DON'T LET CONFLICTING REWARDS PULL GENERATIVE RECOMMENDATIONS OFF TARGET

Anonymous authors
Paper under double-blind review

ABSTRACT

LLM-based generative recommender systems must satisfy multiple objectives, but in practice these objectives may have different priorities, with recommendation accuracy serving as the primary objective and auxiliary objectives such as diversity and rationale coherence providing complementary training signals. In widely adopted reinforcement learning with verifiable rewards (RLVR) frameworks such as Group Relative Policy Optimization (GRPO), these heterogeneous rewards are merged into a fixed weighted sum, implicitly treating all rewards as equally tradable. When rewards conflict and their signals differ in scale, difficulty, and saturation, fixed aggregation can cancel gradients or let fast-improving auxiliary rewards dominate, resulting in reward hacking and signal collapse that can even degrade recommendation accuracy. To address this, building on the RLVR-style optimization framework, we propose Groupwise Gradient-Aligned Reward-Decoupled Policy Optimization (G^2RPO), which dynamically selects reward-aggregation weights to maximize alignment with the primary accuracy gradient while constraining the resulting update to avoid substantial first-order degradation of auxiliary objectives. This yields a primary-centered update within the region defined by auxiliary compatibility constraints. Concretely, G^2RPO computes per-reward aggregation weights via a simplex linear program, which identifies a weighted combination of the primary (i.e., accuracy) and auxiliary reward (i.e., diversity) gradients that is maximally aligned with the primary reward gradient, while enforcing that auxiliary gradients are non-adversarial to the resulting update direction. Experiments on five Amazon domains and MovieLens show generally consistent gains in accuracy, diversity, and rationale coherence. Beyond offline evaluation, our method is deployed in production and achieves the highest click-through rate in a real-world online A/B test[1].

1 INTRODUCTION

Recommender systems are inherently multi-objective: beyond recommendation accuracy, they must also satisfy auxiliary goals such as recommendation diversity and natural-language rationale coherence (Lin et al., 2025). These objectives can conflict; for instance, increasing recommendation diversity can introduce a trade-off with accuracy. Moreover, these objectives need not be equally prioritized: recommendation accuracy is typically the primary objective, motivating an optimization scheme that remains centered on accuracy while explicitly accounting for conflicts with auxiliary objectives. This hierarchy is particularly important as Reinforcement Learning with Verifiable Rewards (RLVR) has become a widely adopted training paradigm for LLM-based generative recommender systems (LLM-Rec) (Lin et al., 2025; Park et al.), making it essential to incorporate heterogeneous rewards with different priorities into stable policy updates.

In RLVR frameworks such as Group Relative Policy Optimization (GRPO) (Shao et al., 2024; Guo et al., 2025), multi-reward policy optimization is typically implemented by aggregating multiple reward signals into a single training objective via a fixed weighted sum. However, this fixed weighted-sum aggregation biases policy updates toward trajectories that maximize the aggregate reward, without explicitly modeling interactions and conflicts among individual reward components. This can induce gradient cancellation across conflicting rewards, weakening the effective learning

[1]We released our data and code to support reproducibility.

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

signal and causing some reward contributions to vanish while others dominate. In the worst case, auxiliary rewards can steer the update in a direction that conflicts with the primary reward gradient, thereby degrading the primary objective. As shown in Fig. 1, under GRPO, the gradient induced by the accuracy reward frequently exhibits negative cosine similarity with auxiliary reward gradients (e.g., diversity), suggesting that these objectives often drive the policy update in opposing directions. Consequently, such conflicting updates can cause auxiliary objectives to dominate the optimization dynamics, ultimately undermining recommendation accuracy, the most critical criterion in recommender systems.

Findings: In RLVR-based LLM recommenders, different reward signals often induce conflicting policy-gradient directions. As shown in Fig. 1, naively summing rewards can cause gradient cancellation, weakening individual reward effects and ignoring reward interactions.

To address these limitations, we propose **Groupwise Gradient-Aligned Reward-Decoupled Policy Optimization (G^{2}RPO)**, a reinforcement learning framework for stable multi-reward joint optimization in generative recommendation. Our key insight is to retain per-reward normalization while replacing fixed scalar aggregation with gradient alignment aware dynamic weighting.

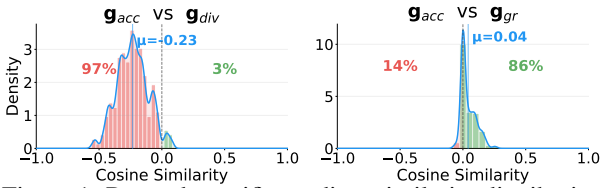

Figure 1: Reward-specific gradient similarity distribution in GRPO-based LLM-Rec, where $\mathbf{g}_{\text{acc}}$, $\mathbf{g}_{\text{div}}$, and $\mathbf{g}_{\text{gr}}$ denote the per-reward gradients for accuracy, diversity, and groundedness, respectively.

We design four recommendation-specific reward signals: accuracy (encouraging next-item match), groundedness (ensuring history-consistent rationale), diversity (mitigating popularity bias), and format compliance (enforcing schema-adherent output). Unlike prior work that simply adds format compliance as another reward term (Lin et al., 2025; Park et al., 2026), we remove the format signal from reward aggregation and instead enforce it as a feasibility constraint by masking out outputs that violate production requirements, so they do not contribute to policy updates. Conditioned on this feasibility constraint, we prioritize recommendation accuracy by maximizing the alignment of the update with the accuracy gradient, subject to auxiliary compatibility constraints. We compute reward-wise advantages and gradient proxies via per-reward group normalization, and determine the final update direction using simplex-constrained weights. This primary-centered formulation explicitly accounts for gradient conflicts while reducing the risk that auxiliary objectives dominate the optimization direction.

Experiments across five Amazon Review domains and MovieLens, compared against strong collaborative filtering and LLM-based generative recommender baselines, show improvements across most settings in recommendation accuracy, diversity, and rationale coherence (groundedness). Moreover, **our method is deployed in a large-scale production service**, where a real-world online A/B test shows the highest click-through rate among production baselines, validating its practical effectiveness.

2 Background and Motivation

Recent reinforcement learning with Verifiable rewards (RLVR) methods such as Group Relative Policy Optimization (GRPO) have become popular in generative recommendation due to their computational efficiency and implementation simplicity. In recommendation research, GRPO has primarily been applied with recommendation-specific reward functions that emphasize recommendation accuracy, as demonstrated in works such as Rec-R1 (Lin et al., 2025) and TWiCE-Rec (Park et al., 2026). As generative models become more capable, however, recent studies have begun incorporating additional reward signals—such as rationale coherence and recommendation diversity—to better reflect human-aligned objectives. In most existing approaches, multi-reward optimization is performed by aggregating multiple reward signals using fixed weights and applying GRPO to optimize the policy with respect to the combined reward.

GRPO Formulation. Formally, in the recommendation task, for each user i, given a prompt q^i consisting of the user history and a candidate item set, the policy $\pi_{\theta_{\text{old}}}$ samples G outputs $\{o^{(i,j)}\}_{j=1}^{G}$. Assuming K reward objectives, the aggregated reward for the j-th rollout is computed as the sum of individual reward values:

$$r_{\text{sum}}^{(i,j)} = r_1^{(i,j)} + r_2^{(i,j)} + \ldots + r_K^{(i,j)}, \tag{1}$$

where $r_k^{(i,j)}$ denotes the k-th reward evaluated on rollout $o^{(i,j)}$. The group-relative advantage for the j-th rollout of user i is computed by normalizing the aggregated rewards within the group:

$$A_{\text{sum}}^{(i,j)} = \frac{r_{\text{sum}}^{(i,j)} - \text{mean}\{r_{\text{sum}}^{(i,1)}, \ldots, r_{\text{sum}}^{(i,G)}\}}{\text{std}\{r_{\text{sum}}^{(i,1)}, \ldots, r_{\text{sum}}^{(i,G)}\}}. \tag{2}$$

The corresponding multi-reward GRPO optimization objective can then be expressed as:

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{\{o^{(i,j)}\}_{j=1}^{G} \sim \pi_{\theta_{\text{old}}}(\cdot|q^i)} \left[\frac{1}{G} \sum_{j=1}^{G} \min \left(s^{(i,j)}(\theta)\, A_{\text{sum}}^{(i,j)},\ \text{clip}(s_j(\theta), 1-\epsilon, 1+\epsilon)\, A_{\text{sum}}^{(i,j)} \right) \right] \tag{3}$$

where $s^{(i,j)}(\theta) = \frac{\pi_\theta(o^{(i,j)} \mid q^i)}{\pi_{\theta_{\text{old}}}(o^{(i,j)} \mid q^i)}$ and ϵ denotes the clipping threshold. For clarity, we omit the KL-divergence loss term in this formulation. Rec-R1 (Lin et al., 2025) and TWiCERec (Park et al., 2026) adopt GRPO for training generative recommenders: Rec-R1 optimizes an item-match reward (i.e., whether the generated item matches the ground-truth next item), while TWiCERec extends this formulation by additionally incorporating collaborative-signal-based rewards over generated items.

However, in multi-reward settings, GRPO exhibits a fundamental limitation for generative recommendation. Directly normalizing and aggregating multiple rollout rewards (Eq. 1) can cause distinct reward combinations to collapse into nearly identical advantage values, effectively diminishing the discriminative power of the training signal. This collapse obscures meaningful differences across reward dimensions, leading to imprecise policy updates, suboptimal convergence, and, in severe cases, unstable or premature training failure.

