GPDPO: LEARNING FROM REWARD CONFLICTS VIA GROUP PARETO-DOMINANCE POLICY OPTIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Multi-reward reinforcement learning offers a promising framework for aligning large language models (LLMs) with diverse objectives, but conflicting reward signals can induce ambiguous update directions and hinder stable optimization. Existing approaches address such conflicts through reward reweighting, gradient coordination, or rollout filtering, yet they primarily operate on per-response or group-level optimization signals and underexploit relational information among sampled responses. As a result, conflict-filtered rollouts may still contain useful supervision that is discarded during training. We propose Group Pareto-Dominance Policy Optimization (GPDPO), a multi-reward policy optimization method that recovers pairwise supervision from conflict-filtered rollouts. GPDPO identifies Pareto-dominance relations among sampled responses and converts them into auxiliary preference signals, which are jointly optimized with the original policy objective. This enables more effective reuse of existing rollouts without additional sampling or reward evaluation. Experiments on API-Bank across multiple model scales and reward configurations show that GPDPO consistently outperforms competitive multi-reward optimization baselines. Further analysis demonstrates that GPDPO reduces both group-relative and pairwise reward conflicts, supporting the effectiveness of exploiting relational supervision for multi-reward alignment. Code will be released.

1 INTRODUCTION

Reinforcement learning provides an effective framework for improving large language models (LLMs) by optimizing task-specific rewards (Shao et al., 2024; Liu et al., 2026b). Group-based policy optimization methods such as GRPO (Shao et al., 2024) construct policy updates from relative rewards among sampled responses. Extending this paradigm to multiple reward signals enables LLMs to jointly optimize diverse alignment objectives, such as task correctness, output format, and response length (Liu et al., 2026b;a). However, multi-reward optimization introduces a distinct challenge: different reward dimensions may provide conflicting signals for the same response.

Existing methods address such conflicts from several perspectives, including reward reweighting (Yuan et al., 2026), gradient coordination (Li et al., 2025a), reward-wise advantage normalization (Liu et al., 2026b), and conflict-aware rollout filtering (Liu et al., 2026a). These approaches improve the compatibility of multiple reward signals by modifying how rewards, advantages, gradients, or samples contribute to policy optimization. In particular, conflict-aware filtering removes responses whose reward-wise advantages disagree in sign, preventing ambiguous signals from directly entering the policy update. While effective for conflict removal, this response-level filtering is coarse: once a response is deemed conflicting, all supervision associated with it is discarded. This raises an important question: if a response is excluded from the policy update due to conflict-filtering, does that necessarily mean it contains no useful information for learning?

Our observation shows that it does not. A response with conflicting group-relative advantages may still form a reliable ordering relative to another. In particular, one response may consistently dominate another across all reward dimensions, yielding an unambiguous Pareto-consistent preference even when the conflicting response itself is unsuitable for a policy update. As illustrated in Figure 1, this reveals a distinction between individual update and pairwise comparison: a response can be unreliable as a direct optimization signal while remaining informative as relational supervision.

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

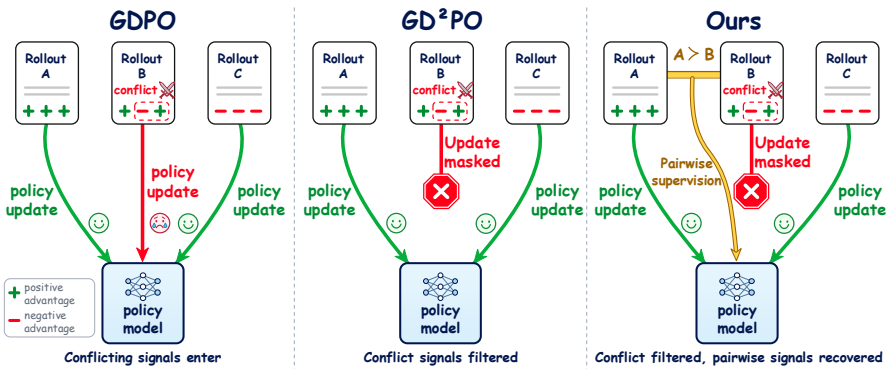


Figure 1: Different strategies for utilizing conflicting responses in multi-reward policy optimization. GDPO retains conflicting responses in policy updates, whereas GD^{2}PO masks them to avoid ambiguous optimization signals. GPDPO preserves conflict-aware masking while recovering pairwise supervision from Pareto-dominance relations, i.e., one response is no worse than another on all reward dimensions and strictly better on at least one.

Motivated by this, we introduce GPDPO, a multi-reward policy optimization framework that preserves conflict-aware masking for direct updates while recovering pairwise supervision from Pareto-dominance relations involving filtered rollouts. Unlike prior approaches that primarily modify per-response optimization signals, such as rewards, advantages, or gradients (Yuan et al., 2026; Li et al., 2025a; Liu et al., 2026b;a), GPDPO explicitly exploits the relational structure among sampled responses. Specifically, GPDPO identifies Pareto-dominance relations among rollouts and transforms these relations involving filtered responses into a pairwise preference loss, which encourages the policy to favor responses with consistent improvements across reward dimensions. By leveraging such supervision from filtered rollouts, GPDPO extracts additional training signals from existing samples without extra sampling or reward evaluation. This enables the policy to suppress ambiguous update directions while retaining informative pairwise preferences that would otherwise be discarded. Extensive experiments on API-Bank (Li et al., 2023), across two Qwen3 model scales (Yang et al., 2025) and two reward configurations, show that GPDPO consistently outperforms competitive baselines in multi-reward optimization.

