_B = 0$ by convention. Otherwise every Beam candidate is non-exact, so $m(\tilde{y}, y^*) \leq D - 1$ and therefore $\psi(\tilde{y}, y^*) \leq 1/2$. Since $\omega(j) \leq 1$,

$$0 \leq R_B = \omega(B) \max_j [\psi(\tilde{y}_j, y^*)\omega(j)] \leq \frac{\omega(B)}{2} = \delta_B. \tag{19}$$

Taking expectations over $\mathsf{H} \dot\cup \mathsf{M} \dot\cup \mathsf{I}$ yields

$$J_R(\pi) = J_N(\pi) + S(\pi) - \alpha p_{\text{inv}}(\pi), \tag{20}$$

with $0 \leq S(\pi) \leq \delta_B \Pr_\pi(\mathsf{M})$.

Because $S(\pi) \geq 0$, the decomposition gives $J_R(\pi) \geq J_N(\pi) - \alpha p_{\text{inv}}(\pi)$. For the upper bound, $N_B \leq \mathbf{1}_{\mathsf{H}}$ implies $J_N(\pi) \leq \Pr_\pi(\mathsf{H})$. Since $\Pr_\pi(\mathsf{M}) \leq 1 - \Pr_\pi(\mathsf{H})$,

$$\begin{aligned} J_R(\pi) &\leq J_N(\pi) + \delta_B \Pr_\pi(\mathsf{M}) \\ &\leq J_N(\pi) + \delta_B\big(1 - J_N(\pi)\big) \\ &= \delta_B + (1 - \delta_B) J_N(\pi). \end{aligned} \tag{21}$$

Let $\pi_N \in \arg\max_{\pi\in\Pi_\xi} J_N(\pi)$. Since $\pi_N \in \Pi_\xi$, $p_{\text{inv}}(\pi_N) \le \xi$. Applying the lower bound to π_N gives

$$\begin{aligned} J_R(\pi_N) &\ge J_N(\pi_N) - \alpha p_{\text{inv}}(\pi_N) \\ &\ge J_N^* - \alpha\xi. \end{aligned} \tag{22}$$

By the optimality of π_R over the same restricted class,

$$J_R(\pi_R) \ge J_R(\pi_N) \ge J_N^* - \alpha\xi. \tag{23}$$

Combining this with $J_R(\pi_R) \le \delta_B + (1-\delta_B)J_N(\pi_R)$ yields

$$J_N^* - \alpha\xi \le \delta_B + (1-\delta_B)J_N(\pi_R), \tag{24}$$

which gives the stated bound after rearrangement. □

Corollary 1 (Exact reward-level optimizer coincidence under rank-one realizability)**.** *Suppose there exists a policy $\pi^* \in \Pi$ for which the target SID appears at Beam rank one almost surely. Then*

$$\arg\max_{\pi\in\Pi} J_R(\pi) = \arg\max_{\pi\in\Pi} J_N(\pi). \tag{25}$$

Proof. Both raw objectives are bounded above by one. The equality $J_N(\pi) = 1$ holds if and only if the target appears at rank one almost surely. The same condition is necessary and sufficient for $J_R(\pi) = 1$: a rank-one hit receives reward 1, every lower-ranked hit receives $\omega(r) < 1$, every valid miss receives at most $\delta_B < 1$, and every invalid trajectory receives a negative reward. Since rank one is realizable, the common upper bound is attainable and the maximizing policy sets coincide. □

Corollary 2 (Within-group order preservation under group normalization)**.** *For two reasoning trajectories g and h sampled for the same user history,*

$$A_g - A_h = \frac{R_g - R_h}{\sigma_R + \varepsilon_R}. \tag{26}$$

Since $\sigma_R + \varepsilon_R > 0$, $R_g > R_h$ if and only if $A_g > A_h$.

Remark 1 (Scope of the reward-level result). The tolerance ξ controls the admissible invalid-trajectory probability in the analysis, while $\alpha = -R_{\text{inv}}$ is the magnitude of the invalidity penalty. The bound follows from the one-sided relation $J_R(\pi) \ge J_N(\pi) - \alpha p_{\text{inv}}(\pi)$. The result concerns the raw ranking-driven reward in Eq. 4; the complete DG-HCA objective additionally contains group normalization, clipping, the SID branch, and KL regularization.

F.2 Hierarchical Informativeness under Sparse Group Feedback

For an admissible rollout group $\mathcal{G} = \{(\tau_g, \mathcal{B}_g)\}_{g=1}^G$, define

$$\mathfrak{I}_{\text{CoT}} = \{\mathcal{G} : \exists g,\ A_g \ne 0\}, \tag{27}$$

and

$$\mathfrak{I}_{\text{DG}} = \{\mathcal{G} : \exists g,\ A_g \ne 0 \text{ or } \exists (g,d,a),\ \Gamma_{g,d}(a) \ne 0\}. \tag{28}$$

We first establish a structural property of the recovery group at the frontier where Beam Search departs from the target path.

Lemma 1 (Non-degeneracy of the recovery group)**.** *Suppose the constrained Beam is non-empty and misses the target, i.e., $\mathcal{B}_g \ne \varnothing$ and $y^* \notin \mathcal{B}_g$. Then $p_g < D$. Moreover, at the recovery depth $d = p_g + 1$,*

$$|\mathcal{A}_{g,d}| \ge 2. \tag{29}$$

Proof. Because $y^* \notin \mathcal{B}_g$, no Beam candidate matches the target SID at all D levels, so $p_g < D$. Let $d = p_g + 1$.