GDPO Formulation. To mitigate this issue, Group reward-Decoupled Normalization Policy Optimization (GDPO) (Liu et al., 2026) was proposed to maintain inter-reward distinctions and improve the fidelity of advantage estimation under multi-reward settings. While GRPO normalizes the summed reward at the group level, GDPO first normalizes each reward component within the group and aggregates them afterward. Specifically, rather than summing all K rewards first (as in Eq. 1) and then applying group-wise normalization to obtain A_{sum} (Eq. 2), GDPO computes the normalized advantage for each reward for the j-th rollout as:

$$A_1^{(i,j)} = \frac{r_1^{(i,j)} - \text{mean}\{r_1^{(i,1)}, \ldots, r_1^{(i,G)}\}}{\text{std}\{r_1^{(i,1)}, \ldots, r_1^{(i,G)}\}}, \quad \ldots, \quad A_K^{(i,j)} = \frac{r_K^{(i,j)} - \text{mean}\{r_K^{(i,1)}, \ldots, r_K^{(i,G)}\}}{\text{std}\{r_K^{(i,1)}, \ldots, r_K^{(i,G)}\}} \tag{4}$$

The final policy-update advantage is computed by aggregating the per-reward normalized advantages:

$$A_{\text{sum}}^{(i,j)} = A_1^{(i,j)} + \cdots + A_K^{(i,j)}. \tag{5}$$

Then applying batch-wise advantages normalization to the sum of the multi-reward advantages, which ensures that the numerical scale of the final advantage $\hat{A}_{\text{sum}}^{(i,j)}$ remains stable and does not grow as additional rewards are introduced:

$$\hat{A}_{\text{sum}}^{(i,j)} = \frac{A_{\text{sum}}^{(i,j)} - \text{mean}\left\{ A_{\text{sum}}^{(i',j')} \mid i' \in D_{\text{Batch}},\ j' = 1, \ldots, G \right\}}{\text{std}\left\{ A_{\text{sum}}^{(i',j')} \mid i' \in D_{\text{Batch}},\ j' = 1, \ldots, G \right\} + \epsilon}, \tag{6}$$

where D_{Batch} denotes the set of samples in the current mini-batch.

However, GDPO still aggregates normalized rewards through a fixed uniform summation (Eq. 5), without accounting for structural interactions among rewards. In recommender systems, such objectives often conflict—for example, improving recommendation accuracy may reduce diversity, while enforcing rationale coherence can constrain recommendation accuracy. Consequently, naïvely aggregating per-reward normalized advantages can propagate these conflicting directions into the final policy update, partially canceling the optimization signal from the primary reward.

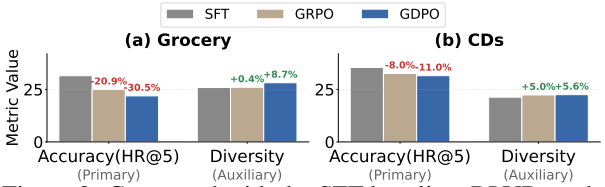

Figure 2: Compared with the SFT baseline, RLVR methods (GRPO/GDPO) improve recommendation diversity but degrade accuracy in the Grocery and CDs domains.

As a result, RL updates such as GRPO and GDPO, applied after SFT, can reduce accuracy relative to the SFT-initialized model, while yielding gains primarily in diversity (Fig. 2), a trend further

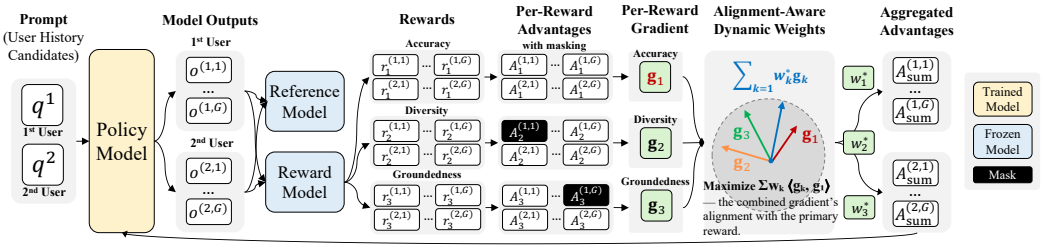

Figure 3: Overview of Groupwise Gradient-Aligned Reward-Decoupled Policy Optimization (G^{2}RPO), illustrated with a mini-batch size of 2. G^{2}RPO maximizes estimated accuracy progress among auxiliary-safe convex combinations.

confirmed by our ablation study in Section 4.3. We therefore seek to optimize multiple reward dimensions while treating recommendation accuracy as the primary objective. Unlike general multi-objective settings such as MGDA (Sener & Koltun, 2018) or PCGrad (Yu et al., 2020) which aim to balance all objectives equally, our formulation assigns asymmetric roles to the rewards: accuracy defines the optimization objective, while auxiliary rewards define compatibility constraints on the update direction.

Findings: We find that composite-reward RLVR often improves auxiliary metrics such as diversity while degrading the primary metric, recommendation accuracy, revealing a reward conflict between auxiliary and primary objectives.

3 Groupwise Gradient-Aligned Reward-Decoupled Policy Optimization

We propose Groupwise Gradient-Aligned Reward-Decoupled Policy Optimization (G^{2}RPO), an alignment-aware dynamic weighting scheme for multi-reward aggregation in RLVR. The overall architecture of G^{2}RPO is illustrated in Fig. 3, with the procedure detailed in Algorithm 1 (Appendix B).

3.1 Problem Setup

Definition 1. User Sequence: We represent a user's interaction sequence as $X = (x_1, x_2, \ldots, x_n)$, where interactions are ordered chronologically and n is the sequence length. Each interaction x is associated with an item's textual metadata (title and description) and, when available, user-generated feedback such as a review and a rating. Then, letting $\mathcal{V}$ denote the set of all items, we define the ground-truth next item as $x_{n+1} = x^+ \in \mathcal{V}$, corresponding to the item the user interacted with after the historical sequence X. Based on this formulation, we construct a training dataset $S = \{(X, x^+)\}$.

Definition 2. Recommendation Rationale: Based on user sequence data, we set post-hoc rationale supervision that explains why a particular next item x^+ should be recommended given the user's history X. We adopt the publicly released rationale dataset from prior work (CURE[1]), which was generated via an agentic workflow using frontier LLMs, resulting in a large-scale, costly collection of over 100K samples drawn from the Amazon Reviews and MovieLens datasets. We denote the ground-truth rationale supervision signal as r^+ and the corresponding candidate item set as $C \subseteq \mathcal{V}$, where $x^+ \in C$. Finally, we set the final training dataset $S = (X, C, x^+, r^+)$.

In recommendation settings, reinforcement-learning-based alignment is commonly applied after an initial **supervised fine-tuning (SFT)** stage using recommendation-specific training data. Using the previously constructed dataset $S = (X, C, x^+, r^+)$, we created instruction data for this task by prompting the model to generate both the rationale r^+ and the next item x^+, given the user history X. See Appendix D for details.

3.2 Multi-Rewards Generative Recommendation

In GRPO and GDPO, the LLM policy generates G rollouts for each given prompt during training. During rollout generation, the input prompt includes explicit instructions to produce both the recommended item and its corresponding rationale within predefined tags (Appendix D). Accordingly, each generated rollout $o^{(i,j)}$ consists of a recommended item $\hat{x}$ and its associated rationale $\hat{r}$, i.e., $o^{(i,j)} = (\hat{x}, \hat{r})$. Each rollout is then evaluated using multiple recommendation-specific reward sig-

[1] https://github.com/cpark88/CURE

nals, capturing distinct objectives such as (1) recommendation accuracy, (2) diversity, (3) rationale coherence (groundedness), and (4) format compliance.

(1) **Accuracy** $r_{\text{acc}}(o^{(i,j)})$: We compute an item-matching indicator: $\mathbb{I}[\hat{x} = x^+]$, where x^+ denotes the ground-truth next item. This corresponds to the primary objective of recommender systems, namely improving recommendation accuracy.

(2) **Diversity** $r_{\text{div}}(o^{(i,j)})$: To mitigate popularity bias and promote more balanced item exposure, we define a diversity reward based on the global popularity of the recommended item. Let $\mathcal{V}$ denote the full item set and $\texttt{cnt}(x)$ the number of interactions for item x recorded in the training data. The popularity of item x is: $p(x) = \frac{\texttt{cnt}(x)}{\sum_{y \in \mathcal{V}} \texttt{cnt}(y)}$. We convert popularity to a raw diversity score via the self-information: $f(x) = -\log\big(p(x) + \epsilon\big)$, where ϵ is a small constant for numerical stability. Finally, we normalize across the item set with min-max scaling:

$$r_{\text{div}} = \frac{f(x) - f_{\min}}{f_{\max} - f_{\min}}, \tag{7}$$

where $f_{\min}$ and $f_{\max}$ are the minimum and maximum raw diversity values over all items in $\mathcal{V}$. By construction $r_{\text{div}} \in [0, 1]$, with $r_{\text{div}} \approx 0$ for the most popular items and $r_{\text{div}} \approx 1$ for the rarest items.

(3) **Groundedness** $r_{\text{gr}}(o^{(i,j)})$: We define a groundedness reward to measure whether a generated rationale $\hat{r}$ helps identify the ground-truth item x^+ among a candidate set C beyond what is possible from the user history alone. Let $C = \{c_1, \ldots, c_N\}$ denote the candidate items in the prompt, X the user's purchase history, and $x^+ \in C$ the ground-truth target item. To prevent trivial copying, we first apply an anti-cheating mask that replaces every occurrence of candidate titles and item IDs in $\hat{r}$ with placeholder tokens, yielding a masked rationale $\tilde{r}$.

We then construct two queries using a sentence encoder ψ (e.g., Sentence-BERT (Reimers & Gurevych, 2019)): $q_0 = \psi(X)$ and $q_1 = \psi(X \oplus \tilde{r})$, where $\oplus$ denotes text concatenation. For each query q_k, we rank the ground-truth item x^+ among all candidates by cosine similarity:

$$\text{rank}_k(x^+) = 1 + \left|\left\{j : \psi(x_j)^\top q_k > \psi(x^+)^\top q_k\right\}\right|, \quad k \in \{0, 1\},\ \forall x_j \in C, \tag{8}$$

where $|\cdot|$ denotes set cardinality. We convert each rank into a percentile score,

$$p_k = \frac{N - \text{rank}_k(x^+)}{N - 1}, \tag{9}$$

so that $p_k = 1$ if x^+ is ranked first and $p_k = 0$ if it is ranked last. Finally, we define groundedness as the fraction of the remaining room for improvement that is recovered by the rationale:

$$r_{\text{gr}} = \begin{cases} \frac{p_1 - p_0}{1 - p_0} & \text{if } p_1 > p_0 \text{ and } p_0 < 1, \\ 0 & \text{otherwise.} \end{cases} \tag{10}$$

Thus, $r_{\text{gr}} = 1$ when the rationale moves the ground-truth item from its baseline position to the top rank, and $r_{\text{gr}} = 0$ when it provides no improvement or makes the ranking worse.

