Stable and Granular Policy Optimization for Generative Recommendation

Muyu Zou[1*] **Haibo Xing**[1*] **Hao Deng**[1*] **Zhezheng Hao**[2*] **Jinxin Hu**[1,†]

Jiawei Chen[2,†] **Yun Chen**[1] **Yu Zhang**[1] **Xiaoyi Zeng**[1]

[1]**Alibaba Group, Beijing, China** [2]**Zhejiang University, Hangzhou, China**
{zoumuyu.zmy, xinghaibo.xhb, denghao.deng}@alibaba-inc.com
{jinxin.hjx, jinuo.cy}@alibaba-inc.com
haozhezheng@outlook.com, sleepyhunt@zju.edu.cn
daoji@lazada.com, yuanhan@taobao.com
[*]Equal contribution. [†]Corresponding author.

Abstract

Generative recommendation (GR), which directly generates sequential semantic IDs for items, has recently attracted increasing attention in recommender systems. Given that core recommendation metrics are inherently evaluated at the sequence level, reinforcement learning (RL) serves as a natural optimization framework. However, adapting recent group-based RL algorithms to GR presents notable challenges: token-level methods such as GRPO exhibit much more severe training instability, while sequence-level methods such as GSPO enforce uniform token weighting and thus ignore the heterogeneous roles of tokens. To address this dilemma, we propose Stable and Granular Policy Optimization (SGPO), a group-based RL algorithm tailored for GR. SGPO maintains a sequence-level ratio to promote optimization stability while injecting fine-grained token-wise advantages. By incorporating token log-probabilities and reward signs, SGPO explicitly up-weights high-confidence tokens in positive trajectories to reinforce successful retrieval, while strictly penalizing them in negative trajectories to encourage exploration, and uses an adaptive scaling factor to bound the variance of reshaped advantages. Experiments on two public benchmarks and a billion-scale industrial dataset show that SGPO consistently outperforms six strong RL baselines and, when deployed on a commercial platform, achieves a 2.36% lift in click-through rate (CTR) and a 1.95% increase in advertising revenue in online A/B testing.

1 Introduction

Generative recommendation (GR) with Semantic IDs (SIDs) has recently been proposed as a scalable paradigm for recommender systems [1–7]. In this framework, each item is encoded as a fixed-length SID sequence that corresponds to a path in a tree-structured codebook [1, 8–11] (left panel of Figure 1), and a GR model is trained to autoregressively produce the SID sequence of a target item.

Notably, core recommendation metrics (*e.g.*, click-through rate) are evaluated at the item (sequence) level—focusing on whether the generated SID sequence resolves to the correct item rather than per-token accuracy. This sequence-level objective is aligned with reinforcement learning (RL) [12, 13], which optimizes rewards over sequences. Recently, group-based RL algorithms [14–18] have gained traction in LLM training, with two representative paradigms: (i) GRPO [14], which samples response groups and computes group-wise advantages. GRPO optimizes the policy at the token level and

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

introduces token-wise importance ratios to correct the mismatch between the sampling and target policies. (ii) GSPO [15], which replaces GRPO's token-level ratios with a smoothed sequence-level ratio, thereby reducing variance and instability caused by per-token ratios.

Although group-based RL holds substantial promise for GR, adapting these algorithms presents significant challenges due to two distinct characteristics of GR: (i) *GR exhibits substantially stronger training instability*. We analyze the training gradients of GRPO and GSPO in GR scenarios, comparing them against standard LLM training (Figure 2). We observe that the variance of gradient norms in GR is significantly higher. In particular, for GRPO, token-level ratios can induce high-variance and unstable optimization behavior [15, 19, 20]. While recent studies [2, 3, 21–24] have attempted to adapt GRPO for GR via reward engineering or trajectory grouping, these ad-hoc solutions do not revise the fundamental optimization objective, ultimately yielding sub-optimal performance. (ii) *GR requires heterogeneous token weighting.* In GR, intra-sequence tokens correspond to different semantic levels in the codebook and thus naturally exhibit heterogeneous roles [8, 25–28]. Our fine-grained analysis reveals the critical need for heterogeneous, asymmetric token weighting: high-probability tokens in positive trajectories must be upweighted to ensure their successful retrieval during beam search, whereas high-probability tokens in negative trajectories require stronger penalties to enforce exploration and steer the policy away from incorrect sub-trees (Figure 3). Consequently, while GSPO achieves gradient stability, its uniform treatment of intra-sequence tokens renders it strictly sub-optimal for GR.

Driven by the unique characteristics of GR and the limitations of existing methods, we propose **S**table and **G**ranular **P**olicy **O**ptimization (**SGPO**), a group-based RL framework explicitly tailored for generative recommendation. SGPO inherits the sequence-level ratio of GSPO to guarantee stable optimization, yet overcomes its limitations by injecting fine-grained token-wise advantages. Specifically, SGPO explicitly incorporates the token's predicted probability and the sign of the reward into its formulation, adaptively highlighting high-confidence tokens during optimization as desired. Furthermore, SGPO derives an adaptive scaling factor to dynamically regulate advantage magnitudes, preventing numerical instability and theoretically controlling advantage variance.

We evaluate SGPO on two standard GR benchmarks and one large-scale industrial dataset with billion-level interactions, using both TIGER [1] and REG4Rec [2] as backbones. SGPO consistently outperforms six strong RL baselines. Further analysis shows that SGPO yields smoother and more stable training dynamics. **We have deployed SGPO in an online A/B test on the industrial platform**, where it achieves a 2.36% lift in click-through rate (CTR) and a 1.95% increase in advertising revenue, demonstrating its practical value for real-world recommender systems.

Our contributions are summarized as follows:

- We identify two key challenges in adapting group-based RL to SID-based GR: (i) markedly higher training instability in GR when using token-level importance ratios, and (ii) the need for heterogeneous token weighting.
- We propose **SGPO**, a group-based RL framework for GR that retains a GSPO-style sequence-level ratio for stability. It introduces probability–aware token-wise advantage reshaping for fine-grained credit assignment. An adaptive scaling factor further controls update magnitude and variance.
- Extensive experiments on two public GR benchmarks and a large-scale industrial dataset show that SGPO consistently outperforms six strong RL baselines and improves training stability. The online A/B test further confirms its practical gains in CTR and advertising revenue.

2 Preliminaries

2.1 Generative Recommendation

GR formulates item recommendation as an autoregressive generation task [1]. Given a user query (or context) q from a query set $\mathcal{D}$, a model parameterized by θ defines a stochastic policy π_θ. The policy predicts a target item as a sequence of discrete SIDs $y = (y_1, \ldots, y_T)$, where T is the SID sequence length. The likelihood of a sequence y is factorized as

$$\pi_\theta(y \mid q) = \prod_{t=1}^{|y|} \pi_\theta(y_t \mid q, y_{<t}), \tag{1}$$

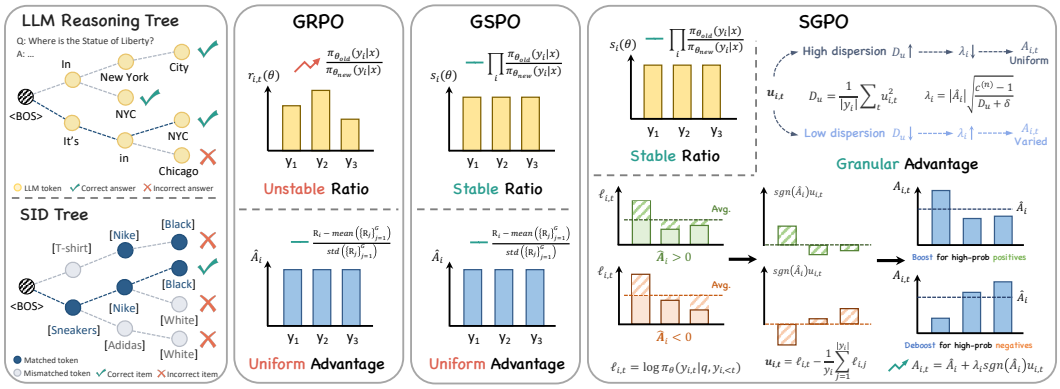

Figure 1: Overview of SGPO for SID-based generative recommendation. Left: each item has a unique SID path in the tree, so retrieval is prefix-sensitive and an incorrect prefix leads to an incorrect item. Right: SGPO combines a GSPO-style sequence-level ratio with log-probability–aware token-level advantage reshaping, with adaptive scaling factor λ for update control.

where $y_{<t}$ denotes the previously generated tokens. After generation, the SID sequence is mapped to an item via a predefined hierarchical structure.

2.2 Reinforcement Learning for Sequence Optimization

Given that the core metrics of recommendation (*e.g.*, click-through rate) [2, 12, 29] are inherently evaluated at the item (sequence) level rather than the token level, reinforcement learning (RL) emerges as a natural optimization paradigm for recommendation. Recently, Group-based RL is gained popularity by eliminating the auxiliary value model. Specifically, for each query q, we independently sample a group of G responses $\{y_1, \ldots, y_G\}$ from a reference policy $\pi_{\theta_{\text{old}}}$. Let R_i be the scalar reward associated with the i-th response y_i. To reduce variance, we compute the group-normalized advantage $\hat{A}_i$ as

$$\hat{A}_i = \frac{R_i - \text{mean}(\{R_j\}_{j=1}^G)}{\text{std}(\{R_j\}_{j=1}^G)}, \tag{2}$$

where $\text{mean}(\cdot)$ and $\text{std}(\cdot)$ denote the mean and standard deviation of rewards in the group, respectively. Recent literature has witnessed two representative lines of group-based RL methods:

Group Relative Policy Optimization (GRPO). GRPO [14] optimizes the policy by maximizing the token-level objective (omitting the KL term for brevity):

$$J_{\text{GRPO}}(\theta) = \mathbb{E}_{q\sim\mathcal{D},\{y_i\}\sim\pi_{\theta_{\text{old}}}} \left[\frac{1}{G}\sum_{i=1}^{G} \frac{1}{|y_i|}\sum_{t=1}^{|y_i|} \min\left(r_{i,t}(\theta)\hat{A}_i, \text{clip}(r_{i,t}(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_i \right) \right], \tag{3}$$

where $r_{i,t}(\theta) = \frac{\pi_\theta(y_{i,t}|q,y_{i,<t})}{\pi_{\theta_{\text{old}}}(y_{i,t}|q,y_{i,<t})}$ is the importance sampling ratio at token t, $\epsilon > 0$ is the clipping hyperparameter, and $\hat{A}_i$ is shared by all tokens in response y_i.

Group Sequence Policy Optimization (GSPO). GSPO [15] turns to optimize a sequence-level objective (omitting the KL term for brevity):

$$J_{\text{GSPO}}(\theta) = \mathbb{E}_{q\sim\mathcal{D},\{y_i\}\sim\pi_{\theta_{\text{old}}}} \left[\frac{1}{G}\sum_{i=1}^{G} \min\left(s_i(\theta)\hat{A}_i, \text{clip}(s_i(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_i \right) \right], \tag{4}$$

where $s_i(\theta)$ is the length-normalized sequence ratio defined as

$$s_i(\theta) = \exp\left(\frac{1}{|y_i|}\sum_{t=1}^{|y_i|} \log \frac{\pi_\theta(y_{i,t} \mid q, y_{i,<t})}{\pi_{\theta_{\text{old}}}(y_{i,t} \mid q, y_{i,<t})} \right). \tag{5}$$

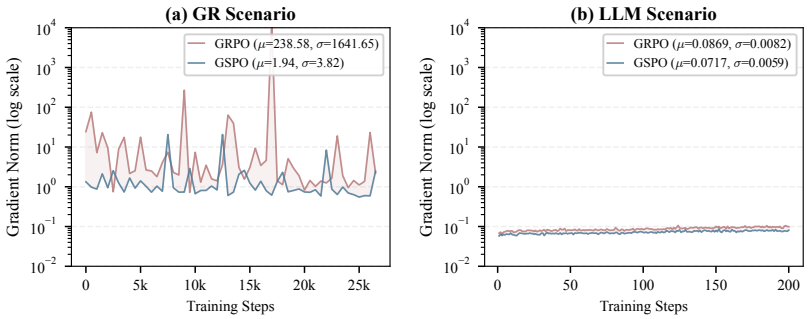

Figure 2: Gradient norms during training. (a) SID-based generative recommendation (industrial dataset, REG4Rec): the token-level ratio of GRPO gives a much larger and far more oscillatory gradient norm than the sequence-level ratio of GSPO. (b) LLM RL training (Qwen2.5-7B with DAPO-Math-17k under VERL): the two ratios give comparable magnitudes, so the instability is specific to GR. Curves in (a) are logged every 500 global steps (54 points in total) and curves in (b) every step (200 points in total).