Since $\mathcal{B}_g$ is non-empty, the definition of p_g guarantees that there exists at least one candidate $\tilde{y}_{g,j} \in \mathcal{B}_g$ such that $\tilde{y}_{g,j,1:p_g} = y^*_{1:p_g}$. Hence its suffix $\tilde{y}_{g,j,d:D}$ belongs to the Beam-derived competitor set in the recovery group.

Target-Path Recovery explicitly inserts the target suffix $y^*_{d:D}$ into the same group. Moreover, $\tilde{y}_{g,j,d:D} \ne y^*_{d:D}$; otherwise the prefix equality together with suffix equality would imply $\tilde{y}_{g,j} = y^*$, contradicting $y^* \notin \mathcal{B}_g$. Thus $\mathcal{A}_{g,d}$ contains the target recovery suffix and at least one distinct Beam-derived suffix, which proves $|\mathcal{A}_{g,d}| \ge 2$. □

Theorem 2 (Hierarchical informativeness expansion). *For every admissible rollout-group space,*

$$\mathfrak{I}_{\mathrm{CoT}} \subseteq \mathfrak{I}_{\mathrm{DG}}. \tag{30}$$

Assume $A_{\max} > 0$, $0 \le w_{\min} \le w_{\max}$, $w_{\max} > 0$, *and* $\kappa > 0$. *If the admissible rollout space contains at least one group* $\mathcal{G}^\dagger$ *with equal trajectory rewards* $R_1 = \cdots = R_G$ *and at least one valid non-empty Beam miss satisfying* $0 < p_g < D$, *then*

$$\mathfrak{I}_{\mathrm{CoT}} \subsetneq \mathfrak{I}_{\mathrm{DG}}. \tag{31}$$

Proof. The non-strict inclusion follows directly from the definition of $\mathfrak{I}_{\mathrm{DG}}$, because DG-HCA retains the trajectory-level CoT branch.

For strictness, consider a witness group $\mathcal{G}^\dagger$ satisfying $R_1 = \cdots = R_G$. Then $\mu_R = R_g$ for all g, and

$$A_g = \frac{R_g - \mu_R}{\sigma_R + \varepsilon_R} = 0, \qquad g = 1, \ldots, G. \tag{32}$$

Hence $\mathcal{G}^\dagger \notin \mathfrak{I}_{\mathrm{CoT}}$.

Take a valid non-empty Beam miss in $\mathcal{G}^\dagger$ with $0 < p_g < D$, and let $d = p_g + 1$. By Lemma 1, $|\mathcal{A}_{g,d}| \ge 2$. Since $p_g > 0$, at least one Beam candidate supports $y^*_{1:p_g}$. We distinguish the two recovery cases in the definition of $e_{g,d}$. If $1 \le p_g \le D - 2$, then r_{g,p_g} is well-defined and

$$e_{g,d} = \omega(r_{g,p_g}) > 0.$$

If $p_g = D - 1$, then $n_{g,D-1} \ge 1$; together with $B \ge 1$ and $\kappa > 0$, Eq. 7 gives $q_{g,D-1} > 0$, and hence

$$e_{g,d} = q_{g,D-1} > 0.$$

Therefore, in either case, $e_{g,d} > 0$.

The local credit magnitude is

$$\gamma_{g,d} = A_{\max}\left[w_{\min} + (w_{\max} - w_{\min}) e_{g,d}\right]. \tag{33}$$

If $w_{\min} > 0$, the bracket is positive. If $w_{\min} = 0$, then $w_{\max} > 0$ and $e_{g,d} > 0$ again make it positive. Thus $\gamma_{g,d} > 0$. By Eq. 8, the target action receives positive credit and every competitor receives negative credit. Consequently, $\mathcal{G}^\dagger \in \mathfrak{I}_{\mathrm{DG}}$ while $\mathcal{G}^\dagger \notin \mathfrak{I}_{\mathrm{CoT}}$, proving the conditional strict inclusion. □

Remark 2 (Boundary cases). The strictness statement in Theorem 2 requires the existence of a witness group with equal trajectory rewards and a valid partial-prefix Beam miss. Without such a group, only $\mathfrak{I}_{\mathrm{CoT}} \subseteq \mathfrak{I}_{\mathrm{DG}}$ is guaranteed.

When $p_g = 0$, DG-HCA sets $e_{g,1} = 0$, so $\gamma_{g,1} = A_{\max} w_{\min}$. Root-level recovery can therefore provide non-zero SID credit when $w_{\min} > 0$ and the recovery group contains a competitor. For $1 \le p_g \le D - 2$, recovery evidence is positive because $\omega(r_{g,p_g}) > 0$; for $p_g = D - 1$, positivity follows from $q_{g,D-1} > 0$. If all reasoning trajectories are invalid, all trajectory-level advantages vanish and no valid Beam evidence is available for SID-level contrast.

F.3 Coordinate-Wise Sign Stability of the Clipped SID Surrogate

Recall

$$\bar{\rho} = \mathrm{clip}(\rho, 1 - \epsilon, 1 + \epsilon),$$

and

$$\Phi(\rho, A) = \min(\rho A, \bar{\rho} A). \tag{34}$$

The following result characterizes the behavior of the surrogate with respect to the scalar log-ratio coordinate $\log \rho$.

Theorem 3 (Coordinate-wise sign stability). *Assume* $0 < \epsilon < 1$ *and* $\rho > 0$.

For $A > 0$,

$$\Phi(\rho, A) = A \min\{\rho, 1 + \epsilon\}, \tag{35}$$

and

$$\frac{\partial \Phi(\rho, A)}{\partial \log \rho} = A\rho\, \mathbf{1}[\rho < 1 + \epsilon] \geq 0 \tag{36}$$

almost everywhere.