(4) **Format Compliance** $r_{\text{fmt}}(o^{(i,j)})$: a binary reward (i.e., 0 or 1) indicating strict adherence to the prescribed output schema, requiring the rationale to appear within `<think> ... </think>` and the recommended item within `<item> ... </item>` tags.

3.3 Feasibility Constraints: Format Compliance as a Hard Gate

In industrial generative recommenders, format compliance is a minimum requirement rather than a preference to be traded off. Moreover, once instruction-following is learned via SFT, format compliance becomes a relatively easy objective in the subsequent RL stage (Tsai et al., 2024). We therefore remove format compliance from reward aggregation and enforce it as a feasibility gate:

$$m^{(i,j)} = \mathbb{I}\Big[o^{(i,j)} \text{ satisfies format compliance}\Big] \in \{0, 1\}. \tag{11}$$

Rollouts that violate constraints do not contribute to policy updates by masking their advantages:

$$A^{(i,j)} \leftarrow m^{(i,j)} A^{(i,j)}. \tag{12}$$

This prevents reward hacking behaviors (e.g., perfectly formatted but semantically wrong outputs) from shaping the optimization signal.

3.4 From Fixed Weighted-Sum to Alignment-Aware Dynamic Weighting

Following GDPO, we compute per-reward group-normalized advantages within each prompt group according to Eq. 4. Recall that in GDPO, per-reward advantages are aggregated with fixed uniform weights: $A_{\text{sum}}^{(i,j)} = \sum_{k=1}^{K} w_k A_k^{(i,j)}$, where GDPO corresponds to a fixed aggregation with $w_k = 1$, collapsing heterogeneous reward signals into a single update direction. We retain GDPO-style per-reward normalization but replace fixed summation with alignment-aware dynamic weighting. We treat accuracy as the primary objective p and the remaining rewards as auxiliaries $A = \{1, \ldots, K\} \setminus \{p\}$. Among convex combinations whose resulting update is non-adversarial to the auxiliary gradients up to a tolerance ϵ, we select the direction that maximizes alignment with the primary accuracy gradient.

Gradient Proxies from Group-Normalized Advantages. To calculate the policy update direction derived from these rewards, for each reward objective, we approximate the parameter update direction that would arise if the policy were optimized solely with respect to that reward. Let $\phi^{(i,j)}$ denote a vector proxy for the resulting policy gradient contribution of rollout $o^{(i,j)}$: $\phi^{(i,j)} \triangleq \nabla_\theta \log \pi_\theta\big(o^{(i,j)} \mid q^i\big)$. Then, for each reward $k \in \{1, \ldots, K\}$ with mini-batch size B, we define a reward-induced update direction (i.e., gradient) that captures the policy-gradient contribution of reward k for the given sample: $\mathbf{g}_k \triangleq \frac{1}{BG} \sum_{i=1}^{B} \sum_{j=1}^{G} A_k^{(i,j)} \phi^{(i,j)}$.

Primary-Aligned Weight Selection via a Simplex Linear Program. We determine per-reward aggregation weights $\mathbf{w} = [w_1, \ldots, w_K]^\top$ by solving a linear program on the probability simplex. The resulting policy update direction is given by $\Delta\theta \propto \sum_{k=1}^{K} w_k \mathbf{g}_k$. Note that the policy update direction of GDPO is $\Delta\theta \propto \sum_{k=1}^{K} \mathbf{g}_k$). To quantify how closely this update direction $\sum_{k=1}^{K} w_k \mathbf{g}_k$ aligns with each reward gradient $\mathbf{g}_\ell$—i.e., how much the effective update resembles optimizing reward ℓ—we measure the inner-product projection:

$$\left\langle \sum_{k=1}^{K} w_k \mathbf{g}_k,\ \mathbf{g}_\ell \right\rangle = \sum_{k=1}^{K} w_k \langle \mathbf{g}_k,\ \mathbf{g}_\ell \rangle, \tag{13}$$

where $\langle \cdot, \cdot \rangle$ denotes the inner product. Then we define the Gram-like matrix $M \in \mathbb{R}^{K \times K}$ with entries $M_{k\ell} \triangleq \langle \mathbf{g}_k, \mathbf{g}_\ell \rangle$, and let $\mathbf{h}_\ell \triangleq M_{:,\ell}$ denote its ℓ-th column (reward), i.e., $\mathbf{h}_\ell = [\langle \mathbf{g}_1, \mathbf{g}_\ell \rangle, \ldots, \langle \mathbf{g}_K, \mathbf{g}_\ell \rangle]^\top$. Then $\mathbf{w}^\top \mathbf{h}_\ell = \langle \sum_{k=1}^{K} w_k \mathbf{g}_k,\ \mathbf{g}_\ell \rangle$ measures the first-order effect of the weighted-sum update direction on reward ℓ. When reward ℓ corresponds to a primary reward objective such as accuracy, this quantity can be interpreted as the weighted sum of gradient similarities between the primary reward and the other reward objectives.

Linear Program (LP) Formulation. Under this formulation, we choose the weights to maximize the alignment between the weighted-sum update direction and the primary reward gradient, while imposing compatibility constraints with auxiliary reward gradients:

$$\begin{aligned} \max_{\mathbf{w}} \quad & \mathbf{w}^\top \mathbf{h}_p \\ \text{s.t.} \quad & \mathbf{w}^\top \mathbf{h}_a \geq -\epsilon,\ \forall a \in A, \\ & \mathbf{w} \geq 0, \quad \mathbf{1}^\top \mathbf{w} = 1, \end{aligned} \tag{14}$$

where the first constraint requires the aggregated update to have a non-negative (up to slack ϵ) first-order projection onto each auxiliary gradient. The non-negativity constraint $\mathbf{w} \geq 0$ ensures that rewards are combined without reversing their update directions, while the simplex constraint $\mathbf{1}^\top \mathbf{w} = 1$ enforces a convex combination of gradients for stable and interpretable aggregation. The tolerance parameter $\epsilon \geq 0$ allows limited first-order degradation to avoid infeasibility when multiple auxiliary constraints are simultaneously active. When $\epsilon = 0$, Eq. 14 reduces to a strict non-harm formulation. We use $\epsilon = 0.05$ throughout. The resulting weight vector $\mathbf{w}^*$ is treated as a **stop-gradient** constant and used to form the alignment-aware advantage: $A_{\text{sum}}^{(i,j)} = \sum_{k=1}^{K} w_k^* A_k^{(i,j)}$. We then apply a batch-wide standardization to stabilize the scale of $A^{(i,j)}$. Note that Eq. 14 hard-constrains only the auxiliaries; accuracy is prioritized through the objective rather than guaranteed. Empirically, the realized update still has positive estimated accuracy progress ($w^{\star\top} h_p > 0$) in over 95% of training steps, and adding an explicit constraint $w^\top h_p \geq 0$ yields no accuracy gain.

Remark 1. Feasibility of the Primary-Only Update. A natural primary-only candidate is the one-hot choice $\mathbf{w} = \mathbf{e}_p$. Under this choice, each auxiliary compatibility constraint reduces to $\langle \mathbf{g}_p, \mathbf{g}_a \rangle \geq -\epsilon$, so the primary-only update becomes infeasible when it sufficiently conflicts with

any auxiliary gradient. Moreover, even when feasible, $\mathbf{w} = \mathbf{e}_p$ need not maximize Eq. 14, since an auxiliary gradient may exhibit a larger positive projection onto $\mathbf{g}_p$. G^{2}RPO therefore selects the feasible convex combination that maximizes primary-gradient alignment, rather than explicitly maximizing the weight on the primary reward. As shown in Fig. 4, GDPO uses fixed unit weights, whereas unconstrained trainable weights concentrate around 0.3–0.4 across rewards.

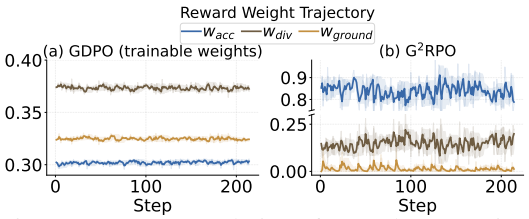

Figure 4: Per-step evolution of reward aggregation weights during training.

Remark 2. Time Complexity. The additional cost of our alignment-aware weighting is negligible. Once the reward-specific gradient proxies are computed, forming the $K \times K$ Gram matrix requires $O(K^2 d)$ operations, where d is the parameter-space dimension, and the subsequent simplex LP involves only K variables and $O(K)$ linear constraints. Since K is fixed and very small in our setting, with three active rewards after treating format compliance as a hard gate, the LP solve contributes only constant per-step overhead relative to the overall LLM training scale. Thus, it does not change the asymptotic complexity, which is dominated by the LLM forward/backward passes. Empirically, under the profiling setup detailed in Appendix B, alignment-aware weighting adds about 1% per-step wall-clock overhead (3.52 $\rightarrow$ 3.55 second/sample) after warm-up compared with the same pipeline without this module.

How does G^{2}RPO prioritize accuracy while controlling auxiliary degradation? Let $d^* = \sum_{k=1}^{K} w_k^* \mathbf{g}_k$ be the optimal policy update direction, and let J_k be the objective for reward k. Assuming each J_k is L_k-smooth (has a Lipschitz-continuous gradient) and the proxy gradient error is bounded in the update direction, every auxiliary reward objective satisfies $J_a(\theta + \eta d^\star) \geq J_a(\theta) - \eta\epsilon - \left(\eta\delta_a + \frac{L_a}{2}\eta^2\right)\|d^\star\|^2$, where η is the step-size and δ_a bounds the proxy-gradient error along the update direction. In other words, the worst-case degradation of each auxiliary objective is upper-bounded by $\eta\epsilon + \left(\eta\delta_a + \frac{L_a}{2}\eta^2\right)\|d^\star\|^2$. Thus, with a small step size η (and small slack/error terms ϵ, δ_a), the degradation of auxiliary objectives remains controlled. For the primary objective, Appendix E further shows that its improvement is lower-bounded by the proxy-level first-order gain $\eta\, \mathbf{w}^{*\top} \mathbf{h}_p$ minus proxy-gradient error and curvature terms; hence, primary improvement is guaranteed when this first-order gain dominates these terms, although Eq. 14 itself does not impose monotonic primary improvement. Full statements and proofs are provided in Appendix E.