Compared to GRPO, GSPO shifts the token-level ratio to the sequence level. This modification mitigates the high variance and potential instability introduced by per-token ratio weights [15, 16].

3 Method

In this section, we first identify three requirements for RL in SID-based generative recommendation: sequence-level stability, token-level granularity, and asymmetric updates for positive and negative trajectories. We then present SGPO, illustrated in Figure 1, as a unified solution to these requirements.

3.1 Design Motivation: Stability, Granularity, and Asymmetry

The Dilemma of Stability and Granularity. We find that RL in GR exhibits distinctive optimization characteristics. Specifically, conventional GR tends to produce substantially larger gradient norms and more pronounced training oscillations than standard LLM training. Figure 2 provides empirical evidence by visualizing the gradient norms during training with GRPO and GSPO, respectively, on an industrial-scale recommendation dataset using the conventional REG4Rec [2] backbone (Appendix C.2 for details). For comparison, we also report the corresponding statistics for LLM RL training using Qwen2.5-7B [30] with DAPO-Math-17k [19] under the default VERL training setting [31]. We observe that both the gradient norms and their variance are significantly higher in GR than in the LLM training scenario. More critically, the coefficient of variation (the ratio of the standard deviation to the mean) in GR exceeds 1.9, indicating substantial scale-normalized fluctuation, whereas that of LLM training is below 0.1.

Such severe training instability motivates adopting GSPO in GR, where sequence-level probability ratios are used to avoid the instability amplified by token-level ratios. In GRPO, token-level ratios can easily become excessively large, especially for tokens with low reference probability [15, 19, 20]. Although reducing the clipping threshold ϵ may mitigate such fluctuations, it also reduces the number of effective training samples, thereby compromising training effectiveness. In contrast, by smoothing the token-level ratios at the sequence level, GSPO yields considerably more stable results. Figure 2 further demonstrates the necessity of this treatment, showing that gradient oscillations can be effectively mitigated by GSPO.

However, GSPO also exhibits limitations in the context of GR: by operating at a coarse-grained sequence level, it blurs fine-grained token-level differences. To illustrate this point, we compare the gradient formulations of GSPO and GRPO:

$$\nabla \mathcal{J}_{\text{GSPO}}(\theta) \propto \hat{A}_i s_i(\theta) \underbrace{\sum_t}_{\text{Weight}=1} \nabla \log \pi_\theta \quad \text{vs.} \quad \nabla \mathcal{J}_{\text{GRPO}}(\theta) \propto \hat{A}_i \underbrace{\sum_t r_{i,t}(\theta)}_{\text{Weight}=r_{i,t}(\theta)} \nabla \log \pi_\theta. \tag{6}$$

Here, GRPO introduces token-aware weights, whereas GSPO aggregates these weights and treats all tokens within the same sequence uniformly. However, this uniform treatment is suboptimal, as it ignores the heterogeneous semantic roles of tokens in GR. Notably, in GR, tokens are organized in a tree structure for retrieval. Consequently, they are semantically diverse, representing everything from coarse-grained branch decisions to fine-grained leaf refinements. Intra-sequence tokens often exhibit diverse probability distributions and optimization landscapes that must be accounted for. We explicitly highlight this necessity in the subsequent analysis.

Summary 1: GR exhibits significantly higher instability during RL training compared to vanilla LLM RL training. Consequently, while employing GSPO is crucial for maintaining stability, its limitation of ignoring token-level heterogeneity need be addressed.

Asymmetric Dynamics between High- and Low-Reward Trajectories. GR can be viewed as a discrete path-search problem over the structured SID codebook. Unlike open-ended text generation, GR operates on short SID sequences (typically 4–8 tokens) within a highly constrained output space and is usually decoded with beam search [1–4]. In this setting, even small adjustments to local token probabilities can determine whether a candidate prefix survives pruning in the beam search tree. Consequently, while alignment of LLMs often prioritizes maintaining distributional diversity to support open-ended text generation, GR instead requires the model to reliably concentrate probability mass on the correct target path so that it stays within the beam under constrained decoding.

To better understand these update dynamics, we conducted a statistical analysis of trajectories from a large-scale industrial dataset (Appendix C.1). We categorize trajectories by their reward R_i and normalized advantage $\hat{A}_i$, defining high-reward (low-reward) trajectories as the top (bottom) 1% of samples. For each group, we measure the average log-probability gap between correct and incorrect tokens, revealing distinct patterns in how the policy assigns probability mass to the target path.

For positive trajectories ($\hat{A}_i > 0$), we observe a clear log-probability gap between correct and incorrect tokens, which becomes more pronounced in the high-reward group (Figure 3 (a,b)). This indicates that the model is operating in a stable, high-confidence regime on these trajectories. Recent work has consistently shown that higher model confidence correlates with response correctness [32], and that confident (high-probability) tokens are critical for model performance while low-probability tokens can over-dominate gradient updates and destabilize training [33, 34]. In GR, maintaining and even further boosting the high probabilities of positive trajectories is particularly crucial, as it increases their retrieval success rate during beam search. In other words, the policy should prioritize exploitation over exploration to avoid perturbing accurate prefixes. However, the following proposition shows that these high-probability tokens may not receive sufficient positive gradients:

Proposition 1 (Gradient Decay for High-Confidence Tokens)**.** *Under the softmax policy, the gradient norm with respect to the logits decreases as the token probability increases:*

$$\|\nabla_{z_t} \log \pi_\theta(y_{i,t} \mid q, y_{i,<t})\|^2 = 1 - 2\pi_\theta(y_{i,t} \mid q, y_{i,<t}) + \sum_{v \in V} \pi_\theta(v \mid q, y_{i,<t})^2. \tag{7}$$

In particular, when $\pi_\theta(y_{i,t} \mid q, y_{i,<t}) \to 1$, the gradient norm tends to zero.

Refer to Appendix B.1 for the detailed proof. This proposition indicates that high-probability tokens easily suffer from limited positive gradients due to the nature of conventional softmax-based policies. Consequently, these high probabilities are easily degraded by the perturbations from gradient signals from other trajectories. Motivated by these analyses, we advocate for a gradient compensation mechanism that upweights the high-confidence tokens in positive trajectories, ensuring they are more robustly preserved in the final candidate set.

For trajectories with negative advantage ($\hat{A}_i < 0$), the log-probability gap is small or even reversed (Figure 3 (c,d)), indicating that the model is overconfident in an erroneous path. Under beam search, such high-confidence wrong prefixes consume search budget and amplify deviations from the target. We therefore upweight penalties on these tokens to encourage exploration and push the policy away.

Summary 2: In GR, intra-sequence token differences must be considered: for positive trajectories, high-probability tokens should be upweighted to compensate for gradient degradation; conversely, for negative trajectories, high-probability tokens should be heavily penalized to encourage exploration.

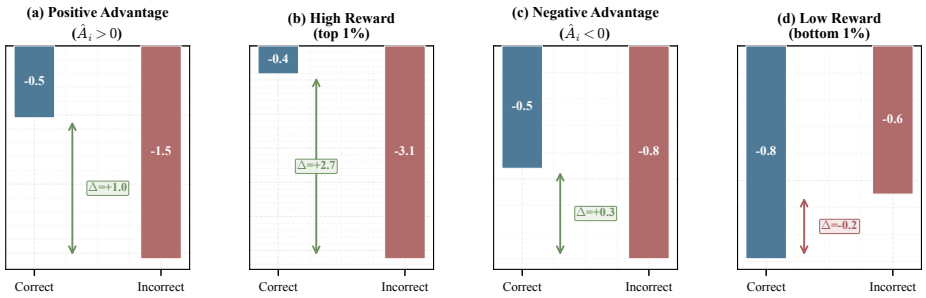

Figure 3: Average log-probabilities of correct and incorrect tokens on the industrial dataset. (a) positive advantage and (b) high reward: the two are clearly separated, i.e., high-probability tokens are mostly correct. (c) negative advantage and (d) low reward: the gap shrinks or even reverses, i.e., the model is overconfident on wrong paths. This asymmetry motivates the sign-aware reshaping of SGPO.

3.2 Stable and Granular Policy Optimization (SGPO)

Token-level Advantage Derivation. The aforementioned analyses underscore the necessity of integrating sequence-level ratios with token-level weights in GR. To achieve this, we propose SGPO, which redistributes the global sequence advantage $\hat{A}_i$ into fine-grained, token-level advantage signals $A_{i,t}$. Specifically, we construct $A_{i,t}$ by introducing a token-level differentiation term $u_{i,t}$ subject to the constraint $\sum_{t=1}^{|y_i|} u_{i,t} = 0$. This design ensures that the overall sequence-level advantage remains invariant, while adaptively adjusting the intra-sequence token importance.

Let $\ell_{i,t} = \log \pi_\theta(y_{i,t} \mid q, y_{i,<t})$ denote the log-probability of token $y_{i,t}$. To align with the insights discussed in Section 3.1, we explicitly incorporate the token's predicted probability as follows:

$$u_{i,t} = \ell_{i,t} - \frac{1}{|y_i|} \sum_{j=1}^{|y_i|} \ell_{i,j}, \quad A_{i,t} = \hat{A}_i + \mathrm{sgn}(\hat{A}_i) \cdot u_{i,t}. \tag{8}$$

See Appendix B.2 for details. For positive trajectories ($\mathrm{sgn}(\hat{A}_i) = 1$), we assign amplified positive advantages to high-confidence tokens. This compensates for gradient degradation and preserves their success rate during beam search. Conversely, for negative trajectories ($\mathrm{sgn}(\hat{A}_i) = -1$), we impose heavier negative advantages on high-confidence tokens, thereby encouraging exploration and steering the policy away from incorrect subtrees. Note that $A_{i,t}$ is treated as a stop-gradient term.

Stability Control and Adaptive Scaling. During implementation, we identified two critical challenges: (1) the magnitude of $u_{i,t}$ fluctuates across training batches and may mismatch the scale of the original advantage, leading to numerical instability; and (2) the introduction of token-aware terms can induce significantly high variance of advantages.

To tackle these issues, we explicitly introduce a sequence-wise coefficient to dynamically rescale the magnitudes and control the variance. Formally, this is supported by the following proposition:

Proposition 2 (Variance-Constrained Scaling)**.** *Given $A_{i,t} = \hat{A}_i + \mathrm{sgn}(\hat{A}_i)\lambda_i u_{i,t}$ with $\sum_t u_{i,t} = 0$, if the variance of the new advantage is bounded with $\frac{1}{|y_i|}\sum_{t=1}^{|y_i|} A_{i,t}^2 \le c\hat{A}_i^2$ for $c \ge 1$, then the maximum feasible scaling factor is*

$$\lambda_i = |\hat{A}_i| \sqrt{\frac{c-1}{\frac{1}{|y_i|}\sum_t u_{i,t}^2 + \delta}}, \tag{9}$$

where $\delta > 0$ is a small constant for numerical stability.