For $A < 0$,

$$\Phi(\rho, A) = A \max\{\rho, 1 - \epsilon\}, \tag{37}$$

and

$$\frac{\partial \Phi(\rho, A)}{\partial \log \rho} = A\rho\, \mathbf{1}[\rho > 1 - \epsilon] \leq 0 \tag{38}$$

almost everywhere.

Thus, the clipped surrogate preserves the update direction implied by the sign of the assigned credit in the corresponding log-ratio coordinate. Positive credit yields a non-decreasing surrogate up to the upper clipping boundary, whereas negative credit yields a non-increasing surrogate above the lower clipping boundary.

Proof. Let

$$\ell = 1 - \epsilon, \qquad u = 1 + \epsilon.$$

For $A > 0$, multiplication by A preserves order. Therefore,

$$\begin{aligned} \Phi(\rho, A) &= \min(\rho A, \bar{\rho} A) \\ &= A \min(\rho, \bar{\rho}) \\ &= A \min\{\rho, u\}. \end{aligned} \tag{39}$$

Hence $\Phi(\rho, A)$ is non-decreasing in ρ, and its derivative with respect to $\log \rho$ is

$$\frac{\partial \Phi(\rho, A)}{\partial \log \rho} = A\rho\, \mathbf{1}[\rho < u] \geq 0 \tag{40}$$

almost everywhere.

For $A < 0$, multiplication by A reverses order, yielding

$$\begin{aligned} \Phi(\rho, A) &= \min(\rho A, \bar{\rho} A) \\ &= A \max(\rho, \bar{\rho}) \\ &= A \max\{\rho, \ell\}. \end{aligned} \tag{41}$$

The function $\max\{\rho, \ell\}$ is non-decreasing in ρ. Since $A < 0$, $\Phi(\rho, A)$ is non-increasing, and

$$\frac{\partial \Phi(\rho, A)}{\partial \log \rho} = A\rho\, \mathbf{1}[\rho > \ell] \leq 0 \tag{42}$$

almost everywhere.

This proves the stated coordinate-wise sign behavior. □

Remark 3 (Coordinate space versus parameter space). Theorem 3 concerns a single surrogate term as a function of its own coordinate $\log \rho$. In the neural policy, action log-probabilities share parameters and the total gradient also includes other SID actions, CoT tokens, and KL regularization. The result therefore does not imply monotonic changes in every action probability after a shared-parameter gradient step.

F.4 Local Contrastive Geometry of SID Credit

Consider a valid local action group with $K = |\mathcal{A}_{g,d}| \geq 2$, target action $a^* = a^*_{g,d}$, and credit magnitude $\gamma = \gamma_{g,d} > 0$. Eq. 8 gives $\Gamma(a^*) = \gamma$ and $\Gamma(a) = -\gamma/(K-1)$ for every $a \neq a^*$.

Define $s_a = \ell_\theta(a \mid x_{g,d}) - \ell_{\theta_{\text{old}}}(a \mid x_{g,d})$, so that $\rho(a) = e^{s_a}$. The local SID maximization surrogate is

$$J_{g,d}(s) = \sum_{a \in \mathcal{A}_{g,d}} \Phi(e^{s_a}, \Gamma(a)). \tag{43}$$

Proposition 1 (Local centered contrastive geometry)**.** *Treating $s = (s_a)_{a \in \mathcal{A}_{g,d}}$ as local surrogate coordinates, in a sufficiently small neighborhood of the old policy $s = 0$,*

$$J_{g,d}(s) = \gamma \left[s_{a^*} - \frac{1}{K-1} \sum_{a \neq a^*} s_a \right] + O(\|s\|^2). \tag{44}$$

Hence the first-order local surrogate geometry favors the target action log-ratio relative to the mean competitor log-ratio.

Proof. The raw credit is zero-sum:

$$\sum_a \Gamma(a) = \gamma + (K-1)\left(-\frac{\gamma}{K-1}\right) = 0.$$

At the old policy, $s_a = 0$ and hence $\rho(a) = e^{s_a} = 1$. Since $0 < \epsilon < 1$, the ratio $\rho = 1$ lies in the interior of the clipping interval $[1-\epsilon, 1+\epsilon]$. Therefore, there exists a sufficiently small neighborhood of $s = 0$ in which clipping is inactive for every local action. In this neighborhood,

$$\Phi(e^{s_a}, \Gamma(a)) = e^{s_a}\Gamma(a).$$

Using

$$e^{s_a} = 1 + s_a + O(s_a^2),$$

we obtain

$$\begin{aligned} J_{g,d}(s) &= \sum_a \Gamma(a)\left[1 + s_a + O(s_a^2)\right] \\ &= \sum_a \Gamma(a) s_a + O(\|s\|^2) \\ &= \gamma s_{a^*} - \frac{\gamma}{K-1} \sum_{a \neq a^*} s_a + O(\|s\|^2). \end{aligned} \tag{45}$$

□

Corollary 3 (Target-versus-competitor ascent direction in score space)**.** *For any competitor $a \neq a^*$, consider at $s = 0$ the score-coordinate direction $s_{a^*} \mapsto s_{a^*} + \delta$ and $s_a \mapsto s_a - \delta$. The corresponding directional derivative is*

$$\gamma + \frac{\gamma}{K-1} = \gamma \frac{K}{K-1} > 0. \tag{46}$$

Remark 4. Corollary 3 describes the local surrogate geometry in score coordinates. It does not assert that this direction is exactly realizable by an infinitesimal perturbation of shared Transformer parameters.

Corollary 4 (Fixed assigned credit budget)**.** *For every valid local group,*

$$\sum_{a \neq a^*} |\Gamma(a)| = \gamma, \qquad \|\Gamma\|_1 = 2\gamma. \tag{47}$$

Thus the assigned negative raw credit mass is invariant to the number of competing actions.