4 Experiments

In this section, we address the following research questions: **RQ1** (Accuracy). Does our model improve recommendation performance? **RQ2** (Diversity). To what extent does our method enhance recommendation diversity? **RQ3** (Groundedness). Are the generated rationales more faithful than those of SOTA baselines? **RQ4** (Ablation). Which components of G^{2}RPO drive the performance improvements the most? **RQ5** (Online Impact). Does G^{2}RPO increase click-through-rate over baselines in online A/B tests?

4.1 Datasets and Evaluation Settings

We evaluate our approach on five domains from **Amazon Reviews 2023** (Hou et al., 2024) and on the **MovieLens** benchmark (Harper & Konstan, 2015), covering diverse recommendation settings from e-commerce review interactions to movie consumption; full dataset statistics are reported in Appendix A. Specifically, the Amazon collection includes five practically relevant domains: (1) *Amazon **Fashion***, (2) ***Clothing** Shoes and Jewelry*, (3) ***Grocery** and Gourmet Food*, (4) ***Health** and Household*, and (5) ***CDs** and Vinyl*. We additionally use MovieLens 32M (***ml-32m***), a widely adopted large-scale benchmark for movie recommendation.

For comparison, we consider two families of baselines: collaborative-filtering-based recommendation methods (**CF-Rec**) and LLM-based recommendation methods (**LLM-Rec**). We adopted `Gemma-3-4B-it` (Team et al., 2025) as the LLM backbone, taking into account deployment constraints in real-world environments. We evaluated recommendation performance using the *leave-one-out* method. We construct the candidate set by including the ground-truth next item together with items the user has not interacted with. These candidate sets are incorporated into the prompt construction for both training and inference. We focused on standard evaluation metrics, including *Hit Ratio* (**HR**) and *Normalized Discounted Cumulative Gain* (**NDCG**). For reproducibility, implementation details are described in Appendix B.

Table 1: Accuracy comparison on ***Amazon*** and ***MovieLens***; results are averaged over 3 seeds, with paired user-level significance tests. The best and second-best results are marked in **bold** and underline, respectively, and † indicates significant improvement over the strongest baseline ($p < 0.05$).

Domain		Fashion			Clothing			Grocery			Healths			CDs			MovieLens		
Types	**Models**	HR@1	HR@5	N@5	HR@1	HR@5	N@5	HR@1	HR@5	N@5	HR@1	HR@5	N@5	HR@1	HR@5	N@5	HR@1	HR@5	N@5
CF Rec	SASRec	4.21	21.55	12.69	3.37	16.74	9.89	6.08	24.43	15.20	4.00	19.06	11.44	4.14	19.40	11.63	2.76	10.24	6.49
	BERT4Rec	3.03	15.79	9.30	2.60	12.77	7.55	2.96	15.68	9.11	2.92	14.92	8.72	3.14	14.96	8.89	2.43	9.71	6.00
	NextItNet	1.34	7.78	4.42	3.15	13.73	8.36	3.16	14.72	8.86	2.86	13.39	8.00	3.68	15.65	9.57	2.34	9.66	5.93
	GRU4Rec	1.89	11.14	6.33	3.34	13.84	8.49	5.32	22.50	13.72	3.08	14.66	8.70	3.51	15.72	9.49	2.96	10.67	6.74
	LightSANs	3.66	16.65	9.94	2.95	15.85	9.25	3.52	16.12	9.63	3.21	16.06	9.48	3.26	16.38	9.66	2.74	9.94	6.32
	FPMC	3.05	15.84	9.31	2.39	11.85	6.95	3.44	16.19	9.70	3.15	16.27	9.57	3.32	17.26	10.14	0.67	3.14	1.91
LLM Rec	Rec-SAVER	12.72	39.60	26.92	8.09	26.82	17.70	10.03	27.92	19.48	4.51	18.29	11.47	10.67	35.22	23.18	3.29	10.69	7.26
	$\text{SumRec}_{\text{DPO}}$	8.06	28.65	18.55	3.84	15.19	9.89	4.62	14.43	9.90	3.97	18.74	11.82	3.27	15.10	9.31	0.79	7.46	3.94
	GDPO	12.61	38.94	26.10	8.15	24.51	16.63	7.30	23.27	15.52	4.65	17.70	9.68	9.42	30.80	20.31	1.32	7.45	4.56
	S-DPO	5.22	23.78	14.38	4.16	19.37	11.75	6.57	18.47	12.91	2.76	15.71	9.28	6.70	29.66	18.28	1.63	8.02	4.76
	Rec-R1	15.34	37.40	26.84	7.05	25.50	16.79	10.23	25.93	18.61	5.12	14.55	10.23	15.23	32.59	24.66	3.15	8.91	6.25
	TWiCERec	12.16	30.49	21.71	5.62	21.88	14.06	5.65	17.95	11.94	3.90	16.64	10.00	6.10	23.82	15.00	1.57	7.49	4.41
	PCGrad	20.90	41.79	32.50	**10.90**	25.06	18.63	11.14	21.00	16.88	6.22	19.41	13.32	15.70	32.86	25.13	2.93	9.51	6.42
	CAGrad	20.02	41.55	31.04	9.61	23.18	17.02	10.36	22.74	16.55	6.58	19.86	13.61	15.94	34.20	25.88	3.11	9.88	6.74
	MGDA	18.35	31.58	23.30	10.20	24.47	18.15	10.54	23.92	18.11	6.71	18.42	13.14	15.16	**37.63**	27.39	**3.75**	4.33	4.11
	G^2RPO	**22.21**†	**45.97**†	**34.70**†	10.46	**29.65**†	**19.90**†	**12.12**	**28.15**	**20.82**†	**7.14**	**21.60**†	**14.73**†	**17.58**†	36.54	**27.79**	3.29	**11.72**†	**7.64**†

4.2 Performance Evaluation (RQ1–3)

Recommendation Accuracy (RQ1) In this section, given a user history and a candidate item set, we aim to identify the ground-truth next item. As shown in Table 1, we observe that our method achieves competitive recommendation accuracy across domains compared to CF-Rec and LLM-Rec baselines, showing improvements over the second-best method in several cases. The gap between CF-Rec and LLM-Rec baselines further suggests that textual semantics and reasoning-aware preference modeling provide complementary signals beyond conventional CF-Rec.

Recommendation Diversity (RQ2) In recommender systems, optimizing solely for accuracy often amplifies popularity bias, leading the model to repeatedly recommend already popular items while underexposing less frequent ones. To quantify how well a method mitigates this bias, we evaluate the popularity-based diversity. Let $p(x)$ denote the normalized popularity of item x estimated from the training data. Given a top-K recommendation list $R_u@K$ for user u, we compute $\text{Diversity@K}(u) = \frac{1}{K}\sum_{x \in R_u@K} - \log p(x)$. For comparability, we further normalize the score into $[0, 1]$ using dataset-level min–max scaling based on the maximum and minimum item popularity. The final score is averaged across all users; values near 1 indicate strong long-tail emphasis, while values near 0 indicate high popularity bias. Our empirical results reveal clear differences across model families. CF-based recommenders (e.g., SASRec, GRU4Rec) tend to exhibit stronger popularity bias, as their training dynamics are heavily influenced by frequently interacted items, resulting in lower diversity at test time. Moreover, compared to LLM-based recommendation baselines, our approach achieves higher diversity in most domains.

Table 2: Comparison of recommendation diversity across baselines.

Domain	Metric	SASRec	GRU4Rec	Rec-SAVER	GDPO	Rec-R1	TWiCERec	Ours
Fashion	Div@1	0.2593	0.1701	0.1991	0.2410	0.2968	0.2900	**0.3115**†
	Div@5	0.2301	0.1764	0.2270	0.2807	0.2641	0.2700	**0.2850**
Clothing	Div@1	0.2663	0.0910	0.2948	**0.3005**	0.2900	0.2900	0.2964
	Div@5	0.2622	0.1811	0.2989	0.2984	0.2940	0.2911	**0.3020**
Grocery	Div@1	0.2208	0.2199	0.2447	0.2905	0.2454	0.2488	**0.3288**†
	Div@5	0.2309	0.2217	0.2736	0.2729	0.2747	0.2600	**0.3348**†
Healths	Div@1	0.1124	0.0452	0.1012	0.1132	0.1062	0.0999	**0.1188**
	Div@5	0.0832	0.0548	0.1010	0.1119	0.1046	0.1000	**0.1208**†
CDs	Div@1	0.1585	0.0340	0.2130	0.2205	0.2282	0.2200	**0.2393**†
	Div@5	0.1586	0.0715	0.2130	0.2295	0.2191	0.2200	**0.2394**†
MovieLens	Div@1	0.0068	0.0009	0.0320	0.0372	0.0385	0.0311	**0.0395**
	Div@5	0.0210	0.0250	**0.0388**	0.0371	0.0356	0.0312	0.0372

Evaluating Rationale Quality (RQ3) We quantitatively evaluate rationale quality using an LLM-as-a-Judge protocol with human validation (Fig. 5). Our model achieves the highest proportion of top-rated persuasive rationales (90.83%), improving over the strongest baseline, Rec-R1, by +3.45%p, with full details provided in Appendix F.

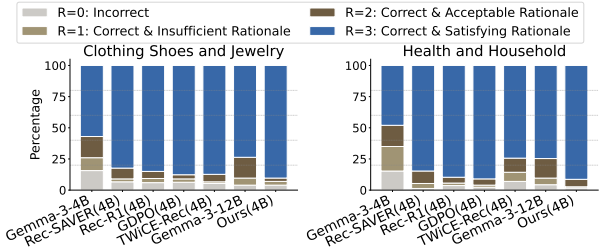

Figure 5: GPT-4o Evaluation of Rationale Quality.

4.3 Ablation Study (RQ4)

As shown in Fig. 6, we conduct a component-wise ablation study to isolate the contribution of each design choice in G^2RPO. We compare eight variants: (A) **Base** (`Gemma-3-4B-it`); (B) **+SFT** (instruction tuning for recommendation tasks); (C) **+SFT+GRPO** (GRPO applied to optimize an

aggregated multi-reward signal); (D) **+SFT+GDPO** (per-reward normalization with uniform aggregation weights); (E) **+SFT+G^{2}RPO** (accuracy-only weighting with $\mathbf{w}=\mathbf{e}_p$ + format-as-hard gating); (F) **+SFT+G^{2}RPO** (LP-constrained dynamic weighting + format-as-reward); (G) **+SFT+G^{2}RPO** (Trainable weighting + format-as-hard gating); (H) **+SFT+G^{2}RPO** (our full method).