See Appendix B.4 for details. This yields a closed-form scaling factor that adapts to signal dispersion and advantage magnitude, and adaptively control the variance of new advantage does not exceed c times the original variance. In practice, to mitigate late-stage collapse [35], we anneal $c^{(n)}$ from c_{start}

to c_{end} over training steps $n = 1, \ldots, N$:

$$c^{(n)} = c_{\text{end}} + (c_{\text{start}} - c_{\text{end}}) \left(1 - \frac{n}{N}\right). \tag{10}$$

3.3 Discussions

We now discuss the relations of the proposed SGPO with recent RL algorithms:

Relations with GSPO. While GSPO achieves stability through a sequence-level ratio, it treats all tokens uniformly and cannot capture the heterogeneous roles of tokens along the SID tree. SGPO preserves this stability mechanism while introducing token-wise advantage reshaping, enabling fine-grained credit assignment at negligible computational cost.

Relations with GRPO. GRPO introduces token-wise weighting via token-level importance ratios and has already been adopted in SID-based GR [2, 22–24]. These works mainly perform reward engineering and trajectory grouping on top of GRPO without changing its optimization mechanism. Our analysis shows that, in GR, token-level ratios can induce high-variance, unstable updates, especially for overconfident trajectories. SGPO takes a different approach: it avoids token-level ratios entirely and instead introduces token-level differences through variance-constrained advantage reshaping, with the sequence-level ratio serving as an outer constraint.

Relations with Entropy-Based Methods. Entropy-based methods encourage distribution-level diversity, but this is misaligned with GR's goal of matching target items. SGPO uses token log-probabilities as the reshaping signal, which directly optimizes target matching, aligns with the accumulated log-probability ranking used in beam search, and distinguishes between tokens sharing identical prefixes. See Appendix B.3 for a detailed comparison.

4 Experiments

4.1 Experimental Setup

Datasets and metrics. We evaluate on three datasets: two public benchmarks and one industrial dataset. The public benchmarks are (i) Amazon Beauty [36], derived from the Beauty category of the Amazon Product Review corpus, and (ii) Tmall [37], both widely used in recommendation research. For large-scale evaluation, we use an in-house industrial dataset from a major e-commerce platform, containing over **5B** interactions among **25M** users and **48M** items and reflecting realistic large-scale production behavior. Recommendation performance is measured by HitRate@5/10 (H@5/10) and NDCG@5/10 (N@5/10) [1, 38, 39]. Further details are provided in Appendix C.1.

Backbones, baselines, and implementation. We compare SGPO with six RL baselines in two categories: (1) **representative group-based policy optimization methods** focusing on sequence-level stability and update precision (**GRPO** [14], **GSPO** [15], **SAPO** [16]); and (2) **entropy-driven exploration–exploitation methods** that use entropy signals to modulate token importance during policy updates (**GTPO** [17], **GRPO-S** [17], **AEPO** [18]). Beyond these baselines, recent RL work for GR roughly follows two additional lines. The first directly applies LLM RL techniques [3, 21, 40], largely reusing algorithms designed for general text generation with limited GR-specific design. We therefore move their comparisons to Appendix D.5 and focus the main paper on the six representative baselines above. The other line focuses on reward engineering or trajectory grouping [2, 22–24] rather than redesigning the underlying optimization mechanism, and is compatible with SGPO, which directly targets policy optimization. Among them, we adopt **REG4Rec** [2] as a representative backbone. REG4Rec employs SIDs of length 8 with fine-grained semantic decomposition and includes an RL fine-tuning stage based on GRPO [14] (*w/ GRPO*). Besides REG4Rec, we also evaluate SGPO on **TIGER** [1], a widely adopted SID-based GR model [41, 42]. TIGER uses SIDs of length 4 and is trained purely via supervised next-token prediction, without any RL stage (*w/o RL*). Further details on the backbones, baselines, and implementation are provided in Appendices C.2, C.3, and C.4.

4.2 Overall Performance

SGPO consistently outperforms all baselines across both datasets and backbones (Table 1). Our main findings are summarized as follows:

Table 1: Overall performance comparison on one offline industry dataset and two public datasets with two backbones. The best results are marked in **bold**, while the second-best results are underlined. * indicates statistically significant improvements over the second-best (p-value < 0.05).

Backbone	Method	Industrial Dataset				Tmall Dataset				Amazon Beauty			
		H@5	N@5	H@10	N@10	H@5	N@5	H@10	N@10	H@5	N@5	H@10	N@10
TIGER	*Base (w/o RL)*	.1509	.1013	.1867	.1065	.0861	.0660	.1032	.0715	.0459	.0324	.0656	.0387
	GRPO	.1890	.1257	.2275	.1320	.0968	.0741	.1158	.0802	.0493	.0348	.0705	.0416
	GSPO	.1971	.1313	.2369	.1362	.1021	.0784	.1228	.0849	.0512	.0361	.0731	.0433
	SAPO	.1946	.1284	.2360	.1333	.1012	.0765	.1218	.0840	.0509	.0356	.0728	.0429
	GRPO-S	.1941	.1285	.2312	.1336	.0996	.0766	.1196	.0825	.0503	.0355	.0718	.0422
	GTPO	.1936	.1274	.2316	.1332	.0992	.0753	.1191	.0820	.0501	.0350	.0716	.0420
	AEPO	.1914	.1280	.2308	.1324	.0984	.0757	.1178	.0815	.0498	.0352	.0712	.0418
	SGPO	**.2015***	**.1360***	**.2432***	**.1415***	**.1061***	**.0816***	**.1276***	**.0881***	**.0526***	**.0371***	**.0751***	**.0443***
REG4Rec	*Base (w/ GRPO)*	.1960	.1254	.2345	.1344	.0989	.0760	.1180	.0821	.0593	.0420	.0803	.0498
	GSPO	.2041	.1386	.2458	.1428	.1054	.0805	.1268	.0868	.0623	.0442	.0843	.0524
	SAPO	.2030	.1363	.2420	.1421	.1046	.0813	.1258	.0876	.0618	.0438	.0838	.0519
	GRPO-S	.2033	.1368	.2421	.1407	.1031	.0793	.1233	.0852	.0611	.0434	.0827	.0513
	GTPO	.2030	.1362	.2417	.1399	.1019	.0775	.1223	.0844	.0607	.0426	.0822	.0508
	AEPO	.1993	.1287	.2391	.1360	.1008	.0776	.1207	.0838	.0602	.0428	.0815	.0506
	SGPO	**.2126***	**.1446***	**.2540***	**.1473***	**.1103***	**.0850***	**.1325***	**.0916***	**.0642***	**.0456***	**.0869***	**.0540***

Stability across architectures. SGPO achieves consistent improvements in both HitRate and NDCG on both TIGER and REG4Rec, indicating robustness under different GR architectures.

Effectiveness in deep decision chains. The improvements are most pronounced on REG4Rec, where longer SID paths lead to deeper, error-prone decision chains. SGPO successfully curbs this propagation by balancing sequence-level stability with dynamic, position-aware credit assignment.

Superiority in credit assignment. Compared with group-based RL methods (GRPO, GSPO, SAPO), SGPO is better suited to the tree-structured SID space. Compared with token-level credit assignment baselines (GTPO, GRPO-S, AEPO), it benefits from asymmetric advantage reshaping rather than indiscriminate high-entropy exploration, yielding more targeted exploration and lower variance.

4.3 Evaluation of SGPO

To validate the effectiveness of SGPO, we further analyze its training dynamics, focusing on stability and the exploitation–exploration trade-off.

4.3.1 Stability Curves

To assess stability, we track entropy and gradient-norm trajectories during training (Figure 4).

Entropy stability. As shown in Figure 4 (a,c), sequence-level reweighting methods (GSPO and SGPO) produce significantly smoother entropy trajectories compared to token-ratio-based baselines (GRPO, GRPO-S, GTPO, AEPO, SAPO). The latter exhibit frequent oscillations whereas GSPO and SGPO maintain consistent entropy, confirming the stabilizing effect of sequence-level constraints.

Gradient magnitude. Figures 4 (b,d) demonstrate that SGPO and GSPO maintain gradient magnitudes orders of magnitude lower than token-level baselines. Notably, SGPO exhibits the smallest gradient extremes among all methods. This resilience stems from our adaptive scaling mechanism, which prevents the high-variance updates typical of token-level methods by anchoring credit assignment to a stable, centered signal.

4.3.2 Exploration and Exploitation

To isolate the effect of stability, we modify GTPO, GRPO-S, and AEPO to use a unified sequence-level ratio while retaining their original entropy-based exploration mechanisms, so that performance differences mainly reflect the token-level credit assignment strategy. As shown in Table 2, SGPO remains superior even under this unified sequence-level optimization, indicating that simple entropy-based exploration is insufficient for GR. Unlike open-ended generation, GR is a target-approximation problem: exploration must act as local correction toward the target, while exploitation should reinforce trajectories already aligned with the target item.

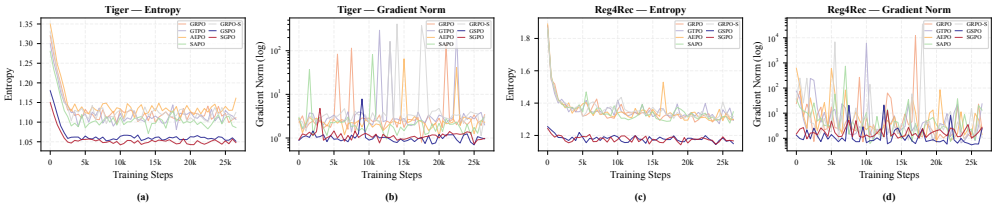

Figure 4: Training dynamics on the industrial dataset: (a,c) entropy and (b,d) gradient norm, with (a,b) for TIGER and (c,d) for REG4Rec. GSPO and SGPO produce smoother entropy trajectories and orders-of-magnitude lower gradient norms than the token-level baselines (GRPO, SAPO, GTPO, GRPO-S, AEPO), whose curves oscillate throughout training. All curves are logged every 500 global steps (54 points in total).

Table 2: Exploration and exploitation comparison on one offline industry dataset and two public datasets. All methods use a unified sequence-level ratio; only the token-level strategy differs.

Backbone	Method	Industrial Dataset				Tmall Dataset				Amazon Beauty			
		H@5	N@5	H@10	N@10	H@5	N@5	H@10	N@10	H@5	N@5	H@10	N@10
TIGER	Sequence Ratio+GTPO	.1997	.1336	.2405	.1385	.1051	.0801	.1261	.0863	.0520	.0364	.0742	.0434
	Sequence Ratio+GRPO-S	.1993	.1329	.2403	.1384	.1049	.0796	.1259	.0861	.0519	.0362	.0741	.0432
	Sequence Ratio+AEPO	.1984	.1329	.2384	.1378	.1042	.0794	.1253	.0858	.0516	.0363	.0737	.0430
	SGPO	**.2015**	**.1360**	**.2432**	**.1415**	**.1061**	**.0816**	**.1276**	**.0881**	**.0526**	**.0371**	**.0751**	**.0443**
REG4Rec	Sequence Ratio+GTPO	.2093	.1415	.2498	.1444	.1082	.0821	.1299	.0897	.0629	.0441	.0852	.0529
	Sequence Ratio+GRPO-S	.2084	.1396	.2492	.1438	.1077	.0814	.1302	.0889	.0626	.0437	.0854	.0525
	Sequence Ratio+AEPO	.2062	.1373	.2475	.1411	.1071	.0803	.1295	.0883	.0622	.0431	.0849	.0521
	SGPO	**.2126**	**.1446**	**.2540**	**.1473**	**.1103**	**.0850**	**.1325**	**.0916**	**.0642**	**.0456**	**.0869**	**.0540**

4.3.3 Ablation Study

We perform ablations on the key components of SGPO: (a) **w/o exploitation**: for positive-advantage trajectories ($\hat{A}_i > 0$), the log-probability-based reshaping $u_{i,t}$ is removed and $A_{i,t}$ reverts to the uniform sequence-level advantage $\hat{A}_i$; (b) **w/o exploration**: similarly, for negative-advantage trajectories ($\hat{A}_i < 0$), the reshaping is disabled; (c) **w/o stability control**: the adaptive scaling factor λ_i is fixed to 1 instead of being derived from the variance constraint; (d) **w/o annealing**: the reshaping bound $c^{(n)}$ is held constant at c_{start} throughout training instead of being annealed to c_{end}. (e) **w/o sequence ratio**: the sequence-level ratio $s_i(\theta)$ is replaced by the token-level importance ratios $r_{i,t}(\theta)$ as in GRPO.