Remark 5. Corollary 4 concerns the raw credit assignment in Eq. 8. Action-dependent policy ratios, clipping, and shared policy parameters generally destroy the zero-sum structure at the surrogate-value or parameter-gradient level.

F.5 Stability of Length-Normalized Recovery Policy Ratios

A recovery action starting at depth d is $a = y^*_{d:D}$ and has length $L_d = D - d + 1$. Recovery lengths therefore vary with the recovery depth. For one fixed recovery action, write $L = L_d$ and $a = (a_1, \ldots, a_L)$.

Define the token-level conditional policy ratio

$$r_k = \frac{\pi_\theta(a_k \mid x, a_{<k})}{\pi_{\theta_{\text{old}}}(a_k \mid x, a_{<k})}, \qquad k = 1, \ldots, L. \tag{48}$$

The ordinary sequence likelihood ratio is $\rho_{\mathrm{raw}} = \pi_\theta(a \mid x)/\pi_{\theta_{\mathrm{old}}}(a \mid x) = \prod_{k=1}^{L} r_k$. DG-HCA uses the length-normalized policy ratio

$$\rho_{\mathrm{DG}} = \exp\left[\frac{1}{L}\log\frac{\pi_\theta(a \mid x)}{\pi_{\theta_{\mathrm{old}}}(a \mid x)}\right]. \tag{49}$$

For $L > 1$, ρ_{DG} is a geometric-mean policy ratio rather than the ordinary sequence likelihood ratio.

Lemma 2 (Geometric-mean contraction of the length-normalized policy ratio)**.** *For any recovery action of length $L \geq 1$,*

$$\rho_{\mathrm{DG}} = \left(\prod_{k=1}^{L} r_k\right)^{1/L} = \rho_{\mathrm{raw}}^{1/L}. \tag{50}$$

Consequently:

Geometric-mean bound.

$$\min_k r_k \leq \rho_{\mathrm{DG}} \leq \max_k r_k. \tag{51}$$

Conditional log-ratio variance scaling. *Conditioned on a fixed action length L, whenever the variance exists,*

$$\mathrm{Var}\left[\log \rho_{\mathrm{DG}} \mid L\right] = \frac{1}{L^2}\mathrm{Var}\left[\log \rho_{\mathrm{raw}} \mid L\right]. \tag{52}$$

Clipping-event containment. *Let $C_\epsilon = [1-\epsilon, 1+\epsilon]$. Then, for every $L \geq 1$,*

$$\left\{\rho_{\mathrm{DG}} \notin C_\epsilon\right\} \subseteq \left\{\rho_{\mathrm{raw}} \notin C_\epsilon\right\}. \tag{53}$$

Token-level ratio-interval inheritance. *If $r_k \in C_\epsilon$ for every k, then $\rho_{\mathrm{DG}} \in C_\epsilon$, while*

$$\rho_{\mathrm{raw}} \in \left[(1-\epsilon)^L, (1+\epsilon)^L\right]. \tag{54}$$

Proof. Autoregressive factorization gives $\pi_\theta(a \mid x)/\pi_{\theta_{\mathrm{old}}}(a \mid x) = \prod_{k=1}^{L} r_k$, which yields

$$\rho_{\mathrm{DG}} = \left(\prod_{k=1}^{L} r_k\right)^{1/L} = \rho_{\mathrm{raw}}^{1/L}. \tag{55}$$

The geometric-mean bound follows immediately.

For fixed L, $\log \rho_{\mathrm{DG}} = (1/L)\log \rho_{\mathrm{raw}}$, giving the conditional variance identity without any independence assumption among the token-level ratios.

For clipping-event containment, $x^{1/L}$ moves every positive x toward one: if $x \geq 1$, then $1 \leq x^{1/L} \leq x$, while if $0 < x \leq 1$, then $x \leq x^{1/L} \leq 1$. Thus $x \in C_\epsilon$ implies $x^{1/L} \in C_\epsilon$, and the event containment follows by contraposition.

Finally, if every $r_k \in C_\epsilon$, their geometric mean belongs to C_ϵ. Multiplying the token-wise bounds gives the stated interval for ρ_{raw}. □

Remark 6 (Variable recovery lengths). Although L is fixed within each application of Lemma 2, it varies across recovery depths through $L_d = D - d + 1$. The normalization satisfies

$$\log \rho_{\mathrm{DG}} = \frac{1}{L}\sum_{k=1}^{L} \log r_k, \tag{56}$$

so actions of different lengths are represented by an average per-token log-ratio instead of a product whose scale accumulates with length. This normalization does not imply identical distributions or variances across different values of L.

If recovery lengths vary across samples, the unconditional variance does not follow a single $1/L^2$ scaling law. By the law of total variance,

$$\begin{aligned}\mathrm{Var}(\log \rho_{\mathrm{DG}}) = \mathbb{E}\left[\frac{1}{L^2}\,\mathrm{Var}(\log \rho_{\mathrm{raw}} \mid L)\right] \\ + \mathrm{Var}\left(\frac{1}{L}\mathbb{E}[\log \rho_{\mathrm{raw}} \mid L]\right).\end{aligned} \tag{57}$$

The recovery weight $\lambda_{g,d} = \sum_{r=d}^{D} \lambda_r$ aggregates the base weights of the hierarchy levels represented by a multi-token recovery suffix. It is complementary to the ratio normalization and is not used as a variance correction for ρ_{DG}.

Remark 7 (Single-token case). When $L = 1$, $\rho_{\mathrm{DG}} = \rho_{\mathrm{raw}} = r_1$. The variance scaling factor is one, clipping-event containment becomes equality, and the interval statement reduces to the standard single-token case.