Our contributions are summarized as follows: 1) We show that responses with conflicting group-relative advantages can still yield reliable pairwise preferences, revealing supervision overlooked by response-level conflict filtering. 2) We propose GPDPO, which preserves conflict-aware masking for direct updates while recovering pairwise supervision from filtered responses through Pareto-dominance relations. 3) Extensive experiments on API-Bank across varied model scales and reward configurations show that GPDPO consistently outperforms competitive multi-reward baselines.

2 Related Work

Reinforcement learning with verifiable rewards. RLVR has shown strong potential for improving LLM reasoning (Shao et al., 2024; Guo et al., 2025). Prior work advances this paradigm through group-relative advantage estimation (Shao et al., 2024), sampling and objective refinement (Yu et al., 2026; Liu et al., 2025b), and sequence-level policy optimization (Zheng et al., 2025). These methods improve the efficiency and stability of reasoning optimization. However, applying GRPO-style algorithms to multiple rewards through scalar aggregation may obscure reward-specific distinctions, limiting their effectiveness in multi-reward optimization (Liu et al., 2026b).

Multi-reward policy optimization. Prior work coordinates multiple rewards through fixed or adaptive weighting (Zhou et al., 2024; Lu et al., 2026), gradient balancing (Li et al., 2025a), and reward-wise or blockwise advantage estimation (Liu et al., 2026b; Pavlenko et al., 2026). GD^{2}PO further filters rollouts with conflicting reward-wise advantages (Liu et al., 2026a). These methods mitigate multi-reward conflicts by adjusting how reward signals contribute to policy updates. However, response-level conflict handling may overlook consistent inter-response relations and discard useful supervision. GPDPO instead complements group-relative conflict handling with pairwise comparisons, recovering reward-consistent supervision from filtered rollouts.

Pareto dominance relations. Pareto dominance is widely used in multi-objective search, including NSGA-II (Emmerich & Deutz, 2018; Deb et al., 2002). In LLM alignment, related methods use Pareto or reward-consistency criteria to assign trajectory advantages (Li et al., 2026), construct preference pairs (Li et al., 2025b; Xu et al., 2026), or downweight conflicting pairs (Xie et al., 2026). These methods use Pareto dominance to guide multi-objective learning. Insipred by these, GPDPO uses Pareto dominance to identify consistent comparison structures among rollouts and draws on these structures to provide supplementary supervision for multi-reward optimization.

3 METHOD

In this section, we first introduce multi-reward optimization and the associated conflict-filtering mechanism (Section 3.1). We then revisit the sign-based filtering criterion, identify two potential limitations, and empirically examine their effects (Section 3.2). Building on these findings, we propose GPDPO, which exploits Pareto-dominance relations between responses to recover fine-grained pairwise supervision for improved conflict mitigation (Section 3.3).

3.1 PRELIMINARIES

Multi-reward group-relative optimization. Given a prompt x, the sampling policy $\pi_{\theta_{\text{old}}}$ generates a group of G responses, denoted by $\mathcal{G}(x) = \{y_1, \ldots, y_G\}$. Each response y_n receives an M-dimensional reward vector $\mathbf{r}_n = (r_n^1, \ldots, r_n^M)$, where $r_n^i = r_i(x, y_n)$ and higher values are preferred on every reward dimension. A straightforward way to extend GRPO (Shao et al., 2024) to the multi-reward setting is to aggregate the rewards before applying group-relative normalization:

$$R_n = \sum_{i=1}^{M} w_i r_n^i, \qquad \overline{R} = \frac{1}{G}\sum_{n=1}^{G} R_n, \qquad A_n^{\text{GRPO}} = \frac{R_n - \overline{R}}{s_R}. \tag{1}$$

Here, w_i is the weight of reward dimension i, and $s_R > 0$ is the within-group normalization scale. The resulting scalar advantage is used in the clipped policy objective (Schulman et al., 2017) to update the probability of the corresponding response. GDPO (Liu et al., 2026b) observes that aggregating rewards too early can lose information specific to individual reward dimensions. It therefore normalizes each reward separately, aggregates the resulting advantages, and applies batch-level scale normalization. GD^{2}PO (Liu et al., 2026a) further addresses conflicts among these advantages: when a response has both positive and negative advantages across reward dimensions, direct aggregation can cause the learning signals to cancel each other out. GD^{2}PO therefore filters responses with conflicting advantage signs and adjusts the update through reweighting. Specifically, the reward-wise advantages and filtering gate are defined as

$$A_n^i = \frac{r_n^i - \mu_i}{s_i}, \qquad \mu_i = \frac{1}{G}\sum_{j=1}^{G} r_j^i, \qquad g(y_n) = \mathbf{1}\big\{\operatorname{sign}(A_n^i)\operatorname{sign}(A_n^k) \geq 0,\ \forall i, k\big\}. \tag{2}$$

Here, s_i denotes the normalization scale for reward dimension i. The binary gate $g(y_n)$ equals 1 for retained responses and 0 for filtered responses. Specifically, the gate selects responses whose reward-wise advantages share a consistent sign across dimensions, while removing the contribution of responses with conflicting directions.