Results in Table 3 show that all ablated variants underperform the full model on both industrial and public datasets, indicating that each component contributes to the overall improvement.

Table 3: Ablation study on key components of SGPO on industrial and public datasets.

Method	Industrial Dataset								Tmall Dataset							
	TIGER				REG4Rec				TIGER				REG4Rec			
	H@5	N@5	H@10	N@10	H@5	N@5	H@10	N@10	H@5	N@5	H@10	N@10	H@5	N@5	H@10	N@10
w/o exploitation	.1997	.1338	.2399	.1401	.2074	.1396	.2488	.1448	.1048	.0800	.1259	.0864	.1087	.0832	.1306	.0898
w/o exploration	.1984	.1329	.2384	.1407	.2079	.1412	.2492	.1448	.1034	.0792	.1244	.0852	.1073	.0824	.1289	.0885
w/o stability control	.1971	.1281	.2370	.1351	.2043	.1308	.2459	.1377	.1029	.0786	.1236	.0836	.1068	.0819	.1281	.0871
w/o annealing	.1984	.1210	.2370	.1280	.2068	.1346	.2475	.1388	.1038	.0780	.1253	.0842	.1079	.0812	.1299	.0876
w/o sequence ratio	.1970	.1303	.2357	.1356	.2042	.1392	.2460	.1426	.1015	.0779	.1216	.0837	.1044	.0801	.1251	.0859
SGPO	**.2015**	**.1360**	**.2432**	**.1415**	**.2126**	**.1446**	**.2540**	**.1473**	**.1061**	**.0816**	**.1276**	**.0881**	**.1103**	**.0850**	**.1325**	**.0916**

4.3.4 Online Experiments

To evaluate SGPO in a production setting, we deployed it in an online A/B test on a major e-commerce platform, where the baseline was the GRPO-based REG4Rec model and the treatment replaced GRPO with SGPO. The experiment used 10% of user traffic split evenly between control and treatment, and evaluated click-through rate (CTR) and monetization revenue as primary metrics. Compared with the baseline, SGPO achieved a 2.36% increase in CTR and a 1.95% lift in revenue, all statistically significant under a two-sided test ($p < 0.05$). These online gains demonstrate that SGPO brings practical business benefits in a large-scale industrial environment.

4.3.5 Further Analysis

We conduct five additional analyses in Appendix D: (i) **Stability curves on public benchmarks.** On the Tmall dataset, we further plot the entropy and gradient-norm trajectories for both TIGER and REG4Rec backbones, and find that the sequence-level-ratio methods GSPO and SGPO remain the most stable among all RL methods (Appendix D.1). (ii) **Signal basis comparison.** We compare log-probability, entropy, and raw probability, and find that log-probability is the most suitable coordinate for optimizing SID sequences (Appendix D.2). (iii) **Hyperparameter sensitivity.** We examine the sensitivity of the reshaping coefficient c and observe that SGPO remains robust over a broad range of values (Appendix D.3). (iv) **Model scale.** We evaluate SGPO under different model sizes and show that it consistently improves performance across parameter scales(Appendix D.4). (v) **Comparison with additional RL baselines.** We further compare SGPO against additional RL methods (SAGE [21], GBPO [40], and ECPO [3]) and demonstrate that SGPO consistently outperforms these approaches (Appendix D.5).

5 Conclusion

In this work, we studied group-based reinforcement learning (RL) for Semantic-ID (SID)–based generative recommendation and identified a core tension between stability and granularity: token-level methods such as GRPO suffer from severe training instability, while sequence-level methods such as GSPO impose uniform token weighting and ignore heterogeneous token roles in SID trees. To address this issue, we proposed Stable and Granular Policy Optimization (SGPO), a group-based RL algorithm that maintains a sequence-level ratio for stable optimization while injecting fine-grained, probability–aware and sign-aware token-wise advantages. This design enables heterogeneous yet stable updates for positive and negative trajectories under top-K decoding. Experiments on two public GR benchmarks and a billion-scale industrial dataset show that SGPO consistently outperforms six strong RL baselines and yields smoother training dynamics. Moreover, when deployed on a commercial platform, SGPO achieves a 2.36% lift in click-through rate (CTR) and a 1.95% increase in advertising revenue in online A/B testing, suggesting that SGPO is a practical step toward more stable and structure-aware group-based RL in generative recommendation.

References

[1] Rajput, S., N. Mehta, A. Singh, et al. Recommender systems with generative retrieval. *Advances in Neural Information Processing Systems*, 36:10299–10315, 2023.

[2] Xing, H., H. Deng, Y. Mao, et al. Reg4rec: Reasoning-enhanced generative model for large-scale recommendation systems. *arXiv preprint arXiv:2508.15308*, 2025.

[3] Zhou, G., J. Deng, J. Zhang, et al. Onerec technical report. *arXiv preprint arXiv:2506.13695*, 2025.

[4] Yang, Y., Z. Ji, Z. Li, et al. Sparse meets dense: Unified generative recommendations with cascaded sparse-dense representations. *arXiv preprint arXiv:2503.02453*, 2025.

[5] Yan, B., S. Liu, Z. Zeng, et al. Unlocking scaling law in industrial recommendation systems with a three-step paradigm based large user model. In *Proceedings of the Nineteenth ACM International Conference on Web Search and Data Mining*, pages 798–807. 2026.

[6] Han, R., B. Yin, S. Chen, et al. Mtgr: Industrial-scale generative recommendation framework in meituan. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, pages 5731–5738. 2025.

[7] Guo, X., B. Chen, S. Wang, et al. Onesug: The unified end-to-end generative framework for e-commerce query suggestion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 40, pages 14774–14782. 2026.

[8] Xu, Y., M. Zhang, C. Li, et al. Mmq: Multimodal mixture-of-quantization tokenization for semantic id generation and user behavioral adaptation. *arXiv preprint arXiv:2508.15281*, 2025.

[9] Hou, Y., J. Li, A. Shin, et al. Generating long semantic ids in parallel for recommendation. *arXiv preprint arXiv:2506.05781*, 2025.

[10] Hou, Y., J. Ni, Z. He, et al. Actionpiece: Contextually tokenizing action sequences for generative recommendation. *arXiv preprint arXiv:2502.13581*, 2025.

[11] Wang, W., H. Bao, X. Lin, et al. Learnable item tokenization for generative recommendation. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pages 2400–2409. 2024.

[12] Lin, J., T. Wang, K. Qian. Rec-r1: Bridging generative large language models and user-centric recommendation systems via reinforcement learning. *arXiv preprint arXiv:2503.24289*, 2025.

[13] Guo, D., D. Yang, H. Zhang, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

[14] Shao, Z., P. Wang, Q. Zhu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

[15] Zheng, C., S. Liu, M. Li, et al. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*, 2025.

[16] Gao, C., C. Zheng, X.-H. Chen, et al. Soft adaptive policy optimization. *arXiv preprint arXiv:2511.20347*, 2025.

[17] Tan, H., Z. Wang, J. Pan, et al. Gtpo and grpo-s: Token and sequence-level reward shaping with policy entropy, 2026.

[18] Dong, G., L. Bao, Z. Wang, et al. Agentic entropy-balanced policy optimization. *arXiv preprint arXiv:2510.14545*, 2025.

[19] Yu, Q., Z. Zhang, R. Zhu, et al. Dapo: An open-source llm reinforcement learning system at scale. *ArXiv*, abs/2503.14476, 2025.

[20] Zhao, Y., Y. Liu, J. Liu, et al. Geometric-mean policy optimization. *arXiv preprint arXiv:2507.20673*, 2025.

[21] Xie, Y., X. Ren, Y. Qi, et al. Sage: Sequence-level adaptive gradient evolution for generative recommendation. *ArXiv*, abs/2601.21452, 2026.

[22] Zhu, Y., H. Steck, D. Liang, et al. Rank-grpo: Training llm-based conversational recommender systems with reinforcement learning. *arXiv preprint arXiv:2510.20150*, 2025.

[23] Jiang, J., H. Tang, W. Wu, et al. S-grec: Personalized semantic-aware generative recommendation with asymmetric advantage. *ArXiv*, abs/2602.10606, 2026.

[24] Jiang, J., Y. Huang, Z. Wang, et al. Spend search where it pays: Value-guided structured sampling and optimization for generative recommendation, 2026.

[25] Zhong, Q., J. Su, Y. Ma, et al. Pctx: Tokenizing personalized context for generative recommendation. *arXiv preprint arXiv:2510.21276*, 2025.

[26] Wei, T., X. Ning, X. Chen, et al. Cofirec: Coarse-to-fine tokenization for generative recommendation. *arXiv preprint arXiv:2511.22707*, 2025.

[27] Mu, L., H. Deng, H. Xing, et al. Masked diffusion generative recommendation. *arXiv preprint arXiv:2601.19501*, 2026.

[28] Liu, Z., Y. Zhu, Y. Yang, et al. Diffgrm: Diffusion-based generative recommendation model. *arXiv preprint arXiv:2510.21805*, 2025.

[29] Deng, H., H. Xing, K. Matsuyama, et al. Csmf: Cascaded selective mask fine-tuning for multi-objective embedding-based retrieval. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2122–2131. 2025.

[30] Yang, Q. A., B. Yang, B. Zhang, et al. Qwen2.5 technical report. *ArXiv*, abs/2412.15115, 2024.

[31] Sheng, G., C. Zhang, Z. Ye, et al. Hybridflow: A flexible and efficient rlhf framework. *Proceedings of the Twentieth European Conference on Computer Systems*, 2024.

[32] Zhao, X., Z. Kang, A. Feng, et al. Learning to reason without external rewards. In *The Fourteenth International Conference on Learning Representations*. 2026.

[33] Yang, Z., X. Luo, Z. Wang, et al. Do not let low-probability tokens over-dominate in RL for LLMs. In *The Fourteenth International Conference on Learning Representations*. 2026.

[34] Agarwal, S., Z. Zhang, L. Yuan, et al. The unreasonable effectiveness of entropy minimization in LLM reasoning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*. 2026.

[35] He, B., Y. Zuo, Z. Liu, et al. How far can unsupervised RLVR scale LLM training? In *The Fourteenth International Conference on Learning Representations*. 2026.

[36] He, R., J. McAuley. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *Proceedings of the 25th International Conference on World Wide Web*, WWW '16, page 507–517. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 2016.

[37] Zhong, W., R. Jin, C. Yang, et al. Stock constrained recommendation in tmall. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 2287–2296. 2015.

[38] Xing, H., K. Matsuyama, H. Deng, et al. Esans: Effective and semantic-aware negative sampling for large-scale retrieval systems. In *Proceedings of the ACM on Web Conference 2025*, pages 462–471. 2025.