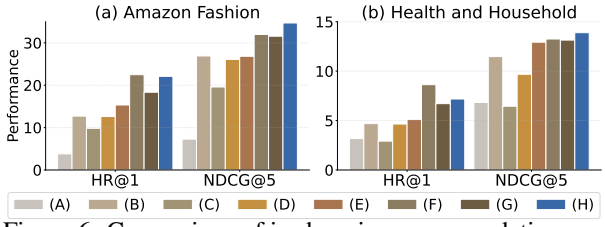

Figure 6: Comparison of in-domain recommendation performance across model variants.

(1) Recommendation-oriented adaptation is essential. The pretrained base model (A) is consistently outperformed by all recommendation-adapted variants (B–H), indicating that aligning the model with recommendation-style instruction following and evaluation criteria is necessary beyond general-purpose pretraining.

(2) Naive multi-reward optimization can underperform SFT. Despite optimizing multiple recommendation objectives, GDPO variants (C–D) yield lower accuracy than SFT-only training (B), suggesting that assigning equal weights to potentially conflicting rewards can hinder effective optimization.

(3) Per-reward normalization helps. Compared to GRPO with a single aggregated reward (C), GDPO with per-reward normalization (D) improves robustness under heterogeneous reward scales.

(4) Alignment-aware weighting improves over accuracy-only optimization. (E) and (H) differ only in whether $\mathbf{w}$ is fixed to $\mathbf{e}_p$ or chosen by the LP, and (H) is better on both metrics in both domains, so the gain is attributable to the LP weighting itself.

(5) Enforcing format via hard gating is more effective than rewarding it. Replacing the format-as-reward design (F) with our masking-based hard gate (H) yields more reliable improvements. This supports treating format compliance as a feasibility constraint to reduce reward hacking and prevent trivial objectives from dominating learning.

(6) Unconstrained trainable weighting may dilute the focus on accuracy. With end-to-end trainable aggregation weights (G), the model converges to near-uniform weights across accuracy, groundedness, and diversity rewards, suggesting limited differentiation among objectives. In contrast, our LP-based weighting selects, among auxiliary-compatible updates, the gradient combination that is most aligned with accuracy (Fig. 4b). This explicitly accounts for gradient conflicts while maintaining a primary-centered optimization direction. Trainable weighting has no notion of gradient conflict and converges to near-uniform weights, so conflicting auxiliary gradients can pull the update away from accuracy. The LP weighting instead keeps the update anchored to the accuracy gradient while constraining first-order harm to auxiliaries within the prescribed tolerance, which empirically maintains strong accuracy.

4.4 Online A/B Test (RQ5)

Table 3: Online A/B Test Results

Model	# Impression	# Clicks	CTR	Latency
Popularity	339.95K	10.46K	3.08%	45ms/call
ML model	438.24K	31.38K	7.16%	60ms/call
SASRec	583.78K	21.13K	3.62%	76ms/call
Ours	711.23K	52.07K	7.32%	138ms/call

We deployed our approach in a live production recommender system and evaluated it through an online A/B test conducted from December 2025 to February 2026. We compared against three baselines: (i) a popularity-based model, (ii) a conventional machine-learning recommender, and (iii) CF-Rec (SASRec). Users were assigned to one of four buckets using a deterministic, user-level hashing scheme to ensure stable randomization. As reported in Table 3, our method achieves a CTR of 7.32%, outperforming all three baselines. Compared with the strongest baseline, the conventional ML recommender, our method yields a modest but statistically significant +0.16 percentage-point gain, corresponding to a +2.2% relative lift. The gains over SASRec are substantially larger, with a +3.70 percentage-point absolute improvement and a +102.3% relative lift; a one-sided two-proportion z-test confirms that this improvement is highly significant ($p < 0.0001$). These results indicate a robust gain under real-world constraints, with an inference latency of approximately 138 ms/call, supporting practical deployability (Appendix G).

5 Conclusion

We proposed G^{2}RPO, an alignment-aware constrained optimization framework for multi-reward generative recommendation. Our approach improves accuracy, diversity, and coherence across benchmarks and delivers significant gains in real-world deployment.

AI USE STATEMENT

We used generative AI tools solely for grammar correction and language polishing. All AI-assisted text was reviewed by the authors, who take full responsibility for the final content of this work.

ETHICS STATEMENT

Our online experiments were conducted in a production recommender system under applicable internal policies and privacy requirements. The experiments used standard recommendation interactions without exposing users to interventions beyond the normal service experience. All reported results are aggregated, and no personally identifiable information or user-level data is disclosed.

REPRODUCIBILITY STATEMENT

We provide detailed information necessary for reproducing our experiments in the Appendix, including model configurations, training and evaluation settings, implementation details, and additional experimental specifications, along with the public release of all our code and datasets used.

REFERENCES

Yuxin Chen, Junfei Tan, An Zhang, Zhengyi Yang, Leheng Sheng, Enzhi Zhang, Xiang Wang, and Tat-Seng Chua. On softmax direct preference optimization for recommendation. *Advances in Neural Information Processing Systems*, 37:27463–27489, 2024.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

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

Yupeng Hou, Jiacheng Li, Zhankui He, An Yan, Xiusi Chen, and Julian McAuley. Bridging language and items for retrieval and recommendation. *arXiv preprint arXiv:2403.03952*, 2024.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.

Jiayi Liao, Xiangnan He, Ruobing Xie, Jiancan Wu, Yancheng Yuan, Xingwu Sun, Zhanhui Kang, and Xiang Wang. Rosepo: Aligning llm-based recommenders with human values. *arXiv preprint arXiv:2410.12519*, 2024.

Jiacheng Lin, Tian Wang, and Kun Qian. Rec-r1: Bridging generative large language models and user-centric recommendation systems via reinforcement learning. *arXiv preprint arXiv:2503.24289*, 2025.

Bo Liu, Xingchao Liu, Xiaojie Jin, Peter Stone, and Qiang Liu. Conflict-averse gradient descent for multi-task learning. *Advances in neural information processing systems*, 34:18878–18890, 2021.

Shih-Yang Liu, Xin Dong, Ximing Lu, Shizhe Diao, Peter Belcak, Mingjie Liu, Min-Hung Chen, Hongxu Yin, Yu-Chiang Frank Wang, Kwang-Ting Cheng, et al. Gdpo: Group reward-decoupled normalization policy optimization for multi-reward rl optimization. *arXiv preprint arXiv:2601.05242*, 2026.

Chung Park, Hyeongjun Yun, Taesan Kim, Junui Hong, Dongjoon Hong, Mira Myong, Jihoon Oh, MinCheol Cho, Kijung Park, Min sung Choi, et al. More than what was chosen: Llm-based explainable recommendation beyond noisy user preferences. In *The Fourteenth International Conference on Learning Representations*.

Chung Park, Taesan Kim, Hyeongjun Yun, Dongjoon Hong, Junui Hong, Kijung Park, MinCheol Cho, Min Sung Choi, Jihwan Seok, and Jaegul Choo. Think wise, collaborate effectively: A rationale-aware llm-based recommender with reinforcement learning from collaborative signals. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 15609–15616, 2026.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*, 2019.

Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. *Advances in neural information processing systems*, 31, 2018.

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

Manato Tajiri and Michimasa Inaba. Refining text generation for realistic conversational recommendation via direct preference optimization. *arXiv preprint arXiv:2508.19918*, 2025.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, et al. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025.

Alicia Y Tsai, Adam Kraft, Long Jin, Chenwei Cai, Anahita Hosseini, Taibai Xu, Zemin Zhang, Lichan Hong, Ed H Chi, and Xinyang Yi. Leveraging llm reasoning enhances personalized recommender systems. *arXiv preprint arXiv:2408.00802*, 2024.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. *Advances in neural information processing systems*, 33: 5824–5836, 2020.

Wayne Xin Zhao, Yupeng Hou, Xingyu Pan, Chen Yang, Zeyu Zhang, Zihan Lin, Jingsen Zhang, Shuqing Bian, Jiakai Tang, Wenqi Sun, Yushuo Chen, Lanling Xu, Gaowei Zhang, Zhen Tian, Changxin Tian, Shanlei Mu, Xinyan Fan, Xu Chen, and Ji-Rong Wen. Recbole 2.0: Towards a more up-to-date recommendation library. In *CIKM*, pp. 4722–4726. ACM, 2022.

A Dataset

We evaluate on five domains from the **Amazon Review 2023** collection[2] (Hou et al., 2024) and the **MovieLens** dataset[3] (Harper & Konstan, 2015). These benchmarks cover diverse recommendation scenarios, from e-commerce product reviews to movie rating behavior (Table 4). The five Amazon domains are: (1) *Amazon* ***Fashion***, (2) ***Clothing*** *Shoes and Jewelry*, (3) ***Grocery*** *and Gourmet Food*, (4) ***Health*** *and Household*, and (5) ***CDs*** *and Vinyl*. For the movie domain, we use MovieLens 32M (***ml-32m***), a widely adopted large-scale benchmark for movie recommendation.

Table 4: Statistics of Training Dataset.

Dataset	Domain	#Users	#Items	Avg Length	Avg Item Purchase	Sparsity
Amazon Review	**Fashion**	11,028	59,004	6.54	1.23	0.9994
	Clothing	13,766	107,353	11.83	1.65	0.9994
	Grocery	11,334	6,149	10.76	21.02	0.9926
	Health	21,152	74,784	8.49	2.50	0.9994
	CDs	17,035	83,774	13.71	1.97	0.9993
MovieLens	**ml-32m**	10,623	12,259	48.96	42.43	0.9960

B Reproducibility and Implementation Details

All experiments were performed on a server with eight NVIDIA H200 GPUs. For reproducibility, we release our implementation via anonymous public links.