Pareto dominance. Pareto dominance compares two vectors in a multi-objective space. $\mathbf{a}$ Pareto-dominates $\mathbf{b}$ if $\mathbf{a}$ is at least as good as $\mathbf{b}$ in all dimensions and strictly better in at least one dimension (Emmerich & Deutz, 2018):

$$\mathbf{a} \succ_{\text{P}} \mathbf{b} \quad \Longleftrightarrow \quad \big(\forall i:\ a_i \geq b_i\big) \ \wedge\ \big(\exists i:\ a_i > b_i\big). \tag{3}$$

Pareto dominance does not allow gains on one objective to compensate for losses on another. Thus, when two vectors are better on different objectives, neither dominates the other.

3.2 RETHINKING CONFLICT FILTERING IN MULTI-REWARD RL

Here, we revisit conflict filtering in multi-reward RL from the perspective of group-relative advantages. Our analysis reveals two limitations: filtering decisions are sensitive to the reference group,

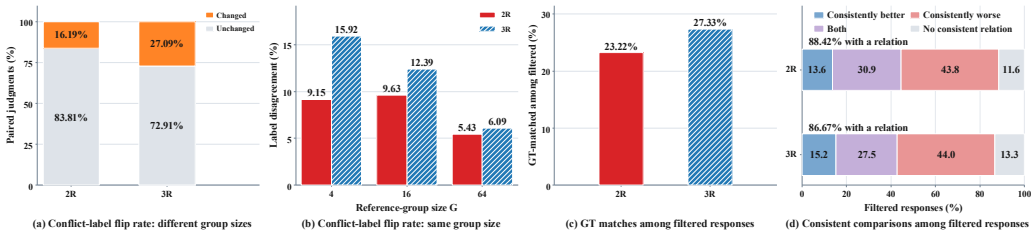


Figure 2: Conflict filtering is sensitive to the reference group and can discard useful learning signals. (a,b) Conflict-label flip rates for fixed responses under different group sizes and different peers at the same group size, respectively. (c) Percentage of filtered responses that match the ground-truth. (d) Consistent comparison relations between filtered responses and their peers. "Both" denotes being consistently better than some peers and consistently worse than others.

and responses removed due to group-wise conflicts may still contain useful pairwise supervision. Through empirical studies, we show that reward conflicts at the group level do not necessarily imply the absence of consistent response-level comparisons.

Prior work (Liu et al., 2026a) filters responses whose advantages disagree in sign across reward dimensions, which can be rewritten as follows:

$$\operatorname{sign}(A_n^i) = \operatorname{sign}(r_n^i - \mu_i) = \operatorname{sign}\left[\sum_{j=1}^{G}(r_n^i - r_j^i)\right]. \tag{4}$$

This formulation shows that the sign of a response's advantage depends on the sum of its reward differences from the other responses in the group. Conflict filtering based on these group-level signs has two limitations. First, **conflict filtering is sensitive to the reference group**: the same response can receive different filtering decisions as its peers change. Moreover, **group-wise conflict filtering can discard useful learning signals**. Coarse-grained conflict signals obtained by summing within-group reward differences cannot capture fine-grained comparisons between responses, leaving useful pairwise training signals unrecognized by the filter. To further analyze these limitations, we conduct the empirical experiments shown in Figure 2. Detailed experiments are provided in the appendix.

Conflict filtering is sensitive to the rollout group. We hold the target responses fixed, vary the reference group size, and examine whether the filtering decisions change. As shown in Figure 2(a), expanding the original group from $G = 4$ to $G = 64$ changes the conflict filtering decisions for 16.19% and 27.09% of the fixed targets under 2R and 3R, respectively. Furthermore, we keep the group size fixed and replace the peer responses. Figure 2(b) shows decision inconsistency rates of 9.15% and 15.92% at $G = 4$, which remain at 5.43% and 6.09% at $G = 64$. These results show that conflict filtering signals are highly sensitive to the reference group.

Group-wise conflict filtering can discard useful learning signals. We further analyze responses filtered due to reward conflicts. As shown in Figure 2(c), 23.22% and 27.33% of these responses under 2R and 3R, respectively, exactly match the ground-truth (GT) ordered tool-call structure. This shows that filtered responses can still contain valuable training signals. We then investigate the fine-grained information in these responses. As shown in Figure 2(d), 88.42% and 86.67% of filtered responses under 2R and 3R, respectively, have a consistent comparison relation with at least one other response in the group, yet these relations are discarded by group-wise conflict filtering. These findings show that group-wise conflict signals lose substantial fine-grained information, causing the filter to incorrectly discard many useful training samples and signals.