[39] Zheng, Y., J. Bian, G. Meng, et al. Multi-objective personalized product retrieval in taobao search. *arXiv preprint arXiv:2210.04170*, 2022.

[40] Zhou, G., H. Hu, H. Cheng, et al. Onerec-v2 technical report. *ArXiv*, abs/2508.20900, 2025.

[41] Li, X., B. Chen, J. She, et al. A survey of generative recommendation from a tri-decoupled perspective: Tokenization, architecture, and optimization. 2025.

[42] Hou, M., L. Wu, Y. Liao, et al. A survey on generative recommendation: Data, model, and tasks. *arXiv preprint arXiv:2510.27157*, 2025.

[43] Chen, B., X. Guo, S. Wang, et al. Onesearch: A preliminary exploration of the unified end-to-end generative framework for e-commerce search. *arXiv preprint arXiv:2509.03236*, 2025.

[44] Kang, W.-C., J. McAuley. Self-attentive sequential recommendation. In *2018 IEEE international conference on data mining (ICDM)*, pages 197–206. IEEE, 2018.

[45] Pancha, N., A. Zhai, J. Leskovec, et al. Pinnerformer: Sequence modeling for user representation at pinterest. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, pages 3702–3712. 2022.

[46] Deng, H., H. Xing, K. Matsuyama, et al. Heterrec: Heterogeneous information transformer for scalable sequential recommendation. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 3020–3024. 2025.

[47] Zhang, J., B. Zhang, W. Sun, et al. Slow thinking for sequential recommendation. *arXiv preprint arXiv:2504.09627*, 2025.

[48] Liu, S.-Y., X. Dong, X. Lu, et al. Gdpo: Group reward-decoupled normalization policy optimization for multi-reward rl optimization. *arXiv preprint arXiv:2601.05242*, 2026.

[49] Jaech, A., A. Kalai, A. Lerer, et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.

[50] Chen, Z., Y. Min, B. Zhang, et al. An empirical study on eliciting and improving r1-like reasoning models. *arXiv preprint arXiv:2503.04548*, 2025.

[51] Zheng, B., Y. Hou, H. Lu, et al. Adapting large language models by integrating collaborative semantics for recommendation. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*, pages 1435–1448. IEEE, 2024.

[52] Liu, S., Z. He, G. Zhan, et al. Stapo: Stabilizing reinforcement learning for llms by silencing rare spurious tokens. *ArXiv*, abs/2602.15620, 2026.

[53] Jegou, H., M. Douze, C. Schmid. Product quantization for nearest neighbor search. *IEEE transactions on pattern analysis and machine intelligence*, 33(1):117–128, 2010.

[54] Paszke, A., S. Gross, F. Massa, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[55] Yang, A., A. Li, B. Yang, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

[56] Su, J., Y. Lu, S. Pan, et al. Roformer: Enhanced transformer with rotary position embedding, 2021.

[57] Loshchilov, I., F. Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[58] Liu, Z., B. Chen, M. Cheng, et al. Towards context-aware reasoning-enhanced generative searching in e-commerce. In *Proceedings of the ACM Web Conference 2026*, page 6551–6561. ACM, 2026.

A Related Work

A.1 Reinforcement Learning for Generative Recommendation

SID-based GR represents each item as a sequence of discrete tokens and generates the target item autoregressively, thereby casting recommendation as a sequence generation problem [1, 2, 7, 43]. Compared with conventional sequential recommendation [38, 29, 44–46], this paradigm operates in a shared discrete semantic space and facilitates end-to-end optimization of recommendation objectives. Recent work [2, 3, 12, 21, 47] has incorporated RL into GR to directly optimize sequence-level business objectives, such as user engagement duration, clicks, and conversions. Existing RL methods for GR fall into two main categories:

The first line of work directly applies existing LLM RL techniques with limited GR-specific design. For example, SAGE [21] adopts higher clipping from DAPO [19], reward normalization from GDPO [48], and entropy encouragement, without modifying the underlying optimization scheme for GR. ECPO [3] and GBPO [40] apply gradient constraints on negative samples but do not address the structural characteristics of SID-based generation.

The second category builds on the GRPO policy-gradient framework and primarily focuses on reward engineering or trajectory grouping. REG4Rec [2] introduces multiple reward signals; Rank-GRPO [22] reweights by ranking positions; A2PO [23] uses auxiliary advantages; and Sibling-GRPO [24] groups trajectories by SID similarity. Importantly, none of these methods redesign the core policy optimization mechanism. This gap motivates our approach, which operates orthogonally to reward or grouping modifications and can be integrated as a plug-in into GRPO-based methods. In our experiments, we adopt REG4Rec [2] as a representative backbone to empirically validate this plug-and-play compatibility.

A.2 Reinforcement Learning in Large Language Models

Reinforcement learning (RL) for large language models (LLMs) is typically used in the post-training stage to improve reasoning and alignment with human preferences [13, 14, 49, 50]. GRPO [14] is widely adopted but can suffer from instability due to token-level importance weighting, whereas GSPO [15] provides smoother sequence-level updates at the cost of reduced token-level granularity. SAPO [16] further introduces a soft gating mechanism over importance ratios to balance stability and token-level adaptivity, but it remains a token-wise ratio–gating method designed for LLMs, without decomposing sequence-level credit across positions or modeling the tree-structured SID space in GR.

In parallel, token-level exploration methods such as GTPO [17], GRPO-S [17], and AEPO [18] encourage high-entropy exploration, which is effective in open-ended text generation but misaligned with recommender settings, where positive samples call for exploitation and negative samples require exploration for correction under top-K metrics [1, 4, 51].

This mismatch motivates SGPO. SGPO retains a GSPO-style sequence-level ratio for stability but explicitly decomposes them along the SID path via probability–aware token-wise advantage reshaping. This preserving sequence-level stability while enabling structure-aware token-level updates and sign-aware asymmetric exploration–exploitation behavior tailored to the SID-based GR.

B Proof of SGPO

B.1 Proof of Proposition 1: Gradient Decay for High-Confidence Tokens

Derivation of gradient decay. Under the GSPO optimization framework, the policy gradient for query q is given by:

$$\nabla \mathcal{J}_{\text{GSPO}}(\theta) \propto s_i(\theta)\hat{A}_i \sum_t \nabla_\theta \log \pi_\theta(y_{i,t}). \tag{11}$$

Let V denote the token vocabulary, $z_t \in \mathbb{R}^{|V|}$ denote the logits vector at step t, and $e_{y_{i,t}}$ be the one-hot vector of the target token. The gradient $\nabla_\theta \log \pi_\theta = (\nabla_\theta z_t)^\top (e_{y_{i,t}} - \pi_t)$ depends on the softmax residual $\|e_{y_{i,t}} - \pi_t\|$, where $\pi_t = \text{softmax}(z_t)$. In particular,

$$\|\nabla_{z_t} \log \pi_\theta(y_{i,t} \mid q, y_{i,<t})\|^2 = 1 - 2\pi_\theta(y_{i,t} \mid q, y_{i,<t}) + \sum_{v \in V} \pi_\theta(v \mid q, y_{i,<t})^2. \tag{12}$$

As $\pi_\theta(y_{i,t} \mid \cdot) \to 1$, the norm $\|\nabla_{z_t} \log \pi_\theta\|$ tends to 0, so the gradient on highly confident tokens becomes very small. In GSPO, since $\hat{A}_i$ is shared across all tokens, the model fails to differentiate the update intensity at individual positions, which can be suboptimal in the discrete SID space.

How advantage reshaping alleviates gradient decay? SGPO redefines the token-level update by scaling the gradient with a position-dependent advantage $A_{i,t}$:

$$\nabla \mathcal{J}_{\text{SGPO}}(\theta) \propto s_i(\theta) \sum_t A_{i,t} \nabla_\theta \log \pi_\theta(y_{i,t}). \tag{13}$$

Here, $A_{i,t} = \hat{A}_i + \text{sgn}(\hat{A}_i)\lambda_i u_{i,t}$. In the confidence-saturated regime, where $\pi_\theta(y_{i,t} \mid \cdot) \to 1$, the log-probability $\ell_{i,t} = \log \pi_\theta \to 0$. Given that the sequence mean $\bar{\ell} = \frac{1}{|y_i|} \sum_j \ell_{i,j}$ is significantly negative, the reshaping term $u_{i,t} = \ell_{i,t} - \bar{\ell} \approx -\bar{\ell} > 0$ becomes large. For positive-advantage samples ($\hat{A}_i > 0$), we have:

$$|A_{i,t}| = |\hat{A}_i + \lambda_i u_{i,t}| = \hat{A}_i + \lambda_i u_{i,t} = |\hat{A}_i| + \lambda_i |u_{i,t}|, \tag{14}$$

where the second equality holds because $\hat{A}_i > 0$, $\lambda_i > 0$, and $u_{i,t} > 0$ in this regime. By decoupling the advantage from the uniform $\hat{A}_i$ and introducing $u_{i,t}$, SGPO scales the gradient by $|A_{i,t}| > |\hat{A}_i|$ for these confidence-saturated tokens. This dynamic credit assignment compensates for the vanishing residual $\|e_{y_{i,t}} - \pi_t\|$, making the parameter updates remain expressive and stable even when the policy becomes highly confident.

B.2 Derivation of the Centered Projection under the Conservation Constraint

Consider the token credit space for the i-th sequence under the zero-sum constraint: $\mathcal{C}_0 = \{u_{i,t} \in \mathbb{R}^{|y_i|} : \sum_{t=1}^{|y_i|} u_{i,t} = 0\}$. We aim to find the optimal $u_{i,t}$ that is closest to the log-probability vector $\ell_{i,t}$, which is formulated as:

$$\min_{u_{i,t} \in \mathcal{C}_0} \sum_{t=1}^{|y_i|} (u_{i,t} - \ell_{i,t})^2, \quad \text{s.t.} \quad \sum_{t=1}^{|y_i|} u_{i,t} = 0. \tag{15}$$

The Lagrangian is $\mathcal{L}(u_{i,t}, \mu) = \sum_{t=1}^{|y_i|} (u_{i,t} - \ell_{i,t})^2 + \mu \left(\sum_{t=1}^{|y_i|} u_{i,t} \right)$, where μ is the Lagrange multiplier. Taking the derivative with respect to $u_{i,t}$ and setting it to zero yields

$$\frac{\partial \mathcal{L}}{\partial u_{i,t}} = 2(u_{i,t} - \ell_{i,t}) + \mu = 0 \quad \Rightarrow \quad u_{i,t} = \ell_{i,t} - \frac{\mu}{2}. \tag{16}$$

Substituting this into the constraint $\sum_t u_{i,t} = 0$ gives

$$\sum_{t=1}^{|y_i|} \left(\ell_{i,t} - \frac{\mu}{2} \right) = \sum_{t=1}^{|y_i|} \ell_{i,t} - \frac{|y_i|\mu}{2} = 0, \quad \frac{\mu}{2} = \frac{1}{|y_i|} \sum_{j=1}^{|y_i|} \ell_{i,j}. \tag{17}$$

The optimal solution is

$$u^\star_{i,t} = \ell_{i,t} - \frac{1}{|y_i|} \sum_{j=1}^{|y_i|} \ell_{i,j}. \tag{18}$$

Geometrically, $u^\star_{i,t}$ is the Euclidean orthogonal projection of $\ell_{i,t}$ onto the hyperplane $\mathcal{C}_0$. This result corresponds to the definition of $u_{i,t}$ in Section 3.2.