Credit magnitude bounds. Since $0 \le e_{g,d} \le 1$,

$$A_{\max} w_{\min} \le \gamma_{g,d} \le A_{\max} w_{\max}. \tag{58}$$

Summary. Theorem 1 characterizes the ranking-driven reasoning reward as the single-target NDCG signal plus a bounded valid-miss refinement and an invalidity penalty. Theorem 2 shows that hierarchical SID credit can provide supervision when trajectory-level group contrast vanishes, with strict expansion under an explicit witness-existence condition. Theorem 3 establishes coordinate-wise sign stability of the clipped SID surrogate, and Proposition 1 identifies its local centered target-versus-competitor geometry. Finally, Lemma 2 characterizes the geometric-mean normalization used for recovery actions and its behavior under variable action lengths.

G Discussions

G.1 Why Does Beam Evidence Switch at $p_g = D - 1$?

DG-HCA uses the rank evidence $\omega(r_{g,p_g})$ when $1 \le p_g \le D-2$, while switching to the aggregate prefix support $q_{g,D-1}$ when $p_g = D-1$. This distinction follows from a structural change in the remaining search space rather than from a depth-specific heuristic.

Let $\mathcal{S}_{g,p} = \left\{j : \tilde{y}_{g,j,1:p} = y^*_{1:p}\right\}$ denote the set of Beam candidates that share the target prefix $y^*_{1:p}$. We further define the set of distinct actions immediately following this prefix as $\mathcal{Z}_{g,p} = \mathrm{Unique}\left\{\tilde{y}_{g,j,p+1} : j \in \mathcal{S}_{g,p}\right\}$. When $p \le D-2$, at least two SID levels remain after the covered target prefix. Two different Beam candidates may therefore take the same action at depth $p+1$ and differ only at deeper levels. In particular, it is possible to have

$$\tilde{y}_{g,j_1,p+1} = \tilde{y}_{g,j_2,p+1}, \qquad \tilde{y}_{g,j_1} \ne \tilde{y}_{g,j_2}. \tag{59}$$

Hence the mapping from supporting Beam candidates to frontier actions can be many-to-one, which gives $|\mathcal{Z}_{g,p}| \le |\mathcal{S}_{g,p}|$, with strict inequality possible.

This observation implies that the number of Beam candidates supporting $y^*_{1:p}$ may also reflect downstream branching after the recovery frontier. Consequently, candidate multiplicity does not directly measure the strength of the local recovery decision when multiple SID levels remain. For all cases satisfying $1 \le p_g \le D-2$, we therefore use $e_{g,p_g+1} = \omega(r_{g,p_g})$, which measures the ranking quality of the strongest Beam path reaching the deepest covered target prefix. The same form can be used throughout this interval because downstream branching remains possible whenever the recovery suffix contains at least two SID levels.

The structure changes when $p_g = D-1$. Every supporting candidate then has the form $\tilde{y}_{g,j} = \left(y^*_{1:D-1}, \tilde{y}_{g,j,D}\right)$. No deeper SID level remains. If duplicate SIDs are removed from the Beam, two distinct candidates sharing $y^*_{1:D-1}$ must correspond to distinct final-level actions. Therefore, $|\mathcal{Z}_{g,D-1}| = |\mathcal{S}_{g,D-1}|$. In this case, candidate multiplicity is no longer affected by deeper branching and directly reflects the local search support under the same target parent state. This motivates

replacing the best-path evidence with the aggregate prefix support $e_{g,D} = q_{g,D-1}$, which combines the coverage of the target prefix with the ranking quality of its supporting candidates.

Therefore, $p_g = D - 1$ forms a natural structural boundary. Before this point, recovery remains a multi-level decision and Beam multiplicity may be influenced by downstream search expansion. At $p_g = D - 1$, recovery reduces to a single SID decision and the supporting candidates become directly comparable at the local action level. The evidence rule is changed accordingly to match the information exposed by Beam Search.

G.2 Why Does SID Credit Use a Zero-Sum Assignment?

The zero-sum design of SID-level credit is mainly intended to control the supervision scale across local action groups of different sizes. Consider a valid local action group with $K = |\mathcal{A}_{g,d}| \geq 2$, target action a^*, and credit magnitude $\gamma > 0$. DG-HCA assigns $\Gamma(a^*) = \gamma$, while the remaining $K - 1$ competing actions equally share a total negative credit of the same magnitude,

$$\Gamma(a) = -\frac{\gamma}{K-1}, \qquad a \neq a^*. \tag{60}$$

This construction keeps the total negative credit independent of the number of competing actions. As shown in Corollary 4,

$$\sum_{a \neq a^*} |\Gamma(a)| = \gamma, \qquad \|\Gamma\|_1 = 2\gamma. \tag{61}$$

Hence, local action groups with different numbers of Beam competitors receive raw credit budgets on the same scale.

This property is particularly relevant for Beam-based local optimization. The size of a local action group is affected not only by the underlying SID decision, but also by how many alternatives are explored by Beam Search under the corresponding parent state. If every competing action were assigned a fixed negative credit with the same magnitude, groups containing more competitors would naturally accumulate stronger negative supervision. In that case, the optimization strength would depend partly on the number of explored branches rather than on the quality of the local SID decision. The zero-sum assignment removes this dependence by fixing the total positive and negative credit budget within each local group.

The same construction also induces a centered local contrastive structure. Proposition 1 shows that, in a sufficiently small neighborhood of the old policy, the local SID surrogate admits the first-order expansion

$$J_{g,d}(s) = \gamma \left[s_{a^*} - \frac{1}{K-1} \sum_{a \neq a^*} s_a \right] + O(\|s\|^2). \tag{62}$$

Therefore, the local update favors the target action relative to the average competing action, rather than accumulating an increasingly large penalty as the number of competitors grows. This is consistent with the local SID decision perspective of DG-HCA, where actions sharing the same target parent are compared according to their relative preference.