These observations lead to our motivation. First, conflict filtering signals should be decoupled from the group mean to improve training stability. Moreover, conflict detection should go beyond group-level relations to exploit comparisons between individual rollouts, which depend only on the properties of the two rollouts themselves. Finally, even when a response has conflicting advantages relative to the group mean, pairwise comparisons with individual rollouts can still reveal consistent preference relations and provide useful supervision.

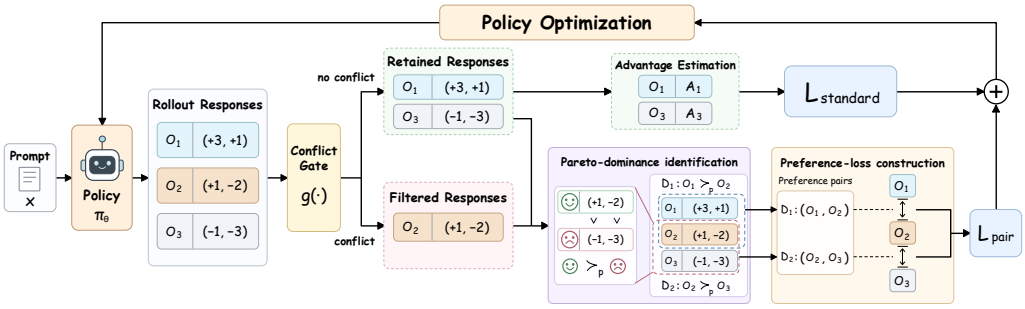


Figure 3: Overview of GPDPO. Based on conflict filtering, we identify Pareto dominance relations within filtered responses and leverage them to construct preference loss. The resulting preference loss is combined with the standard multi-reward policy-gradient objective for policy optimization.

3.3 GPDPO: Group Pareto-Dominance Policy Optimization

In this section, we introduce GPDPO, a multi-reward policy optimization method that recovers pairwise supervision from conflict-filtered rollouts through Pareto dominance. Rather than relying solely on group-level conflict handling, GPDPO exploits consistent preference relations between individual responses and incorporates the resulting pairwise supervision into the original policy objective. As illustrated in Figure 3, GPDPO consists of three components: (1) Pareto-dominance identification, (2) preference-loss construction, and (3) joint policy optimization.

Pareto-dominance identification. Based on the conflict filtering mechanism of prior work (Liu et al., 2026a), we obtain the filtered set $\mathcal{F}(x) = \{y_n \in \mathcal{G}(x) : c(y_n) = 1\}$ and retained set $\mathcal{R}(x) = \mathcal{G}(x) \setminus \mathcal{F}(x)$ from the sampled group $\mathcal{G}(x)$. Here, $c(y_n) = 1 - g(y_n) = 1$ indicates that the response is filtered due to conflict. We define a consistent preference relation based on Pareto dominance (Emmerich & Deutz, 2018). Following Section 3.1, for two responses y_a and y_b to the same prompt x, we say that y_a Pareto-dominates y_b, if and only if $\mathbf{r}_a \succ_{\mathrm{P}} \mathbf{r}_b$. For notational simplicity, we use $y_a \succ_{\mathrm{P}} y_b$ to denote this relation between responses y_a and y_b.

Pareto dominance depends only on the relative reward vectors of the two responses, independent of the group mean or other rollouts. This pairwise criterion therefore exposes reward-consistent, conflict-free comparison structures that can remain hidden under group-level aggregation.

We collect all Pareto dominance relations within the sampled group into the following set:

$$\mathcal{D}(x) = \left\{(y_w, y_l) \in \mathcal{G}(x)^2 : y_w \succ_{\mathrm{P}} y_l\right\}. \tag{5}$$

Here, y_w represents the dominating response and y_l represents the dominated response.

Preference-loss construction. To convert the dominance set into optimization signals, we construct a preference-pair set by selecting dominance edges in $\mathcal{D}(x)$ involving at least one response in the filtered set $\mathcal{F}(x)$:

$$\mathcal{E}(x) = \left\{(y_w, y_l) \in \mathcal{D}(x) : y_w \in \mathcal{F}(x) \ \vee \ y_l \in \mathcal{F}(x)\right\}. \tag{6}$$

This selection focuses the auxiliary objective on supervision recovered from filtered rollouts, while retaining the original policy optimization pathway for non-filtered responses.

We then define the mean token-level log-probability ratio to measure the policy update of a response relative to the sampling policy:

$$u_\theta(y \mid x) = \frac{1}{|y|} \sum_{t=1}^{|y|} \log \frac{\pi_\theta(y_t \mid x, y_{<t})}{\pi_{\theta_{\mathrm{old}}}(y_t \mid x, y_{<t})}. \tag{7}$$

Here, $|y|$ denotes the number of tokens in the response, and $\pi_{\theta_{\mathrm{old}}}$ is the sampling policy used to generate the rollout group. The score $u_\theta(y \mid x)$ captures the relative change in the response likelihood under the updated policy.