B.3 Proof of the Superiority of Log-Probability as a Reshaping Signal

This section justifies our choice of log-probability as the signal for credit reshaping. We provide a three-fold analysis: (1) the fundamental task divergence between GR and standard LLM alignment; (2) the gradient norm dynamics of target-token optimization; and (3) the numerical advantages of log-probability for additive advantage attribution.

Differences between GR and LLM. We choose log-probability as the core signal for advantage reshaping, rather than the entropy regularization commonly used in LLM alignment, due to the fundamental differences between GR and LLMs.

First, LLMs emphasize distributional diversity, whereas GR emphasizes point matching. The training objective of LLMs is to generate fluent and diverse text, and entropy is used to shape the distribution to avoid collapse. In contrast, the objective of GR is to achieve point matching to the target item in a discrete SID-constrained space. Excessive entropy may cause the model to drift unnecessarily in the search space and interfere with accurate target approximation. Log-probability shifts the optimization focus from shaping the distribution to improving the ranking advantage of the target path, which is more consistent with the hit-first nature of recommendation.

Second, during inference, generative recommendation typically uses Top-k beam search for candidate generation, and the final output is determined entirely by the accumulated log-probability ranking of candidate trajectories. Entropy acts on the distribution as a whole and does not directly improve the ranking of a single path. By contrast, log-probability-based advantage reshaping directly optimizes the log-probability of SID trajectories, ensuring that the training objective is naturally aligned with the filtering logic of beam search.

Third, entropy is a distribution-level statistic that captures uncertainty at the current position. However, in generative recommendation, different paths may share long prefixes; when the prefix is identical, the entropy value is the same regardless of the current token value, so entropy cannot distinguish the sampled token from other candidates at the token level. Log-probability, in contrast, is a token-level signal and can distinguish such token-level differences.

Gradient norm bounds. From Appendix B.1, we have

$$\|\nabla_{z_t} \log \pi_\theta(y_{i,t} \mid q, y_{i,<t})\|^2 = 1 - 2\pi_\theta(y_{i,t} \mid q, y_{i,<t}) + \sum_{v \in V} \pi_\theta(v \mid q, y_{i,<t})^2. \tag{19}$$

Let V denote the token vocabulary, $\mathcal{C}_t = \sum_{v \in V} \pi_\theta(v \mid q, y_{i,<t})^2$, and define the Shannon entropy as $H_t = -\sum_{v \in V} \pi_\theta(v \mid q, y_{i,<t}) \log \pi_\theta(v \mid q, y_{i,<t})$, and define the Rényi-2 entropy as

$$H_2(\pi_\theta(\cdot \mid q, y_{i,<t})) = -\log \sum_{v \in V} \pi_\theta(v \mid q, y_{i,<t})^2 = -\log \mathcal{C}_t. \tag{20}$$

Proposition 3 (Gradient Norm Bounds [52])**.** *Since Rényi entropy is no larger than Shannon entropy, $H_2(\pi_\theta(\cdot \mid q, y_{i,<t})) \le H_t$, we have $-\log \mathcal{C}_t \le H_t$, which gives the lower bound*

$$\mathcal{C}_t \ge e^{-H_t}. \tag{21}$$

Moreover, following the entropy-collision inequality in [52, 33], for a finite vocabulary V,

$$\mathcal{C}_t \le 1 - C_V H_t^2, \quad \textit{where} \quad C_V = \frac{|V| - 1}{|V|(\ln |V|)^2}. \tag{22}$$

Substituting the above bounds into the gradient expression, we obtain

$$1 - 2\pi_\theta(y_{i,t} \mid q, y_{i,<t}) + e^{-H_t} \le \|\nabla_{z_t} \log \pi_\theta(y_{i,t} \mid q, y_{i,<t})\|^2 \le 2 - 2\pi_\theta(y_{i,t} \mid q, y_{i,<t}) - C_V H_t^2. \tag{23}$$

Therefore, the gradient norm is negatively correlated with the target-token probability $\pi_\theta(y_{i,t} \mid q, y_{i,<t})$ and the local entropy H_t [52]. In particular, when a token has low probability and low entropy, its gradient norm is substantially amplified. If such a token also receives a positive advantage, it will be encouraged more strongly during optimization. However, tokens with this property are not necessarily the ones that should be reinforced: low entropy indicates that the local distribution is already sharp, which means the model has developed a strong bias at that position, while low probability suggests that the token is not a high-confidence choice. Further reinforcing such tokens may drive the model toward overfitting rare outliers and become a source of training instability.

Probability attribution and numerical dynamics. We choose log-probability rather than raw probability as the core signal for two reasons.

Probability attribution. For an autoregressive model, the response probability satisfies

$$\pi_\theta(y \mid q) = \prod_{t=1}^{|y|} \pi_\theta(y_t \mid q, y_{<t}), \quad \log \pi_\theta(y \mid q) = \sum_{t=1}^{|y|} \log \pi_\theta(y_t \mid q, y_{<t}). \tag{24}$$

Log-probability maps the multiplicative probability structure into an additive space, providing a local coordinate system that is directly consistent with the sequence-level objective. This makes the centered projection described in Appendix B.2:

$$u^\star_{i,t} = \ell_{i,t} - \frac{1}{|y_i|} \sum_{j=1}^{|y_i|} \ell_{i,j}. \tag{25}$$

more mathematically smooth and structurally justified.

Difference amplification. Compared with raw probability, whose values are strictly bounded in $[0, 1]$, log-probability stretches the numerical range in the logarithmic domain, allowing credit reshaping to capture subtle differences between sequences more sensitively.

B.4 Proof of Proposition 2: Variance-Constrained Scaling

The magnitude of $u_{i,t}$ may vary across training batches. To control gradient oscillation, we impose a second-moment constraint on the reshaped advantages. Specifically, given

$$A_{i,t} = \hat{A}_i + \operatorname{sgn}(\hat{A}_i)\lambda_i u_{i,t}, \tag{26}$$

with $\sum_t u_{i,t} = 0$, we require

$$\frac{1}{|y_i|} \sum_{t=1}^{|y_i|} A_{i,t}^2 \le c^{(n)} \hat{A}_i^2, \quad c^{(n)} \ge 1. \tag{27}$$

Expanding the left-hand side yields

$$\begin{aligned} \frac{1}{|y_i|} \sum_{t=1}^{|y_i|} A_{i,t}^2 &= \frac{1}{|y_i|} \sum_{t=1}^{|y_i|} \Big(\hat{A}_i + \operatorname{sgn}(\hat{A}_i)\lambda_i u_{i,t} \Big)^2 \\ &= \hat{A}_i^2 + 2|\hat{A}_i|\lambda_i \cdot \frac{1}{|y_i|} \sum_{t=1}^{|y_i|} u_{i,t} + \lambda_i^2 \frac{1}{|y_i|} \sum_{t=1}^{|y_i|} u_{i,t}^2 \\ &= \hat{A}_i^2 + \lambda_i^2 \frac{1}{|y_i|} \sum_{t=1}^{|y_i|} u_{i,t}^2, \end{aligned} \tag{28}$$

where the cross term vanishes due to the zero-sum constraint $\sum_t u_{i,t} = 0$. Enforcing the constraint then gives

$$\hat{A}_i^2 + \lambda_i^2 \frac{1}{|y_i|} \sum_{t=1}^{|y_i|} u_{i,t}^2 \le c^{(n)} \hat{A}_i^2, \tag{29}$$

which can be rearranged as

$$\lambda_i^2 \le \frac{(c^{(n)} - 1)\hat{A}_i^2}{\frac{1}{|y_i|} \sum_t u_{i,t}^2}. \tag{30}$$

Thus, the maximum feasible scaling factor is

$$\lambda_i = |\hat{A}_i| \sqrt{\frac{c^{(n)} - 1}{\frac{1}{|y_i|} \sum_t u_{i,t}^2}}. \tag{31}$$

Table 4: Statistics of the datasets.

Dataset	Industrial Dataset	Tmall	Amazon Beauty
#User	25,578,429	2,554,524	22,363
#Item	48,580,380	3,898,675	12,101
#Interaction	5,248,208,274	33,577,654	198,360

In practice, we add a small constant $\delta > 0$ for numerical stability:

$$\lambda_i = |\hat{A}_i| \sqrt{\frac{c^{(n)} - 1}{\frac{1}{|y_i|} \sum_t u_{i,t}^2 + \delta}}. \tag{32}$$

This yields a closed-form scaling factor that adapts to signal dispersion and advantage magnitude, reducing update intensity for highly dispersed signals and small-advantage trajectories to avoid over-correction.

C Supplementary Experimental Setup

C.1 Dataset Statistics

We evaluate SGPO on one large-scale industrial dataset and two public benchmarks, to cover both realistic production environments and standard academic settings. The detailed statistics of these datasets are summarized in Table 4.

Industrial Dataset. We construct an in-house offline dataset from sequential user behaviors and feedback logs on a large-scale e-commerce advertising platform. The dataset spans October 2025–March 2026 and covers multiple traffic scenarios, including homepage feeds, product detail pages, and interactive gaming sections. It contains over 5.2 billion interactions from 25.6 million users over 48.6 million advertisements, providing broad coverage of real-world recommendation and advertising behaviors. Compared with public benchmarks, this dataset is substantially larger and more heterogeneous, with richer feedback signals and stronger cross-scenario distribution shifts, making it a challenging testbed for RL-based generative recommendation.

Tmall. The Tmall dataset [37] is collected from the recommendation logs of Tmall, a large-scale B2C e-commerce platform in China, under an item-to-item recommendation setting. In this scenario, related products are displayed on the detail page of a target item, and the recommendation task is to retrieve items that are likely to be clicked under the current source-item context. The dataset contains over 33 million interactions from more than 2.5 million users across nearly 3.9 million items, covering multiple product categories such as apparel, electronics, beauty, and home goods. It provides large-scale user behavior signals, including exposure, click, and purchase records, together with rich product-side features such as price, inventory level, predicted traffic, and conversion-related signals. Based on historical interactions, an item-to-item click-through-rate matrix is constructed to characterize the transition likelihood between source and target items. Since the full item-pair space is extremely large and sparse, only the most relevant candidate pairs are retained. Overall, the dataset reflects a realistic large-scale e-commerce recommendation environment with both behavioral and product-related information.

Amazon Beauty. We use the Beauty category from the Amazon Product Reviews dataset [36], a widely adopted benchmark for recommendation research. Following standard preprocessing practice [1, 4], we retain users and items with sufficient interaction frequency and organize each user's interactions in chronological order. After preprocessing, the resulting dataset contains about 198 thousand interactions from over 22 thousand users on more than 12 thousand items. Compared with industrial-scale datasets, Amazon Beauty is smaller in scale but exhibits higher sparsity, making it a commonly used benchmark for evaluating recommendation models in sparse settings. Following the standard leave-one-out protocol [44], we use the last item of each sequence for testing, the second-to-last for validation, and the remaining interactions for training, except for REG4Rec, where a prediction is counted as a successful recall as long as at least five of the eight SIDs are predicted correctly.

C.2 Backbones

TIGER [1]. TIGER is a widely adopted SID-based GR model for item retrieval. It first builds a semantic codebook with RQ-VAE to quantize item content features into discrete SIDs, and then employs an autoregressive Transformer to generate SIDs token by token along a tree-structured decoding path. The model is trained purely with supervised next-token prediction, without any RL.

REG4Rec [2]. REG4Rec is a GR backbone for item retrieval based on a parallel long-SID design. It constructs a higher-capacity SID space by using multiple parallel quantization codebooks [8, 53], leading to longer and more expressive SID sequences. During inference, REG4Rec performs tree-structured decoding over these SIDs to retrieve target items. Unlike TIGER, REG4Rec includes an RL fine-tuning phase that adopts GRPO [14] as the group-based policy optimization algorithm and designs task-specific rewards aligned with recommendation metrics.