■ **Code Link**: https://anonymous.4open.science/r/G2RPO-FDB9

Hyperparameters Table 5 details the hyperparameter settings used across all experiments. We use `Gemma-3-4B-it` (Team et al., 2025) as the shared LLM backbone. All baselines are first trained via supervised fine-tuning (SFT) on a held-out sample set disjoint from the data used in the subsequent reinforcement learning stage; G^{2}RPO then continues training on a separate non-overlapping sample set. For the G^{2}RPO stage, we train for 2 epochs with a batch size of 4, using the 8-bit paged AdamW optimizer under a cosine learning rate schedule (initial rate 5.0×10^{-5}, decaying to a predefined minimum). Gradient accumulation is set to a factor of 8, the warm-up ratio to 0.05, and the maximum input length to 10k tokens. We employ DeepSpeed ZeRO-2 with bf16 mixed precision and gradient checkpointing. Parameter-efficient adaptation is performed with LoRA (Hu et al., 2022) (rank r=8, scaling factor α=16, dropout 0), applied to the q-, v-, o-, $gate$-, up-, and $down$-projection layers. All of the above settings are kept identical across every baseline for fair comparison. For GRPO-based methods, including Rec-R1, TWiCE-Rec, and our model, each sample produces 16 output generations, and we compute the reward mean and standard deviation over the 16 outputs for each user in a mini-batch. For the preceding SFT stage, we follow the same

Table 5: Training hyperparameters for G^{2}RPO.

Hyperparameter	Value
Batch size	4
Epochs	2
Optimizer	Paged AdamW (8-bit)
Learning rate	5.0×10^{-5} (cosine schedule, min LR applied)
Warm-up ratio	0.05
Gradient accumulation steps	8
Max sequence length	10,000
Weight decay	0.01 (default)
Gradient clipping	1.0 (default)
Non-harm tolerance ϵ	0.05
Rollouts per prompt G	16
LoRA rank (r)	8
LoRA α	16
LoRA dropout	0
LoRA target modules	v-, q-, o-, gate-, up-, down-projections
Precision	bf16
Gradient checkpointing	True
Parallelism	DeepSpeed ZeRO-2
Hardware	8 × NVIDIA H200 160GB GPUs

[2] https://amazon-reviews-2023.github.io
[3] https://grouplens.org/datasets/movielens

configuration listed in Table 5, except that the batch size is increased to 12. This SFT checkpoint also serves as a common initialization for all reinforcement-learning-based baselines.

Evaluation Protocol. We evaluate all methods under a leave-one-out protocol. For each user sequence, the most recent interaction is used as the test target, the second most recent interaction is used for validation, and the remaining interactions are used for training. At inference time, each prompt contains the true target item and a candidate set of 200 items per request, constructed from the user-specific pool of previously unseen items by combining random negative sampling with popularity-based hard-negative sampling. The same candidate set construction is applied to all baselines. Since DPO-based baselines rely on explicit preference pairs, we additionally construct negative samples for their training. We use the same user context and candidate set for every prompt across all baselines and our model.

Alignment-Aware Weighting Implementation. Algorithm 1 gives the implementation of a single G^2RPO training step. After sampling multiple responses per prompt, we apply the format gate and compute group-normalized advantages separately for each reward. We then form reward-specific policy-gradient proxies over the LoRA trainable parameters and use their pairwise inner products to construct the Gram matrix used by the alignment-aware linear program.

The solution of the linear program provides the reward weights for the current batch. These weights are detached from the computation graph and used to aggregate the reward-specific advantages before the final policy-gradient update. Since the base model is frozen, all proxy gradients and final updates are computed only over the LoRA parameters.

C Related Methods and Baselines

Traditional collaborative filtering (CF) methods infer user preference patterns from past user–item interactions, such as clicks, purchases, and ratings. Rather than relying on item text or metadata, these approaches usually represent each item as an independent ID and learn embeddings from interaction regularities observed in the data. Because of this design, CF-based recommenders are typically effective when a domain provides sufficiently rich interaction histories.

All CF baselines in our experiments were implemented using the RecBole framework (Zhao et al., 2022)[4]. While not tailored specifically for recommendation tasks, general gradient surgery methods that resolve conflicting objectives (i.e., PCGrad (Yu et al., 2020), CAGrad (Liu et al., 2021), and MGDA (Sener & Koltun, 2018)) are also evaluated as baselines in Table 1. For a fair comparison, these baselines are run on exactly the same training pipeline as G^2RPO: they operate on the same K=3 per-reward gradient proxies computed from group-normalized, format-gated advantages, and replace only our LP-based weighting with their respective (symmetric) update rules. All other settings, including the SFT initialization and hyperparameters in Table 5, are kept identical. Below, we outline the LLM-based baselines considered in this work and briefly describe how each was implemented.

■ **Rec-SAVER** (Tsai et al., 2024) studies LLM-based recommendation with supervised rationale learning, training the model on target items together with explanatory rationales. The method shows that rationale-guided instruction tuning can strengthen recommendation quality under both zero-shot transfer and task-specific fine-tuning, including non-overlapping cross-domain settings. We use our own in-house implementation for this baseline.

■ **S-DPO** (Chen et al., 2024) extends DPO for recommendation by replacing the standard pairwise formulation with a softmax-based objective over one positive item and multiple negatives. This formulation encourages stronger discrimination between the target item and challenging negative candidates during training. We implemented this method independently.

■ **TWiCERec** (Park et al., 2026) augments LLM recommendation with collaborative filtering signals, combining rationale-oriented instruction tuning with CF-based guidance. Its training further incorporates confidence-aware reinforcement learning, where the reward reflects both recommendation correctness and the confidence of an external CF model. This baseline is particularly relevant to our

[4] https://github.com/RUCAIBox/RecBole

LLM-based setting and showed competitive results in the zero-shot non-overlapping cross-domain scenario. Our implementation is based on the authors' official repository[5].

■ **Rec-R1** (Lin et al., 2025) introduces GRPO-style reinforcement learning into recommendation, using next-item correctness as the optimization signal. In this setup, the model is rewarded when the generated item matches the ground-truth next interaction. We implemented this baseline on top of a public GRPO codebase[6].

■ **SumRec$_{\text{DPO}}$** (Tajiri & Inaba, 2025) was originally proposed for conversational recommendation, where DPO is applied using dialogue summaries produced by an LLM together with recommendation supervision. To make it compatible with our sequential single-turn setting, we reformulate the pipeline so that both SFT and DPO operate on single-turn inputs derived from user interaction histories. We implemented this baseline independently.

■ **GDPO** (Liu et al., 2026) is a multi-reward RL method that identifies a key limitation of GRPO: when multiple rewards are summed first and normalized afterward, different reward combinations can collapse into similar advantage signals. GDPO addresses this by normalizing each reward separately before aggregation, thereby preserving finer reward distinctions during policy optimization. While not designed specifically for recommendation, GDPO is closely related to our setting because it provides a strong multi-reward RL baseline for training LLM-based recommenders with heterogeneous reward signals. We implemented this baseline on top of a public GRPO codebase.

D Supervised Fine-Tuning before G^{2}RPO

Training Supervised fine-tuning (SFT) is widely used to adapt a general-purpose LLM to recommendation tasks before applying preference-based optimization. In our setting, this stage starts by transforming recommendation data into instruction-style examples. For each user u, we construct an input prompt q^i that contains the interaction history X, a candidate item set C, and an instruction specifying the sequential recommendation task. The corresponding target output y^i consists of the ground-truth item $x_+ \in C$ and its rationale r_+, forming a supervised training pair (q^i, y^i). The SFT dataset is strictly disjoint from that used in the subsequent G^{2}RPO phase, ensuring no samples are shared across stages.

In the second stage, the LLM-based recommender f_θ is trained on these prompt–output pairs using a causal language modeling objective (Liao et al., 2024). This objective teaches the model to generate the target sequence autoregressively, predicting each token conditioned on the prompt and the previously generated tokens. Under this formulation, recommendation is cast as a sequence generation problem in which the model produces both the recommended item and its rationale. Formally, given training pairs (q^i, y^i), the objective is defined as:

$$\max_{\theta} \sum_{(q^i, y^i)} \sum_{t=1}^{|y^i|} \log P_\theta\left(y_t^i \mid q^i, y_{<t}^i\right), \tag{15}$$

where $|y^i|$ denotes the length of the output sequence, y_t^i is the t-th token, and $y_{<t}^i$ represents all preceding output tokens. Although this objective provides an effective initialization, it mainly fits observed targets and does not explicitly optimize alignment with user preferences, which is central to recommendation.

Prompt Example In SFT, the prompt is formulated to encourage the model to infer the most appropriate item from the candidate set based on the user's purchase history and to generate a natural-language justification for that choice.

[5] https://github.com/cpark88/TWiCE-Rec

[6] https://github.com/huggingface/open-r1

An example prompt for Supervised Fine-Tuning

#Prompt q^i
Based on **[Purchase History]**, use your logical reasoning process to identify the most suitable item for this customer from the **[Candidate List]**. Then, include your reasoning inside the <think></think>tag and the recommended item inside the <item></item>tag.
[Purchase History] (1) Title: Skechers performance women's go walk slip-on walking shoe (2) Title: Walstar womens ballerina ballet flat shoe slip on (3) Title: Naimo women's sequin evening,...(truncated) $\leftarrow X$
[Candidate List]: (omitted for brevity)$\leftarrow C$

#Response y_u
<think>Based on your history of purchasing comfortable and practical women's footwear-such as Skechers slip-ons, walking shoes, and loafer-it's clear that comfort and ease of use are important to you. Many of these shoes, ... (truncated) </think>$\leftarrow r_+$
<item>4 pairs no show lace cotton liner hidden non-skid boat socks women </item>$\leftarrow x_+$

E Safety of Auxiliary Reward Objectives and Dual Interpretation

Notation. Let $J_k(\theta)$ denote the expected objective induced by reward $k \in \{1, \dots, K\}$ under policy π_θ (e.g., correctness, diversity, groundedness, etc.). Let p be the primary reward objective (i.e., correctness), and $\mathcal{A} = \{1, \dots, K\} \setminus \{p\}$ be auxiliaries. For a prompt of user i, define $\phi^{(i,j)} \triangleq \nabla_\theta \log \pi_\theta(o^{(i,j)} \mid q^i)$. Given group-normalized advantages $A_k^{(i,j)}$, define reward-induced gradient proxies

$$\mathbf{g}_k \triangleq \sum_{j=1}^{G} A_k^{(i,j)} \phi^{(i,j)}, \tag{16}$$

where G is the number of rollouts from policy.