To ensure that the auxiliary objective updates only on filtered responses, we stop gradients through retained responses when they participate in selected preference pairs:

$$\widetilde{u}_\theta(y \mid x) = \begin{cases} u_\theta(y \mid x), & y \in \mathcal{F}(x), \\ \mathrm{sg}[u_\theta(y \mid x)], & y \notin \mathcal{F}(x), \end{cases} \tag{8}$$

where sg denotes the stop-gradient operator. Retained responses serve as fixed comparison anchors, whereas filtered responses receive the preference supervision. When both endpoints of a preference pair are filtered, both responses contribute gradients to the auxiliary update.

For each selected dominance relation, we optimize the following pairwise preference objective:

$$\ell_{\text{pair}}(y_w, y_l \mid x) = -\log \sigma(\beta\,[\widetilde{u}_\theta(y_w \mid x) - \widetilde{u}_\theta(y_l \mid x)])\,, \tag{9}$$

where y_w and y_l denote the dominating and dominated responses, respectively. This objective encourages the policy to increase the relative likelihood of responses consistent with Pareto dominance, thereby incorporating reward-consistent supervision into policy optimization. Here, $\sigma(\cdot)$ is the sigmoid function and $\beta > 0$ controls the preference margin.

Joint policy optimization. For each training batch $\mathcal{B}$, we assign equal weight to all selected preference pairs and define the auxiliary objective as:

$$\mathcal{L}_{\text{pair}} = \frac{\sum_{x\in\mathcal{B}} \sum_{(y_w,y_l)\in\mathcal{E}(x)} \ell_{\text{pair}}(y_w, y_l \mid x)}{\sum_{x\in\mathcal{B}} |\mathcal{E}(x)|}. \tag{10}$$

We finally formulate the GPDPO objective by combining the original training loss with the auxiliary pairwise objective, weighted by λ:

$$\mathcal{L}_{\text{GPDPO}} = \mathcal{L}_{\text{standard}} + \lambda \mathcal{L}_{\text{pair}}, \tag{11}$$

where $\mathcal{L}_{\text{standard}}$ denotes the standard multi-reward policy-gradient objective used in GDPO (Liu et al., 2026b) and GD^{2}PO (Liu et al., 2026a), and λ controls the contribution of the pairwise loss.

Our GPDPO recovers conflict-related supervision at a finer granularity by operating on pairwise relations between rollouts. Responses filtered by group-level conflict criterion may still form reward-consistent comparisons with specific peers. GPDPO identifies and exploits these fine-grained relations, reusing supervision already present in the rollout group without additional sampling or reward evaluation. The complete algorithm is provided in the appendix.

4 Experiments

4.1 Experimental Settings

Task and rewards. We study multi-reward tool-call policy optimization on the 3,920 training examples from RLLA-4K (Qian et al., 2026). The upstream ToolRL training mixture draws on ToolACE (Liu et al., 2025a), Hammer (Masked) (Lin et al., 2024), and xLAM (Zhang et al., 2025). In the two-reward setting (2R), the reward consists of correctness and length; the three-reward setting (3R) additionally includes format. The correctness reward measures agreement between generated and annotated tool calls in tool names, argument names, and argument values, and is scaled to $[-3, 3]$. The length reward is defined as $\min(n_{\text{think}}/512, 1)$, where n_{think} denotes the number of words inside the `<think>` block; responses without a `<think>` block receive zero. The format reward is binary and indicates whether the response follows the prescribed output-tag structure. Unless otherwise stated, all reward dimensions are weighted equally.

Model training and evaluation. We evaluate Qwen3-1.7B and Qwen3-4B (Yang et al., 2025), comparing our GPDPO against GRPO (Shao et al., 2024), GDPO (Liu et al., 2026b), and GD^{2}PO (Liu et al., 2026a). GRPO aggregates reward dimensions before normalization, whereas GDPO normalizes each reward dimension separately before aggregation; GD^{2}PO further introduces conflict filtering and query-level reweighting. All methods are trained for 200 rollout steps with 512 prompts per step and $G = 4$ responses per prompt. We use one epoch per step, mini-batches of 128 prompts, and a learning rate of 10^{-6}. Responses are sampled with temperature 1.0, without a KL penalty or entropy bonus. The maximum prompt and response lengths are 2,048 and 1,024 tokens, respectively. Our implementation is built on verl (Sheng et al., 2025) and vLLM (Kwon et al., 2023). Unless otherwise specified, GPDPO uses uniform pair weighting with $\lambda = 0.1$ and $\beta = 0.1$, and refreshes the sampling-policy reference after each rollout step.

We evaluate on API-Bank (Li et al., 2023), which contains 597 questions across Levels 1/2/3, with 399/67/131 questions, respectively. Each row in the main results table corresponds to a single evaluation of the selected step-200 checkpoint using greedy decoding and a maximum generation length

Table 1: Results on API-Bank under the two-reward (2R) setting. Accuracy metrics are reported as percentages. The best and second-best results are highlighted in bold and underlined, respectively.