C.3 RL Baselines

We compare our method with six RL baselines, which fall into two categories: three representative group-based policy optimization methods and three entropy-driven exploration–exploitation methods. **Representative group-based policy optimization baselines. GRPO [14]** samples multiple responses per query and computes advantages through group-normalized rewards. It applies clipping to the importance ratio at the token level, which can cause high-variance gradient estimates and hence unstable training [15, 16, 20]. **GSPO [15]** improves stability by lifting the importance ratio to the sequence level; however, using a single sequence-level weight may obscure the heterogeneous contributions of individual tokens, potentially losing fine-grained credit information in complex sequences. **SAPO [16]** replaces hard clipping with a temperature-controlled soft gating function over importance ratios, thereby forming a continuous trust region for off-policy updates. Nevertheless, SAPO still primarily focuses on modulating update magnitudes and does not explicitly model credit assignment along the tree-structured decoding process used in the GR setting.

Entropy-driven exploration–exploitation baselines. GTPO [17] redistributes rewards through a dynamic entropy-weighting mechanism. It assigns higher rewards to high-entropy tokens in successful trajectories to encourage exploration, while imposing larger penalties to low-entropy tokens in unsuccessful trajectories to discourage overconfident predictions. **GRPO-S [17]** is a sequence-level variant of GTPO that performs reward shaping based on the average entropy of the entire sequence, aiming for a more stable trade-off between token-level precision and overall sequence coherence. **AEPO [18]** balances entropy-guided exploration and training stability by monitoring token entropies to allocate rollout budgets across tokens, and by applying a stop-gradient operation to preserve gradients of high-entropy tokens during policy updates.

C.4 Implementation Details

All experiments are implemented in PyTorch [54] and conducted on a distributed training platform with 2 parameter servers and 10 workers, where each worker is equipped with a single NVIDIA L20 GPU. Unless otherwise specified, all compared methods are trained under the same data pipeline, rollout procedure, and decoding configuration on the same backbone, so that performance differences can be attributed to the policy optimization algorithm rather than engineering variations. To ensure a fair comparison, semantic attribute features are incorporated into the encoder input for all methods.

For both backbones, item semantic representations are first obtained using the pretrained Qwen3-8B model [55], and are then quantized into discrete Semantic IDs through vector quantization. The tokenizer is trained separately on each dataset and fixed afterward. All GR methods on the same dataset share the same codebook and SID assignment during both training and inference. The Industrial and Tmall datasets are multi-positive.

TIGER. TIGER is a standard SID-based generative retrieval backbone that assigns each item a fixed semantic path in a shared codebook and performs autoregressive decoding along this path. Following its original design, we use a 4-layer residual quantization codebook with 512 codewords per layer, resulting in SIDs of length 4. The model adopts an encoder–decoder Transformer architecture with Rotary Position Embedding (RoPE) [56] for positional encoding. On the Industrial Dataset and Tmall Dataset, we use 2 Transformer encoder layers and 5 Transformer decoder layers, with input embedding size 1024, hidden size 1024, 64 attention heads, and feed-forward intermediate size 2560.

On Amazon Beauty, we use a smaller model for efficiency while keeping the same encoder–decoder structure and positional encoding scheme. Specifically, the input embedding size and hidden size are both set to 256, the number of attention heads is set to 8, and the feed-forward intermediate size is reduced to 640.

TIGER is first trained with supervised next-token prediction on target SID sequences and then fine-tuned with RL when applicable. In both stages, we use AdamW [57] as the optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and weight decay 10^{-4}. For the Industrial Dataset and Tmall Dataset, the learning rate is set to $7\mathrm{e}{-4}$ during supervised training and $7\mathrm{e}{-5}$ during RL fine-tuning. On Amazon Beauty, we use the same optimization strategy while adopting the smaller model configuration described above for efficiency.

REG4Rec. REG4Rec differs from TIGER mainly in its finer-grained SID design and more flexible decoding mechanism. Specifically, REG4Rec uses an 8-layer semantic codebook with 300 codewords per layer, resulting in longer SIDs and a larger structured retrieval space. As in TIGER, the codebook is constructed from item semantic embeddings derived from the pretrained Qwen3-8B model [55] and is fixed after tokenizer training. Unlike TIGER, which follows a single fixed semantic path, REG4Rec employs a dynamic reasoning path during decoding, enabling more flexible trajectory construction under long SID sequences. In addition, REG4Rec incorporates an MTP-based design to provide denser intermediate supervision and reward signals, which is particularly beneficial for optimization over deep decision chains. The model adopts an encoder–decoder Transformer architecture with RoPE [56] for positional encoding. On the Industrial Dataset and Tmall Dataset, we use 2 Transformer encoder layers and 5 Transformer decoder layers, with input embedding size 1024, hidden size 1024, 64 attention heads, and feed-forward intermediate size 2560. On Amazon Beauty, we use a smaller model for efficiency while preserving the same architecture family and positional encoding scheme. Specifically, the input embedding size and hidden size are both set to 256, the number of attention heads is set to 8, and the feed-forward intermediate size is reduced to 640.

REG4Rec is trained with the same two-stage pipeline of supervised pretraining followed by RL fine-tuning. In both stages, we use AdamW [57] with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and weight decay 10^{-4}. For the Industrial Dataset and Tmall Dataset, the learning rate is set to $7\mathrm{e}{-4}$ during supervised training and $7\mathrm{e}{-5}$ during RL fine-tuning. On Amazon Beauty, we follow the same optimization strategy while using the smaller model configuration described above.

RL training details. For all RL methods, we adopt the same group-based training protocol. For each query, we sample a group of 8 candidate SID sequences from the reference policy and compute group-normalized advantages as in Eq. 2. The clipping coefficient is set to $\epsilon = 0.2$ for all clipping-based methods. The RL loss weight is set to 5.0, and KL regularization is enabled with $\beta = 1.0$.

For SAPO, we set $\tau_{\text{pos}} = 1.0$ and $\tau_{\text{neg}} = 1.2$. For GTPO and GRPO-S, we set $\alpha_1 = 1.0$ and $\alpha_2 = 0.5$. For AEPO, we set the entropy reshaping coefficient to 0.5 on the industrial dataset and 0.3 on the public datasets. For SGPO, we set $c_{\text{start}} = 4$ on the industrial dataset and $c_{\text{start}} = 2.5$ on the public datasets, with $c_{\text{end}} = 1$ in all cases.

Unless otherwise stated, all RL baselines use the same reward signals, rollout budget, and decoding configuration on the same backbone.

Inference and decoding. During inference, all methods use beam search with beam size 100 to decode SID sequences and retrieve items through the corresponding fixed codebook. For fair comparison, the beam size is kept identical across all methods under the same backbone. This ensures that the reported differences reflect the effect of policy optimization rather than variations in search configuration.

D Further Analysis

D.1 Stability curves on public benchmarks

We further plot the entropy and gradient-norm trajectories on the Tmall dataset for both TIGER and REG4Rec backbones, as shown in Figure 5. The results show that the sequence-level-ratio methods GSPO and SGPO remain the most stable among all RL methods, with smoother entropy evolution and significantly lower gradient norms throughout training. In contrast, token-level methods exhibit more pronounced oscillations in both curves, indicating less stable optimization dynamics.

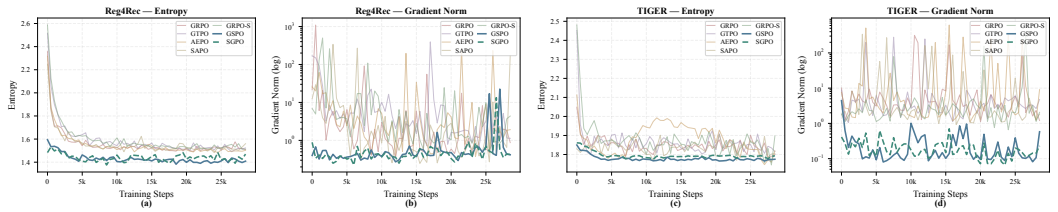

Figure 5: Entropy and gradient-norm trajectories of seven RL algorithms on the Tmall dataset under the TIGER and REG4Rec backbones. Across both backbones, GSPO and SGPO maintain the smoothest entropy curves and the lowest gradient norms, demonstrating the most stable training dynamics. All curves are logged every 500 global steps (54 points in total).

Table 5: Signal basis comparison on the industrial dataset.

Method	TIGER				REG4Rec			
	H@5	N@5	H@10	N@10	H@5	N@5	H@10	N@10
Raw probability	.1989	.1326	.2401	.1382	.2081	.1403	.2497	.1441
Entropy	.1979	.1312	.2378	.1369	.2078	.1371	.2486	.1410
Log-probability	**.2015**	**.1360**	**.2432**	**.1415**	**.2126**	**.1446**	**.2540**	**.1473**

D.2 Signal basis comparison

We conduct a comprehensive comparison of different signal bases for advantage reshaping: log-probability, entropy, and raw probability. As shown in Table 5, log-probability consistently outperforms the other two signals across all metrics on both TIGER and REG4Rec backbones. Entropy performs the worst among the three options, which aligns with our theoretical analysis that entropy operates at the distribution level and cannot distinguish between tokens sharing identical prefixes. Raw probability shows moderate performance but still falls short of log-probability. In contrast, log-probability provides a token-level signal that directly reflects the model's confidence in each generation step, making it more suitable for controlling the exploration-exploitation trade-off in generative retrieval. The superior performance also validates that log-probability better aligns with the accumulated scoring mechanism used in beam search decoding. These empirical results validate our design choice of using log-probability as the primary signal for advantage reshaping.

D.3 Hyperparameter Sensitivity

We further study the sensitivity of SGPO to the initial reshaping strength c_{start}, while fixing $c_{\text{end}} = 1$. Conceptually, a smaller c_{start} reduces the magnitude of token-level reshaping, limiting the model's ability to differentiate updates between tokens (and between positive and negative trajectories). Conversely, a larger c_{start} amplifies token-level differences but may introduce stronger local perturbations and compromise overall training stability.

Figure 6 reports the retrieval performance on the industrial dataset with the REG4Rec backbone. We observe that all four metrics remain stable for $c_{\text{start}} \in [3, 5]$, indicating that SGPO is reasonably robust to this hyperparameter. The best overall performance is obtained around $c_{\text{start}} = 4$.

D.4 Model Scale

We analyze how SGPO performs across different model scales by varying the encoder–decoder depth of the REG4Rec backbone on the industrial dataset. Specifically, we evaluate three representative settings:

- 0.16B: encoder layers = 2, decoder layers = 5, token embedding size = 1024, intermediate size = 2560.
- 0.88B: encoder layers = 2, decoder layers = 5, token embedding size = 2048, intermediate size = 10240.

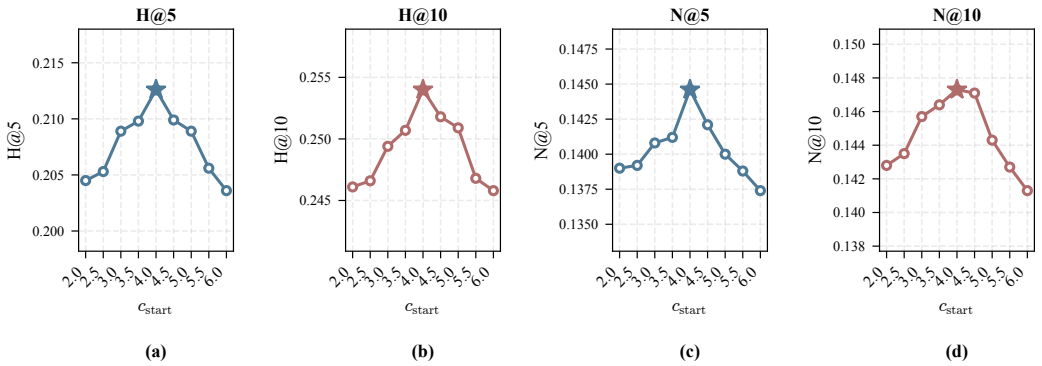

Figure 6: Sensitivity of c_{start} on the industrial dataset (REG4Rec backbone), with c_{end} fixed at 1. All four metrics remain stable across $c_{\text{start}} \in [3, 5]$, with the best performance around $c_{\text{start}} = 4$. Stars indicate the optimal value.