From this perspective, the zero-sum assignment serves two complementary purposes. First, it normalizes the credit budget across local action groups and prevents groups with more Beam branches from receiving disproportionately strong negative supervision. Second, it yields a centered target-versus-competitor objective whose first-order behavior depends on the relative separation between the target action and the mean competitor.

The zero-sum property applies to the raw SID credit assignment itself. After entering policy optimization, different actions have different policy ratios and are jointly affected by clipping and shared model parameters. Therefore, the surrogate values and parameter gradients are not required to preserve an exact zero-sum relation, as discussed in Remark 5.

G.3 How Does Skipping Stage 2 Reasoning GRPO Affect Model Performance?

We employ Stage 2 Reasoning GRPO as a key step in eliciting task-relevant reasoning. Through rapid convergence, this stage yields a checkpoint with preliminary reasoning capabilities for the

downstream task, providing an initialization for subsequent training. However, bypassing Stage 2 and applying Stage 3 DG-HCA directly to the Stage 1 checkpoint leads to a decline in model performance. A plausible explanation is that the Chain-of-Thought (CoT) reasoning learned in Stage 1 is not yet sufficiently aligned with the final SID ranking, while the reward in Stage 3 depends on the Beam Search outputs induced by this reasoning.

These observations suggest that Stage 2 primarily provides a reasoning-policy initialization suitable for Hierarchical Credit Assignment. In contrast, Stage 3 is better suited to refining reasoning trajectories that already contain target-relevant evidence than to establishing fully developed CoT reasoning capabilities from a standard Supervised Fine-Tuning (SFT) policy.

Table 9: **Effect of Stage 2 Reasoning GRPO.** We compare the full training pipeline with a variant that skips Stage 2 Reasoning GRPO and directly applies Stage 3 DG-HCA after Stage 1.

Dataset	R@5	N@5	R@10	N@10
Office	0.1307	0.1060	0.1545	0.1138
Industrial	0.1052	0.0826	0.1306	0.0907
Games	0.0532	0.0354	0.0892	0.0470

H NOTATION

See Table 10.

I CASE STUDY

See Figure 7, Figure 8, and Figure 9.

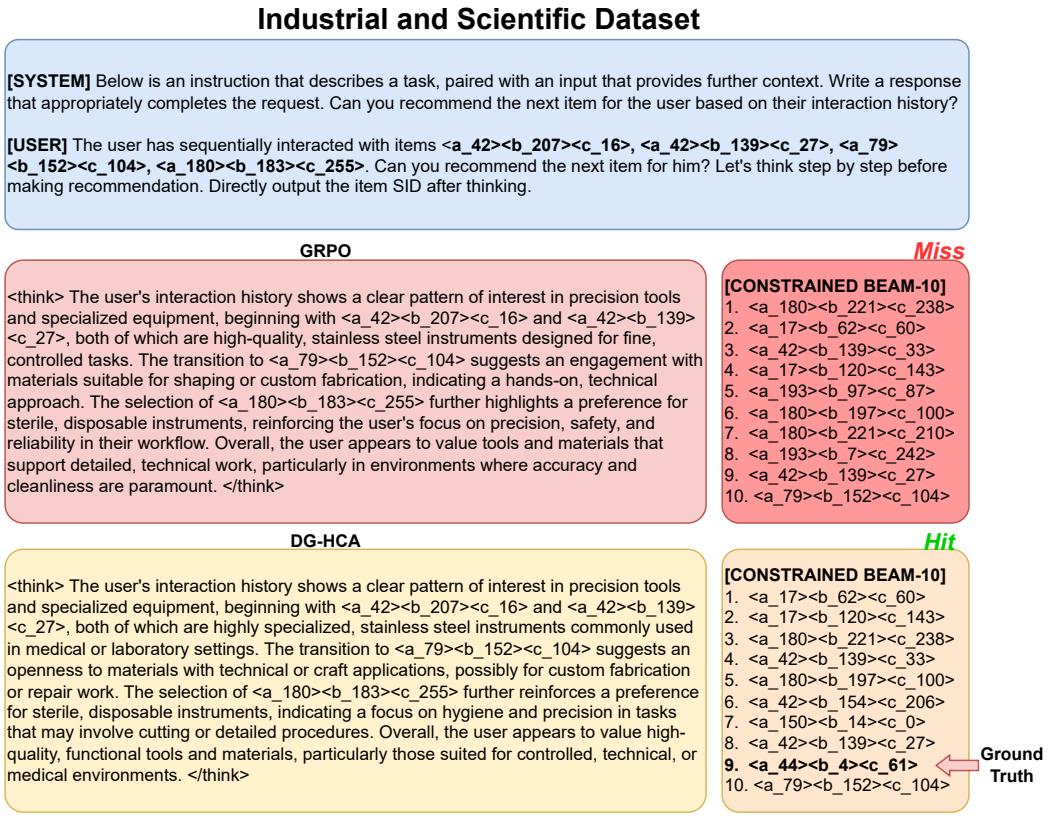


Figure 7: Comparison of outcomes between DG-HCA and GRPO on Industrial and Scientific test set.