Method	Level 1 Acc.	Level 2 Acc.	Level 3 Acc.	Correct Acc.	Length Rew.	Overall
Qwen3-1.7B						
GRPO(Shao et al., 2024)	66.17	53.73	46.56	60.47	0.9643	1.569
GDPO (Liu et al., 2026b)	66.67	61.19	43.51	60.97	**0.9875**	1.597
GD^{2}PO (Liu et al., 2026a)	69.17	61.19	42.75	62.48	0.9793	1.604
GPDPO	**70.93**	**61.19**	**48.85**	**64.99**	0.9812	**1.631**
Qwen3-4B						
GRPO (Shao et al., 2024)	71.93	59.70	45.80	64.82	0.9898	1.638
GDPO (Liu et al., 2026b)	**72.93**	56.72	**47.33**	65.49	0.9978	1.653
GD^{2}PO (Liu et al., 2026a)	71.68	59.70	45.80	64.66	0.9987	1.645
GPDPO	**72.93**	**64.18**	45.80	**66.00**	**0.9995**	**1.659**

Table 2: Results on API-Bank under the three-reward (3R) setting. Accuracy metrics are reported as percentages. The best and second-best results are highlighted in bold and underlined, respectively.

Method	Level 1 Acc.	Level 2 Acc.	Level 3 Acc.	Correct Acc.	Format Acc.	Length Rew.	Overall
Qwen3-1.7B							
GRPO(Shao et al., 2024)	65.91	**62.69**	39.69	59.80	98.99	0.9519	2.540
GDPO (Liu et al., 2026b)	68.92	53.73	41.22	61.14	**100.00**	0.9608	2.572
GD^{2}PO (Liu et al., 2026a)	68.92	59.70	**44.27**	62.48	**100.00**	0.9756	2.600
GPDPO	**70.43**	61.19	42.75	**63.32**	99.66	**0.9760**	**2.606**
Qwen3-4B							
GRPO(Shao et al., 2024)	70.93	58.21	46.56	64.15	98.99	0.9874	2.619
GDPO (Liu et al., 2026b)	68.42	56.72	46.56	62.31	99.83	**0.9976**	2.619
GD^{2}PO (Liu et al., 2026a)	71.18	56.72	42.75	63.32	**100.00**	0.9967	2.630
GPDPO	**71.43**	**64.18**	**47.33**	**65.33**	**100.00**	0.9839	**2.637**

of 4,096 tokens. We report accuracy for each difficulty level, overall accuracy across all questions (CORRECT ACC.), format accuracy in the 3R setting, and mean length reward. Accuracy metrics are reported as percentages, while length reward remains on its original $[0, 1]$ scale. The OVERALL score is defined as CORRECT ACC.$/100$ + LENGTH REW. for 2R, and additionally includes FORMAT ACC.$/100$ for 3R. We treat CORRECT ACC. as the primary measure of task performance.

4.2 MAIN RESULTS

As shown in Tables 1 and 2, GPDPO attains the highest Correct Acc. and Overall score across all four combinations of model scale and reward setting. Under 2R, GPDPO improves Correct Acc. over GD^{2}PO from 62.48% to 64.99% on Qwen3-1.7B and from 64.66% to 66.00% on Qwen3-4B, corresponding to gains of 2.51 and 1.34 percentage points, respectively. The improvement is particularly pronounced on the more challenging Level 3 subset for Qwen3-1.7B, where accuracy increases from 42.75% to 48.85%, a gain of 6.10 points. These results demonstrate that GPDPO effectively extracts useful supervision from reward conflicts, leading to improvements in the primary task objective rather than gains driven solely by auxiliary rewards or the aggregate Overall score.

The advantage persists under the more demanding 3R setting. GPDPO achieves 63.32% Correct Acc. on Qwen3-1.7B and 65.33% on Qwen3-4B, outperforming GD^{2}PO by 0.84 and 2.01 percentage points, respectively. Importantly, format accuracy is already near saturation across methods, yet GPDPO still improves Correct Acc. and attains the highest Overall score at both model scales. This indicates that GPDPO does more than improve auxiliary objectives such as Format Acc. and Length Rew.; by recovering pairwise supervision from filtered responses, it extracts additional learning signal that directly benefits task correctness.

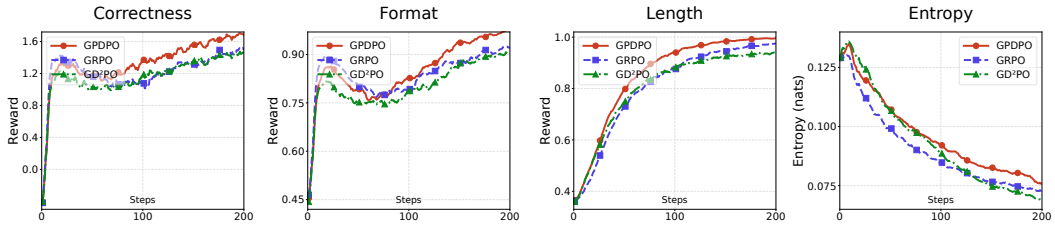


Figure 4: Training dynamics of Qwen3-1.7B under the three-reward (3R) setting.