Table 6: Effect of model scale on the industrial dataset with the REG4Rec backbone. We compare three configurations with increasing parameter counts.

Method	0.16B				0.88B				1.7B			
	H@5	N@5	H@10	N@10	H@5	N@5	H@10	N@10	H@5	N@5	H@10	N@10
GRPO	.1960	.1254	.2345	.1344	.2012	.1328	.2424	.1382	.2126	.1446	.2646	.1535
GSPO	.2041	.1386	.2458	.1428	.2073	.1493	.2521	.1563	.2185	.1595	.2791	.1701
SAPO	.2030	.1363	.2420	.1421	.2054	.1438	.2459	.1502	.2158	.1490	.2720	.1650
GRPO-S	.2033	.1368	.2421	.1407	.2058	.1400	.2467	.1510	.2160	.1492	.2765	.1672
GTPO	.2030	.1362	.2417	.1399	.2054	.1396	.2465	.1520	.2168	.1520	.2773	.1676
AEPO	.1993	.1287	.2391	.1360	.2047	.1331	.2445	.1443	.2139	.1397	.2661	.1602
SGPO	**.2126**	**.1446**	**.2540**	**.1473**	**.2145**	**.1523**	**.2617**	**.1596**	**.2241**	**.1658**	**.2849**	**.1735**

- 1.7B: encoder layers = 3, decoder layers = 7, token embedding size = 3072, intermediate size = 16384.

As shown in Table 6, SGPO consistently improves performance across all network sizes. Notably, even at the smallest scale (0.16B), SGPO achieves substantial gains over all baselines, demonstrating that the benefit of SGPO is not contingent on large model capacity. As the model capacity increases, SGPO maintains a consistent advantage, confirming that our method is complementary to scaling rather than a substitute for it.

D.5 Comparison with Additional RL Baselines

We further compare SGPO with RL methods for GR on the industrial dataset, using both TIGER and REG4Rec backbones. These methods fall into two groups, both of which are complementary to the policy optimization mechanism.

Methods that adapt LLM post-training techniques. SAGE [21], GBPO [40], and ECPO [3] adapt techniques from LLM post-training. SAGE is a sequence-level adaptive policy optimization method that improves training stability by applying asymmetric gradient bounds and multi-objective signal decoupling for generative recommendation. GBPO is a gradient-bounded optimization method that stabilizes RL training by constraining policy updates with dynamic bounds derived from BCE-style gradients. ECPO is an early-clipped variant of GRPO that improves stability by clipping negative-advantage updates before the policy ratio grows too large. As discussed in Section A, these methods naively combine existing LLM RL techniques (higher clipping, reward normalization, gradient constraints) without redesigning the core policy optimization mechanism for generative recommendation.

Methods that modify the reward or the trajectory grouping. Sibling-GRPO [24] groups the sampled trajectories by their position in the SID tree, so that a group consists of siblings that share a prefix and differ at the position right after it, and normalizes the advantage locally within each group;

Table 7: Comparison with additional RL baselines on the industrial dataset. The last four rows report two GR-specific designs together with their combinations with SGPO.

Backbone	Method	H@10	N@10
TIGER	SAGE	.2391	.1392
	GBPO	.2287	.1329
	ECPO	.2279	.1326
	SGPO	**.2432**	**.1415**
REG4Rec	SAGE	.2468	.1436
	GBPO	.2361	.1379
	ECPO	.2356	.1358
	Sibling-GRPO	.2372	.1393
	Sibling-GRPO + SGPO	.2421	.1436
	R-GRPO	.2386	.1538
	R-GRPO + SGPO	.2417	.1594
	SGPO	**.2540**	**.1473**

this adaptation follows beam-search decoding, where candidates that share a prefix compete for the same beam slots. R-GRPO [58] identifies the ranking bias of GRPO-style updates in search scenarios and introduces a debiased variant whose reward is aggregated in a rank-aware manner over the top-K SID list returned by beam search, instead of being computed from the binary match of a single path. Both therefore adapt the sampling, the advantage normalization, or the reward to the SID tree and to beam search, and leave the optimization mechanism untouched, so they are orthogonal to SGPO and can be combined with it. We therefore apply SGPO as a plug-in on top of each of the two designs, and report the corresponding rows in Table 7.

Table 7 presents the comparison. SGPO outperforms SAGE, GBPO, and ECPO on all metrics under both backbones. Adding SGPO further improves Sibling-GRPO by 0.0049 H@10 and 0.0043 N@10, and R-GRPO by 0.0031 H@10 and 0.0056 N@10. R-GRPO alone already attains a higher N@10 than SGPO, since its rank-aware reward directly targets the ordering of the retrieved list; the combination R-GRPO + SGPO gives the best N@10 in the table. This indicates that reward-level ranking information and token-level credit assignment are complementary rather than competing.

D.6 Factorial Analysis of Reshaping, Annealing, and Adaptive Scaling

The ablation in Table 3 removes one component at a time. To further disentangle the three components of the advantage, namely the reshaping term $u_{i,t}$, the adaptive scaling factor λ_i, and the annealing schedule of the bound $c^{(n)}$, we enumerate all variants that can actually be constructed. Two of the four cells are mathematically empty: annealing schedules the bound $c^{(n)}$ inside λ_i, which is by definition the coefficient of $u_{i,t}$, so both *annealing only* and *scaling only* reduce to GSPO without reshaping and have nothing to act on. Enumerating the three components therefore yields five valid variants E1–E5. E4 is new, while E1, E2, E3, and E5 coincide with the GSPO baseline and the corresponding rows of Table 3.

Specifically, E2 applies the reshaping uniformly ($A_{i,t} = \hat{A}_i + \mathrm{sgn}(\hat{A}_i)u_{i,t}$); E3 switches on the adaptive scaling but fixes c at c_{start}; E4 anneals the reshaping through a coefficient $\kappa^{(n)}$ that follows the same schedule as $c^{(n)}$ but carries no data-adaptive variance constraint; and E5 is the full SGPO. Table 8 reports the five variants on the industrial dataset with both backbones.

Pairwise findings. E2 vs. E1: reshaping alone cannot deliver its effect, since without annealing it collapses the entropy in late training and brings no consistent gain over GSPO. E3 vs. E1: a fixed c keeps the reshaping strong late in training and still collapses the entropy, so this variant sometimes remains behind GSPO. E4 vs. E1: reshaping together with annealing surpasses GSPO on all eight combinations, so the two are already sufficient. E4 vs. E2 (adding annealing) improves all eight combinations, and E4 vs. E3 (annealing instead of a fixed bound) repairs the collapse caused by a fixed c and wins on all eight combinations, indicating that annealing matters more than the variance constraint. E5 vs. E4 (adding the variance constraint) brings a further gain on all eight combinations, since the constraint removes the update oscillation that annealing alone does not.

Table 8: Factorial analysis of reshaping, adaptive scaling, and annealing on the industrial dataset.

	TIGER				REG4Rec			
Variant	H@5	N@5	H@10	N@10	H@5	N@5	H@10	N@10
E1 GSPO	.1971	.1313	.2369	.1362	.2041	.1386	.2458	.1428
E2 reshaping only	.1971	.1281	.2370	.1351	.2043	.1308	.2459	.1377
E3 w/o annealing	.1984	.1210	.2370	.1280	.2068	.1346	.2475	.1388
E4 w/o scaling	.1992	.1331	.2388	.1381	.2087	.1409	.2496	.1442
E5 SGPO	**.2015**	**.1360**	**.2432**	**.1415**	**.2126**	**.1446**	**.2540**	**.1473**

Summary. Reshaping is the source of the gain but only takes effect together with annealing; adding the adaptive scaling on top of the two removes the residual oscillation and gives the best variant. The three components are therefore indispensable, and every one of them is singled out by a row of Table 3.

E Limitations

Sensitivity to base model capability. The effectiveness of SGPO relies on the backbone model's initial capacity to represent meaningful SID sequences. When the backbone is less mature or the initial trajectories are still noisy, the model may occasionally assign relatively high log-probability to suboptimal paths, making it appear overconfident on some incorrect branches. In such cases, likelihood-based advantage reshaping may be less effective, since the reshaping signal is influenced by the quality of the underlying trajectory distribution and may require a more reliable base model to fully realize its benefit.

Scenario-specific applicability. Our experiments are conducted on SID-GR, which focuses on relatively short reasoning trajectories. We have not yet evaluated whether SGPO transfers to text-generation tasks or other sequential settings such as query rewriting. Exploring its applicability across broader task families remains an interesting direction for future work.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The paper's scope and main contributions are clearly articulated in both the abstract and Section 1. In particular, the opening paragraph of Section 1 delineates the problem setting and scope, while the closing paragraph summarizes the key contributions.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Limitations of our work is detailed in Appendix E

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: All relatively complex mathematical formulas are accompanied by explicit derivations and intermediate transformations, with the complete proofs presented in Appendices B.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The paper provides sufficient experimental details for reproducing the main results. The implementation details are provided in Appendix C.4.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: SID training and the GR backbone have already brought us substantial commercial revenue, so our advanced SID and GR cannot be open-sourced; only the RL algorithm part, copied verbatim from the production environment, can be released.

   However, to promote reproducibility and to verify the generalization ability of SGPO, although our SID and GR cannot be open-sourced, we reproduced the effect of SGPO on the open-source **Rec-R1** backbone and **ReSID** backbone.

   Rec-R1 is not SID-based GR but is also GR and is relatively classic; ReSID is a SID-based beam-search GR. The SIDs on both backbones use the Amazon embeddings and SIDs provided by MiniOneRec.

   We repeated the experiments **3–5 times** on both backbones, covering multiple datasets, and report statistical significance, and **release the checkpoints**.

   Code is available at `https://github.com/Gabriel-reina/sgpo`.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We introduce the datasets and their preprocessing procedures in Section C.1. We present the core experimental results in Section 4, and provide additional analysis experiments and extended results in Appendix D.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: In the main experiments, we use two-sided t-tests with $p < 0.05$ as the significance threshold, and the improvements observed in the online A/B test are also statistically significant under this criterion.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: We report the details of the computing resources in the implementation section of Appendix C.4.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

Answer: [Yes]

Justification: We have ensured that our paper complies with the NeurIPS Code of Ethics.

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.

- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: The current version of the paper focuses on the technical contributions and empirical performance of SGPO, and does not explicitly discuss broader positive or negative societal impacts.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not release high-risk models or scraped datasets and therefore poses no notable risk of misuse requiring additional safeguards.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We comply with these requirements by carefully acknowledging the creators and original owners of all assets used in our work. For every asset—including datasets, code, and models—we cite the original source and clearly specify the exact version employed. We also explicitly document the corresponding licenses and terms of use, and have cross-checked them multiple times to ensure that we fully respect and adhere to each asset's usage conditions.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [N/A]

Justification: The paper does not release new assets.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: Our work focuses solely on recommendation tasks and does not touch on the high-risk applications described here. In particular, we conduct no crowdsourcing experiments and no experiments involving human subjects, so this item is not applicable to our paper.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: We are not involved in these risks as we are only engaged in recommendation tasks. Our research does not involve human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLM is used only for formatting purposes.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.