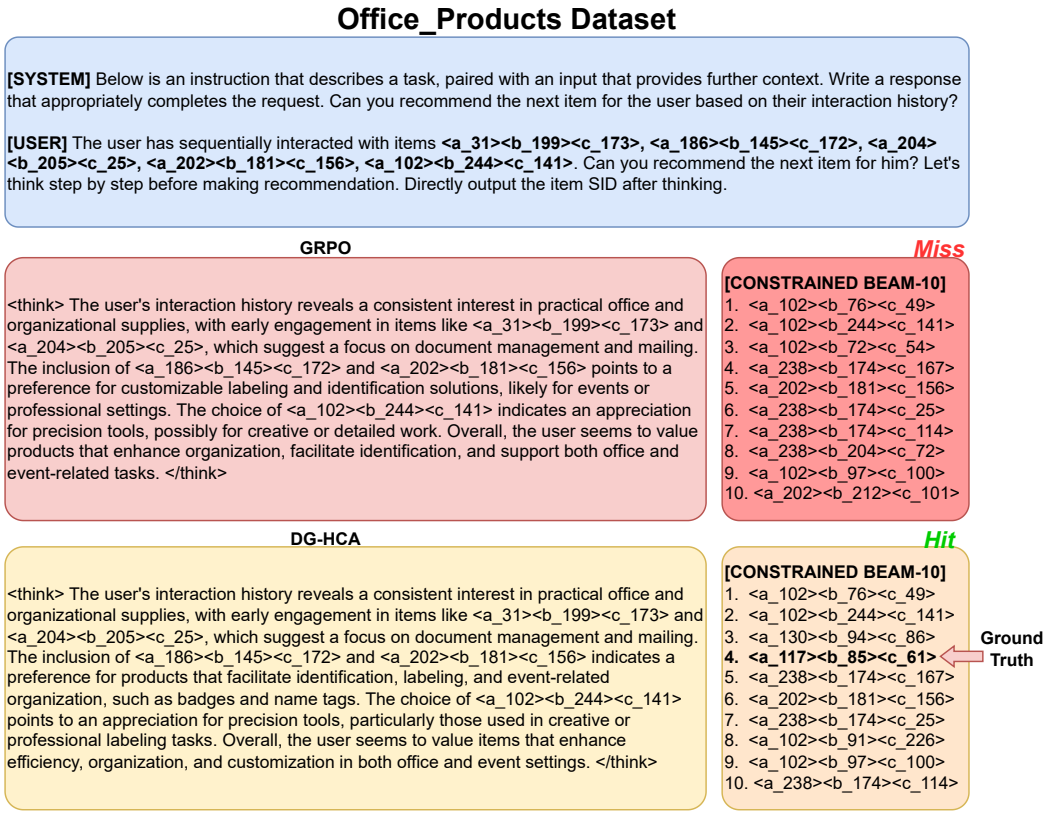


Figure 8: Comparison of outcomes between DG-HCA and GRPO on Office Products test set.

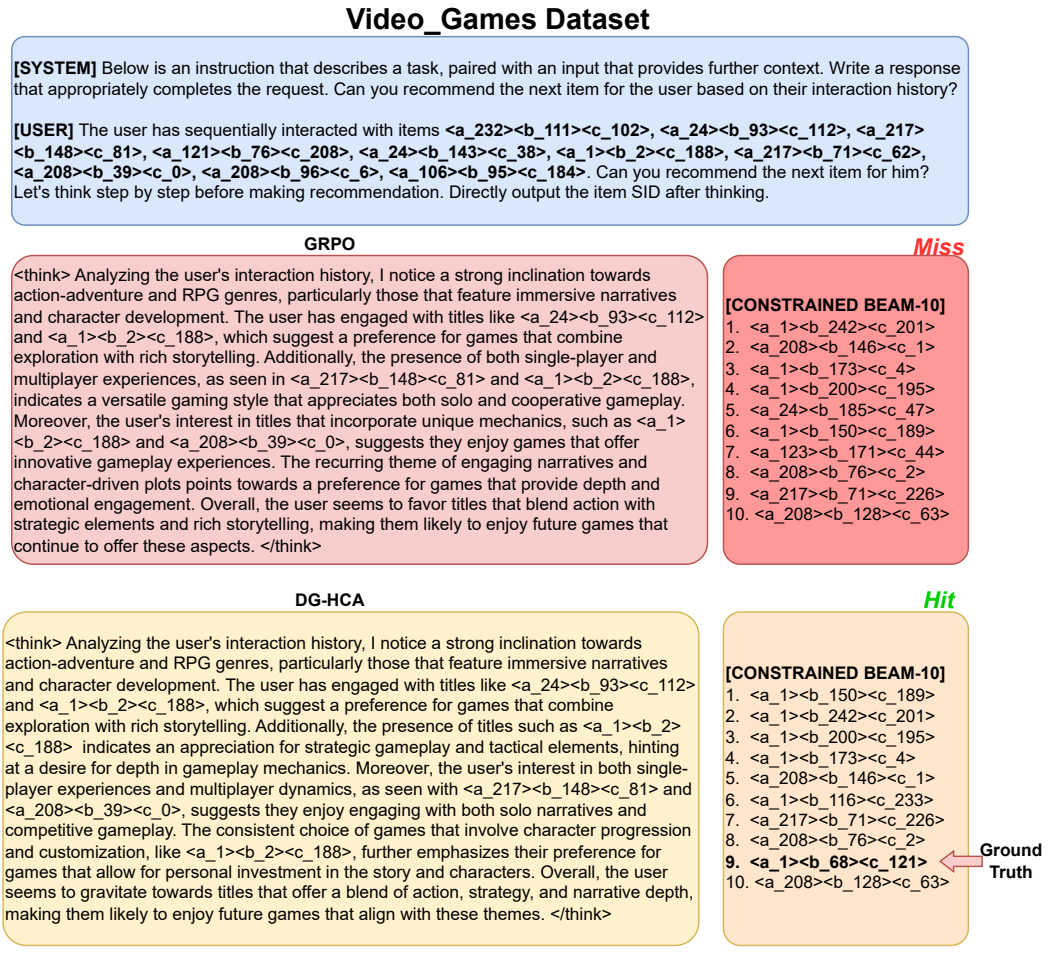


Figure 9: Comparison of outcomes between DG-HCA and GRPO on Video Games test set.