Let $M \in \mathbb{R}^{K \times K}$ be the Gram matrix $M_{k\ell} = \langle \mathbf{g}_k, \mathbf{g}_\ell \rangle$, and $\mathbf{h}_\ell \triangleq M_{\cdot,\ell}$ the ℓ-th column so that $\langle \sum_{k=1}^K w_k \mathbf{g}_k, \mathbf{g}_\ell \rangle = \mathbf{w}^\top \mathbf{h}_\ell$. We choose mixture weights $\mathbf{w}^\star$ by

$$\begin{aligned} \max_{\mathbf{w}} \quad & \mathbf{w}^\top \mathbf{h}_p \\ \text{s.t.} \quad & \mathbf{w}^\top \mathbf{h}_a \geq -\epsilon, \ \forall a \in A, \\ & \mathbf{w} \geq 0, \quad \mathbf{1}^\top \mathbf{w} = 1, \end{aligned} \tag{17}$$

and take the weighted-sum update direction (i.e., gradient)

$$d^\star \triangleq \sum_{k=1}^{K} w_k^\star \, \mathbf{g}_k. \tag{18}$$

E.1 Objective-Level Auxiliary Reward Safety under Smoothness and Proxy Error

Assumption 1 (Smoothness of Objectives)**.** Each objective J_k associated with reward k is L_k-smooth: for all θ, θ',

$$J_k(\theta') \geq J_k(\theta) + \langle \nabla J_k(\theta), \theta' - \theta \rangle - \frac{L_k}{2} \|\theta' - \theta\|^2. \tag{19}$$

This is a standard regularity assumption in first-order optimization and RL theory. When the policy network uses common smooth components (e.g., linear layers, LayerNorm, softmax) and we take sufficiently small updates, J_k is locally well-approximated by its first-order Taylor expansion with a quadratic remainder, which is exactly the L_k-smooth inequality above. Empirically, this smoothness condition holds as a local property around the optimization trajectory, making it a mild and widely used assumption.

Assumption 2 (Proxy-Gradient Error)**.** For each reward k, there exists $\delta_k \geq 0$ such that for the chosen update direction $d^\star$,

$$\left| \langle \nabla J_k(\theta) - \mathbf{g}_k, \ d^\star \rangle \right| \leq \delta_k \, \|d^\star\|^2. \tag{20}$$

Note that $\mathbf{g}_k$ is a proxy gradient for J_k. The inner product $\langle \nabla J_k(\theta), d^\star \rangle$ is the *true* first-order improvement of J_k along the update direction, while $\langle \mathbf{g}_k, d^\star \rangle$ is the proxy's *predicted* improvement.

This assumption bounds their discrepancy: $\left|\langle \nabla J_k(\theta), d^\star\rangle - \langle \mathbf{g}_k, d^\star\rangle\right| \le \delta_k \|d^\star\|^2$, so the proxy does not over/under-estimate the improvement by more than a second-order term. In practice, $\mathbf{g}_k$ is constructed from finite-sample rollouts and advantage estimates, hence it is noisy but typically approximately unbiased for $\nabla J_k(\theta)$. With small policy updates (standard in policy optimization), the resulting estimation error affects the improvement prediction only at second order in the step size, which is captured by δ_k.

Theorem 1 (Auxiliary Degradation is Bounded)**.** *Let $d^\star$ be formed by Eq. 17. Under Assumptions 1–2, for any step size $\eta > 0$ and any auxiliary reward $a \in \mathcal{A}$,*

$$J_a(\theta + \eta d^\star) \;\ge\; J_a(\theta) - \eta\,\epsilon - \Big(\eta\,\delta_a + \frac{L_a}{2}\eta^2\Big)\,\|d^\star\|^2. \tag{21}$$

Moreover, the primary objective enjoys the lower bound

$$J_p(\theta + \eta d^\star) \;\ge\; J_p(\theta) + \eta\,(\mathbf{w}^{\star\top}\mathbf{h}_p) - \Big(\eta\,\delta_p + \frac{L_p}{2}\eta^2\Big)\,\|d^\star\|^2. \tag{22}$$

Proof. Fix any k. By smoothness (Assumption 1) with $\theta' = \theta + \eta d^\star$,

$$J_k(\theta + \eta d^\star) \ge J_k(\theta) + \eta\langle \nabla J_k(\theta), d^\star\rangle - \frac{L_k}{2}\eta^2\|d^\star\|^2.$$

Decompose $\langle \nabla J_k(\theta), d^\star\rangle = \langle \mathbf{g}_k, d^\star\rangle + \langle \nabla J_k(\theta) - \mathbf{g}_k, d^\star\rangle$. By Assumption 2, $\langle \nabla J_k(\theta) - \mathbf{g}_k, d^\star\rangle \ge -\delta_k\|d^\star\|^2$. Thus

$$J_k(\theta + \eta d^\star) \ge J_k(\theta) + \eta\langle \mathbf{g}_k, d^\star\rangle - \Big(\eta\delta_k + \frac{L_k}{2}\eta^2\Big)\|d^\star\|^2.$$

For an auxiliary reward a, note $\langle \mathbf{g}_a, d^\star\rangle = \mathbf{w}^{\star\top}\mathbf{h}_a \ge -\epsilon$ by feasibility of $\mathbf{w}^\star$. This yields Eq. 21. For the primary reward p, note $\langle \mathbf{g}_p, d^\star\rangle = \mathbf{w}^{\star\top}\mathbf{h}_p$ by definition of $\mathbf{h}_p$, yielding Eq. 22. □

Eq. 21 decomposes the auxiliary change into two degradation parts: (i) a *first-order* loss $\eta\,\epsilon$ that is explicitly allowed by the LP feasibility constraint $\langle \mathbf{g}_a, d^\star\rangle \ge -\epsilon$, and (ii) a *second-order* loss $\big(\eta\delta_a + \frac{L_a}{2}\eta^2\big)\|d^\star\|^2$ capturing (approximate) proxy-gradient error and curvature. Hence, the auxiliary objective cannot decrease by more than the sum of these two terms in one update.

Eq. 22 separates the primary change into an *improvement* term $+\eta\,(\mathbf{w}^{\star\top}\mathbf{h}_p) = \eta\langle \mathbf{g}_p, d^\star\rangle$, which is the proxy-level first-order gain along the chosen update direction, minus the same *degradation* term $\big(\eta\delta_p + \frac{L_p}{2}\eta^2\big)\|d^\star\|^2$ from proxy mismatch and curvature. Thus, Eq. 22 provides a sufficient condition for primary improvement: the first-order primary gain must dominate these error terms. Importantly, the LP in Eq. 17 prioritizes primary alignment within the auxiliary-compatible feasible region, rather than imposing monotonic primary improvement as a hard constraint.

Findings: Theorem 1 turns the proxy-level non-harm constraints ($\mathbf{w}^\top\mathbf{h}_a \ge -\epsilon$) into an *objective-level* guarantee: auxiliary reward objectives cannot deteriorate beyond a controlled amount ($\eta\epsilon + O(\eta)\|d^\star\|^2$), while the primary improvement is lower-bounded up to the same second-order and proxy-error terms.

E.2 Dual interpretation: adaptive penalties for conflicts

Theorem 2 (Dual Certificate and Adaptive-penalty View)**.** *Consider the primal LP Eq. 17. Its Lagrangian dual is*

$$\min_{\lambda\ge 0,\ \nu\in\mathbb{R}} \quad \nu + \sum_{a\in\mathcal{A}} \lambda_a\,\epsilon \quad \textit{s.t.} \quad \nu\,\mathbf{1} \;\ge\; h_p + \sum_{a\in\mathcal{A}} \lambda_a\,h_a. \tag{23}$$

Let $(\lambda^\star, \nu^\star)$ be an optimal dual solution. Then any primal optimum $w^\star$ satisfies

$$w^\star \in \arg\max_{w\in\Delta}\; w^\top\Big(h_p + \sum_{a\in\mathcal{A}} \lambda_a^\star h_a\Big), \tag{24}$$

together with complementary slackness:

$$\lambda_a^\star\,\big(w^{\star\top}h_a + \epsilon\big) = 0, \qquad \forall a \in \mathcal{A}. \tag{25}$$

Proof. Write Eq. 17 as $\max_{w\in\Delta}\ w^\top h_p$ s.t. $-(w^\top h_a+\epsilon)\le 0$ for each $a\in\mathcal{A}$. The Lagrangian is

$$\mathcal{L}(w,\lambda,\nu)=w^\top h_p+\sum_{a\in\mathcal{A}}\lambda_a\,(w^\top h_a+\epsilon)+\nu\,(1-\mathbf{1}^\top w),\quad \lambda\ge 0.$$

Rearrange:

$$\mathcal{L}(w,\lambda,\nu)=w^\top\Big(h_p+\sum_a\lambda_a h_a-\nu\mathbf{1}\Big)+\nu+\sum_a\lambda_a\epsilon.$$

Taking $\sup_{w\ge 0}$ with no upper bound yields finiteness iff $\nu\mathbf{1}\ge h_p+\sum_a\lambda_a h_a$ elementwise, in which case $\sup_{w\in\Delta}\mathcal{L}=\nu+\sum_a\lambda_a\epsilon$. This gives the dual Eq. 23. Strong duality holds since the primal is feasible and bounded. Karush-Kuhn-Tucker conditions (KKT) conditions yield Eq. 24 and complementary slackness Eq. 25. □

Findings: Theorem 2 provides a dual interpretation in which each active auxiliary compatibility constraint contributes an adaptive correction to the primary-alignment score through its multiplier $\lambda_a^\star$. By complementary slackness, $\lambda_a^\star>0$ implies that the corresponding auxiliary constraint is tight. When all auxiliary constraints are inactive, the dual correction vanishes, and Eq. 24 reduces to maximizing the primary-alignment score $\mathbf{w}^\top\mathbf{h}_p$ over the simplex. Importantly, this does not necessarily imply $\mathbf{w}=\mathbf{e}_p$; an auxiliary gradient may still receive weight when it has a strong positive projection onto the primary gradient.