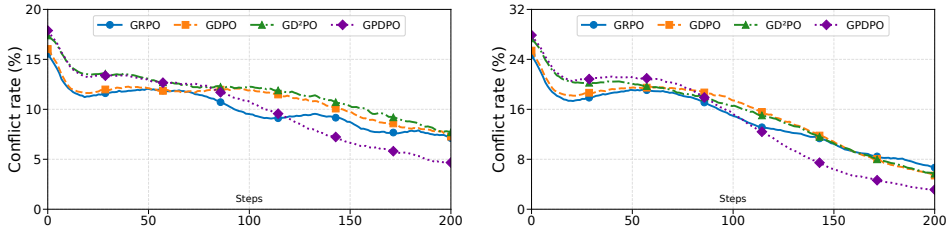


Figure 5: Conflict dynamics on Qwen3-1.7B under 3R setting. **Left:** pairwise preference conflict rate over all non-tied within-group response pairs, including retained–retained pairs. **Right:** GD^{2}PO conflict rate over valid rollouts.

4.3 Training and Conflict Dynamics

Beyond improving downstream performance, GPDPO exhibits the training dynamics: it reduces reward conflicts and achieves stronger reward optimization without accelerating entropy collapse.

Conflict metrics. We measure multi-reward conflict at two levels. Following GD^{2}PO, the *group-relative conflict rate* is the fraction of valid rollouts whose reward-wise advantages contain both positive and negative entries, with zeros treated as neutral. We further define the *pairwise preference conflict rate* as the fraction of Pareto-incomparable pairs among all non-tied, unordered response pairs within a group. These metrics capture conflict at the aggregate rollout level and the atomic pairwise level, respectively.

Training dynamics. As shown in Figure 4, GPDPO achieves stronger late-stage reward optimization while retaining higher policy entropy. Over steps 151–200, its mean correctness reward reaches 1.6260, compared with 1.4242 for GRPO and 1.4035 for GD^{2}PO. Its format and length rewards are also higher than GD^{2}PO (0.9566/0.9914 vs. 0.8902/0.9328), while maintaining higher entropy (0.0795 vs. 0.0746 for GRPO and 0.0719 for GD^{2}PO). Thus, the higher rewards do not coincide with faster policy collapse in this representative run.

Conflict dynamics. As shown in Figure 5, our method exhibits faster declines in conflict rates at both levels and achieves lower final conflict rates than GRPO, GDPO, and GD^{2}PO. This is consistent with GPDPO's design: reward-consistent Pareto relations recover auxiliary supervision from filtered responses, and each group-relative deviation can be decomposed into aggregated pairwise reward differences. The simultaneous reduction in both conflict measures suggests that recovering reward-consistent pairwise supervision is an effective strategy for alleviating multi-reward conflict.

4.4 Ablation and Sensitivity

Component analysis of GPDPO. To understand the contribution of each design choice in GPDPO, we conduct ablations on four variants: (1) GPDPO (Strict Pareto Dominance), which requires strict improvement on every reward dimension instead of allowing ties; (2) GPDPO (Scalarized Reward Ordering), which replaces Pareto-based comparisons with ordering based on an aggregated scalar reward; (3) GPDPO (Fixed Reference Model), which keeps the initial policy as the reference model throughout training; and (4) GPDPO (Pareto-Guided Advantage Adjustment), which incorporates recovered Pareto relations by modifying advantages instead of directly optimizing preference margins. All variants are evaluated on Qwen3-1.7B under 2R setting (Table 3).

Table 3: Ablations on Qwen3-1.7B with 2R setting. Best Correct Acc. and Length Rew. are bold.

Method	Level 1 Acc.	Level 2 Acc.	Level 3 Acc.	Correct Acc.	Length Rew.
GD^{2}PO	69.17	61.19	42.75	62.48	0.9793
GPDPO (Strict Pareto Dominance)	68.67	62.69	48.09	63.48	0.9650
GPDPO (Scalarized Reward Ordering)	68.42	49.25	48.09	61.81	0.9675
GPDPO (Fixed Reference Model.)	71.18	58.21	39.69	62.81	0.9767
GPDPO (Pareto-Guided Advantage Adjustment)	70.68	61.19	44.27	63.82	0.9762
GPDPO (Full)	70.93	61.19	48.85	**64.99**	**0.9812**

Strict Pareto dominance reduces supervision coverage. GPDPO with strict Pareto dominance achieves 63.48% Correct Acc., 1.51 percentage points below full GPDPO. Strict dominance requires improvement on every reward dimension and therefore discards comparisons with ties. Standard Pareto dominance retains such pairs when one response is no worse on all rewards and strictly better on at least one, recovering additional supervision without violating reward consistency. These results suggest that allowing ties under standard Pareto dominance recovers broader yet still reward-consistent supervision, leading to more effective policy optimization.

Scalarized ordering loses reward consistency. Replacing Pareto dominance with scalarized reward ordering reduces Correct Acc. from 64.99% to 61.81%, falling below GD^{2}PO's 62.48%. This degradation indicates that the benefit of recovered supervision comes from preserving consistent preference directions across reward dimensions rather than from additional pairwise signals alone.