Table 10: Core notation used in the paper.

Symbol	Definition
n, d, g, j	Indices of the training instance, SID hierarchy level, rollout for the same instance, and ranked Beam candidate, respectively.
D	Number of hierarchical levels in an SID.
$\mathcal{H}$	User interaction history of the current training instance.
$y^* = (y_1^*, \ldots, y_D^*)$	Target SID of the current training instance.
τ_g / G	The g-th reasoning trajectory / number of trajectories sampled for the same user history.
$\pi_\theta, \pi_{\theta_{\text{old}}}, \pi_{\text{ref}}$	Current policy, old policy used for rollout generation and Beam Search, and reference policy used for KL regularization.
$\mathcal{B}_g$ / B_g / B	Ranked Beam list for trajectory g / number of returned Beam candidates, $B_g = \lvert\mathcal{B}_g\rvert$ / maximum Beam width.
$\tilde{y}_{g,j}$	Candidate SID ranked at position j in $\mathcal{B}_g$.
$\omega(r)$	Rank discount, $\omega(r) = 1/\log_2(r+1)$.
$m(\tilde{y}, y^*)$	Length of the longest common prefix between candidate $\tilde{y}$ and target SID y^*.
$\psi(\tilde{y}, y^*)$	Hierarchical prefix score of a non-exact candidate.
R_{inv}	Reward assigned to an invalid reasoning trajectory, with $R_{\text{inv}} < 0$.
R_g / A_g	Ranking-driven CoT reward for trajectory g / corresponding group-normalized CoT advantage in Stage 3.
μ_R / σ_R	Mean / standard deviation of the trajectory rewards in the rollout group.
ε_R	Positive numerical stabilizer used in the group-normalized CoT advantage.
p_g	Deepest target-prefix depth covered by the Beam of trajectory g.
$\mathcal{A}_{g,d}$ / $a_{g,d}^*$	Local SID action group at depth d / target action in that group.
$r_{g,d}$	Highest Beam rank at which the target prefix $y_{1:d}^*$ appears.
$\mathcal{S}_{g,h}$ / $n_{g,h}$	Set / number of Beam candidates supporting target prefix $y_{1:h}^*$.
$q_{g,h}$	Prefix-support score computed from the coverage and ranking quality of $\mathcal{S}_{g,h}$.
κ	Positive saturation parameter in the prefix-support score.
$e_{g,d}$ / $\gamma_{g,d}$	Beam evidence at depth d / corresponding SID-credit magnitude.
$A_{\max}, w_{\min}, w_{\max}$	Hyperparameters controlling the range of the SID-credit magnitude.
$\Gamma_{g,d}(a)$	SID credit assigned to action $a \in \mathcal{A}_{g,d}$.
$x_{g,d}$	Conditioning context for the SID action at depth d, $x_{g,d} = (\mathcal{H}, \tau_g, y_{<d}^*)$.
$\ell_\theta(a \mid x)$	Length-normalized log-probability of SID action a under policy π_θ.
$\rho_{g,t}$	Token-level policy ratio for the t-th CoT token of trajectory g.
$\rho_{g,d}(a)$	Length-normalized policy ratio for SID action a at depth d.
$\bar{\rho}$	Clipped policy ratio, $\bar{\rho} = \text{clip}(\rho, 1-\epsilon, 1+\epsilon)$.
ϵ	Clipping threshold used in the policy surrogate.
$\Phi(\rho, A)$	Clipped surrogate function used for positive and negative credit.
$\mathcal{V}_g$	Set of starting depths of valid SID action groups for trajectory g.
λ_d / $\lambda_{g,d}$	Base weight for hierarchy depth d / effective weight assigned to the local SID action group.
η	Relative weight of the SID-level branch in the Stage 3 objective.
β_{KL}	Weight of KL regularization toward the reference policy.
$\mathcal{L}_{\text{DG-HCA}}$	Joint optimization objective of Stage 3.
N_B	Per-instance single-target retrieval score, equal to $\text{NDCG@}B$ when there is one relevant target item.
R_B	Random variable corresponding to the raw ranking-driven reasoning reward.
$\mathsf{H}, \mathsf{M}, \mathsf{I}$	Events of a valid exact Top-B hit, a valid miss, and an invalid reasoning trajectory, respectively.
δ_B	Upper bound on the valid-miss reward, $\delta_B = \omega(B)/2$.
α	Magnitude of the invalid-trajectory penalty, $\alpha = -R_{\text{inv}} > 0$.
$J_R(\pi)$ / $J_N(\pi)$	Expected ranking-driven reward / expected single-target retrieval score under policy π.
$S(\pi)$	Expected auxiliary reward contributed by valid misses.
$p_{\text{inv}}(\pi)$	Probability that a reasoning trajectory sampled under π is invalid.
ξ / Π_ξ	Invalid-trajectory probability tolerance / restricted policy class $\{\pi \in \Pi : p_{\text{inv}}(\pi) \le \xi\}$.
J_N^* / π_R	Best expected retrieval score over Π_ξ / a policy maximizing J_R over Π_ξ.
s_a	Local log-ratio coordinate, $s_a = \ell_\theta(a \mid x) - \ell_{\theta_{\text{old}}}(a \mid x)$.
$J_{g,d}(s)$	Local SID maximization surrogate expressed in the score coordinates s.
L_d	Length of a recovery suffix starting at depth d, $L_d = D - d + 1$.
r_k	Token-level conditional policy ratio for the k-th token of a recovery action.
ρ_{raw} / ρ_{DG}	Ordinary sequence likelihood ratio / length-normalized recovery policy ratio.
C_ϵ	Clipping interval $[1-\epsilon, 1+\epsilon]$.