Corollary 1 (Dual multipliers identify binding auxiliary constraints). If, for some auxiliary a, the compatibility constraint is slack, $\mathbf{w}^{\star\top}\mathbf{h}_a>-\epsilon$, then complementary slackness implies $\lambda_a^\star=0$. Thus, auxiliary a contributes no dual correction through its compatibility constraint in Eq. 24, although its gradient may still receive nonzero weight through the primary-alignment score $\mathbf{h}_p$. Conversely, $\lambda_a^\star>0$ implies that the corresponding constraint is tight, $\mathbf{w}^{\star\top}\mathbf{h}_a=-\epsilon$, so it acts as an active compatibility constraint on the primary-aligned solution.

F Rationale Evaluation Details

In this section, we explain how rationale quality is evaluated, following the setup introduced in Section 4.2. Each generated rationale is first assessed by a frontier LLM (i.e., `gpt-4o-series`) using a fixed scoring rubric designed to measure coherence. To keep the evaluation independent, this scoring model is not the same model that was used to produce the rationales during dataset construction. We further validated the automatic scores by asking human annotators to rate a subset of the same samples under the identical rubric. Finally, we compared the LLM-based and human judgments through statistical analysis and observed a significant positive correlation, which indicates that the automatic rationale evaluation is reasonably consistent with human assessment.

Rationale Evaluation Prompt We measure the plausibility of generated recommendations by asking a state-of-the-art LLM to rate their accompanying rationales on a four-point scale using the prompt shown below.

Prompt for the Rationales Evaluation

#Instruction
Please act as a neutral evaluator and assess the AI assistant's recommendation and its rationales based on the user's purchase history, the predicted item, and the ground truth item. Use the following four-level scale to assign your rating:
RATING-0: Irrelevant rationale — The recommendation rationale is largely unrelated to the user's history and provides little or no meaningful connection to the user's observed preferences.
RATING-1: Weak rationale — The recommendation provides a vague, generic, off-topic, or partially hallucinated explanation with limited alignment to the user's history.
RATING-2: Reasonable rationale — The recommendation is plausibly aligned with the user's history and provides a logically coherent explanation, but the rationale may lack depth, specificity, or persuasive detail.
RATING-3: Strong rationale — The recommendation is well aligned with the user's history and provides a clear, specific, and insightful explanation that connects the user's behavior patterns to the suggested item.
Please submit your evaluation using the format <eval>RATING-n </eval>, for instance: <eval>RATING-2 </eval>. Avoid letting the explanation length affect your judgment. Aim for an objective assessment.
#Known Information
[Purchase history] (1) Title: Fansing costume jewelry fathers day gift punk screw stud earrings (2) Title: Iblue jewelry stainless steel 3 pairs mens black gold silver stud earrings set,...(truncated)
[Recommended Item] Jstyle stainless steel mens womens stud earring hoop earrings for men 7 pairs
[Ground-Truth Item] Jstyle stainless steel mens womens stud earring hoop earrings for men 7 pairs
[Rationale] The customer previously purchased two sets of stainless steel stud earrings - one with a punk aesthetic and... (truncated)

Agreement Analysis between LLM and Human Ratings To assess whether the LLM-based rationale evaluation reflects human judgment, we conducted an additional expert validation study on approximately 250 samples drawn from the 1,500 instances used in Section 4.2. Two domain experts independently annotated these samples, and we measured their alignment with the LLM-derived scores using Quadratic Weighted Kappa (QWK = 0.72, $p < 0.0001$) and Spearman's rank correlation (ρ = 0.67, $p < 0.0001$). The results show a strong correspondence between expert and LLM evaluations: QWK indicates robust ordinal agreement while assigning larger penalties to more severe score discrepancies, and Spearman's ρ confirms that the relative quality ordering is largely preserved. Overall, this validation suggests that the LLM-based scoring provides a reliable proxy for human evaluation, capturing both score-level agreement and ranking consistency.

G Online A/B Test Details

We validated our approach in a real-world production deployment and ran an online A/B test during December 2025–February 2026. In the live setting, each user was shown a single (top-1) recommended item together with a short rationale explaining the suggestion. We served a customized variant of our model that satisfies product and business constraints. We compared against three baselines: (i) a popularity-based policy, (ii) a conventional machine-learning (ML) recommender, and (iii) a CF-based sequential recommender (SASRec). Traffic was assigned at the user level using a stable hash-based randomization into four buckets (PoP, ML, SASRec, Ours).

Table 6: Statistical Significance Tests (Two-Proportion z-test, One-sided)

Comparison	Δ CTR	z-statistic	p-value	95% CI for Δ
Ours vs **PoP**	+4.24pp	86.06	< 0.001	[+4.160%, +4.328%]
Ours vs **ML model**	+0.16pp	3.22	6.32×10^{-4}	[+0.063%, +0.258%]
Ours vs **SASRec**	+3.70pp	90.76	< 0.001	[+3.624%, +3.779%]

Tables 3 and 6 report the main online results. Our method achieves a CTR of 7.32%, outperforming all three baselines. Relative to the strongest baseline (ML model, 7.16%), we observe an absolute gain of +0.16 percentage points, corresponding to a +2.2% relative lift. Although the margin is modest in absolute terms, a one-sided two-proportion z-test confirms that the difference is statistically significant ($z \approx 3.22$, $p < 0.001$), with a 95% confidence interval of approximately $[+0.063\%\text{p}, +0.258\%\text{p}]$ that does not contain zero. The gains over the popularity-based model and SASRec are substantially larger: +4.24%p ($z \approx 86.06$, $p < 0.001$, 95% CI $= [+4.160\%\text{p}, +4.328\%\text{p}]$) and +3.70%p ($z \approx 90.76$, $p < 0.001$, 95% CI $= [+3.624\%\text{p}, +3.779\%\text{p}]$), respectively. All three comparisons are significant at the $\alpha = 0.05$ level. Importantly, these improvements are obtained while maintaining sub-150 ms inference latency ($\sim$138 ms/call), supporting practical deployability. The online experiment was conducted with randomized user-sticky bucket assignment, and all variants were served concurrently under identical eligibility and placement conditions. While traffic allocation was intentionally unbalanced for risk control, randomization ensures that CTR comparisons remain unbiased, with unequal sample sizes primarily affecting statistical power rather than estimate validity.

H Limitations

G^{2}RPO is evaluated on multiple public recommendation benchmarks and a production A/B test, but its performance may depend on the choice and quality of reward functions, candidate construction, and deployment domain. While our method adds only a small optimization overhead relative to LLM generation, scaling to substantially larger models or many more reward dimensions may require further engineering. Future work will study more efficient extensions to richer reward functions, optimization under multiple primary rewards, and scalable inference and training with larger candidate sets.

Algorithm 1 G^{2}RPO Training Step with Alignment-Aware Dynamic Weighting

Require: Policy model π_θ, frozen reference model π_{ref}, prompt batch $\{q^i\}_{i=1}^{B}$, number of rollouts G, reward functions $\{r_k\}_{k=1}^{K}$, primary reward index p, tolerance ϵ
Ensure: Updated policy parameters θ

1: Sample G responses for each prompt:

$$o^{(i,j)} \sim \pi_\theta(\cdot \mid q^i), \qquad i = 1, \ldots, B, \quad j = 1, \ldots, G.$$

2: Compute the format mask $m^{(i,j)} \in \{0, 1\}$ indicating whether $o^{(i,j)}$ satisfies the required output format.
3: **for** $k = 1, \ldots, K$ **do**
4: Compute reward values:

$$r_k^{(i,j)} = r_k(o^{(i,j)}).$$

5: Compute group-normalized advantages within each prompt group:

$$A_k^{(i,j)} = \frac{r_k^{(i,j)} - \text{mean}_{j'=1}^{G}\big(r_k^{(i,j')}\big)}{\text{std}_{j'=1}^{G}\big(r_k^{(i,j')}\big) + \varepsilon}.$$

6: Apply the format gate:

$$A_k^{(i,j)} \leftarrow m^{(i,j)} A_k^{(i,j)}.$$

7: Define the reward-specific surrogate loss:

$$\mathcal{L}_k(\theta) = -\frac{1}{BG} \sum_{i=1}^{B} \sum_{j=1}^{G} A_k^{(i,j)} \log \pi_\theta(o^{(i,j)} \mid q^i).$$

8: Compute the reward-improving update-direction proxy over LoRA trainable parameters:

$$d_k = -\nabla_{\theta_{\text{LoRA}}} \mathcal{L}_k(\theta) = \frac{1}{BG} \sum_{i=1}^{B} \sum_{j=1}^{G} A_k^{(i,j)} \nabla_{\theta_{\text{LoRA}}} \log \pi_\theta(o^{(i,j)} \mid q^i).$$

9: **end for**
10: Construct the update-direction Gram matrix:

$$M_{k\ell} = \langle d_k, d_\ell \rangle, \qquad k, \ell = 1, \ldots, K.$$

11: Solve the alignment-aware weighting problem:

$$w^\star = \arg\max_{w \in \Delta_K} \sum_{k=1}^{K} w_k M_{kp} \quad \text{s.t.} \quad \sum_{k=1}^{K} w_k M_{ka} \geq -\epsilon, \quad \forall a \neq p,$$

where $\Delta_K = \{w \in \mathbb{R}_{\geq 0}^K : \sum_{k=1}^{K} w_k = 1\}$.
12: If the LP is infeasible or $g_p = 0$, set $w^\star \leftarrow e_p$ (accuracy only).
13: Detach $w^\star$ from the computation graph.
14: Aggregate the advantages:

$$A_{\text{G}^2\text{RPO}}^{(i,j)} = \sum_{k=1}^{K} w_k^\star A_k^{(i,j)}.$$

15: Apply batch-wide standardization over all BG rollouts and reapply the format gate:

$$A_{\text{G}^2\text{RPO}}^{(i,j)} \leftarrow m^{(i,j)} \frac{A_{\text{G}^2\text{RPO}}^{(i,j)} - \text{mean}_{i',j'}\big(A_{\text{G}^2\text{RPO}}^{(i',j')}\big)}{\text{std}_{i',j'}\big(A_{\text{G}^2\text{RPO}}^{(i',j')}\big) + \varepsilon}.$$

16: Update θ_{LoRA} by minimizing the final policy-gradient loss:

$$\mathcal{L}_{\text{G}^2\text{RPO}}(\theta) = -\frac{1}{BG} \sum_{i=1}^{B} \sum_{j=1}^{G} A_{\text{G}^2\text{RPO}}^{(i,j)} \log \pi_\theta(o^{(i,j)} \mid q^i).$$

17: Take one optimizer step on θ_{LoRA}.