Reference refreshing improves optimization. Replacing the refreshed reference policy with a fixed reference reduces Correct Acc. to 62.81%, compared with 64.99% for full GPDPO. The fixed-reference variant keeps the initial policy as the reference throughout training, with $\beta = 32$ and $\lambda = 0.0003125$ selected to match the zero-margin gradient scale of full GPDPO. This result suggests that refreshing the reference policy along with the evolving sampling policy provides a more effective optimization target under the tested configuration.

Recovered relations remain useful under alternative updates. GPDPO (Pareto-Guided Advantage Adjustment) improves Correct Acc. from GD^{2}PO's 62.48% to 63.82%, confirming that the recovered Pareto relations themselves contain useful optimization signals. Full GPDPO further improves performance to 64.99%, suggesting that directly optimizing pairwise preference margins provides a more effective way to exploit the recovered relational supervision.

Robustness to Auxiliary Supervision Strength. Figure 6 evaluates the sensitivity of GPDPO to the parameter λ. Across all tested values ($\lambda = 0.05$, 0.1, 0.15, and 0.2), GPDPO consistently outperforms GD^{2}PO, achieving Correct Acc. scores of 63.82%, 64.99%, 64.66%, and 63.15%, respectively, compared with 62.48% for GD^{2}PO. These results indicate that GPDPO maintains consistent improvements across a range of auxiliary supervision strengths rather than relying on a single carefully tuned coefficient.

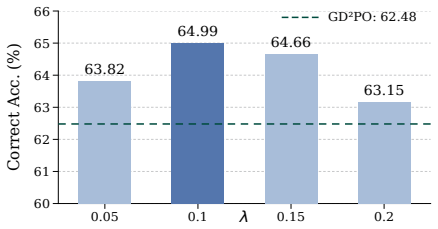


Figure 6: Sensitivity of λ on Qwen3-1.7B under 2R setting.

5 Conclusion

We presented GPDPO, a multi-reward policy optimization method that recovers reward-consistent pairwise supervision from responses discarded by conflict filtering. Our key observation is that conflicts in group-relative update signals do not necessarily eliminate reliable pairwise preferences between responses. GPDPO exploits Pareto-dominance relations within existing rollout groups to recover this supervision without additional sampling or reward evaluation. Experiments show that GPDPO consistently improves multi-reward optimization performance while reducing both group-relative and pairwise reward conflicts. These results demonstrate the value of fine-grained relational supervision for more effective multi-objective LLM alignment.

AI USE STATEMENT

We used generative AI tools solely to assist with manuscript writing and improve the clarity, grammar, and readability of the text. We did not use generative AI tools for any other aspect of the research. The authors reviewed and revised all AI-assisted text to ensure that it accurately reflected the methods and reported results. The authors take full responsibility for the final manuscript, including its accuracy, originality, and the integrity of its claims and conclusions.

REPRODUCIBILITY STATEMENT

To facilitate reproducibility, we describe the GPDPO formulation in Section 3 and provide detailed pseudocode and implementation notes in Appendix A.2. Section 4.1 introduces the datasets, base models, baselines, reward definitions, and evaluation metrics. Appendix A.1 documents the training and evaluation configurations, including hyperparameters, hardware, decoding settings, and checkpoint reporting conventions. Appendix C specifies the experimental setups, sampling procedures, and statistical definitions used in the conflict-filtering analysis in Figure 2. Our implementation builds on verl and vLLM, and we will release the source code soon.

REFERENCES

K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002. doi: 10.1109/4235.996017.

Michael TM Emmerich and André H Deutz. A tutorial on multiobjective optimization: fundamentals and evolutionary methods. *Natural computing*, 17(3):585–609, 2018.

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

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th symposium on operating systems principles*, pp. 611–626, 2023.

Chengao Li, Hanyu Zhang, Yunkun Xu, Hongyan Xue, Xiang Ao, and Qing He. Gradient-adaptive policy optimization: Towards multi-objective alignment of large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 11214–11232, 2025a.

Junyi Li, Xiaowei Qian, Yingyi Zhang, Wenlin Zhang, Guojing Li, Sheng Zhang, Xiao Han, Yichao Wang, and Xiangyu Zhao. Towards pareto-optimal tool-integrated agents with pareto ranking policy optimization. *arXiv preprint arXiv:2606.16111*, 2026.

Minghao Li, Yingxiu Zhao, Bowen Yu, Feifan Song, Hangyu Li, Haiyang Yu, Zhoujun Li, Fei Huang, and Yongbin Li. Api-bank: A comprehensive benchmark for tool-augmented llms. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pp. 3102–3116, 2023.

Moxin Li, Yuantao Zhang, Wenjie Wang, Wentao Shi, Zhuo Liu, Fuli Feng, and Tat-Seng Chua. Self-improvement towards pareto optimality: Mitigating preference conflicts in multi-objective alignment. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 11010–11031, 2025b.

Qiqiang Lin, Muning Wen, Qiuying Peng, Guanyu Nie, Junwei Liao, Jun Wang, Xiaoyun Mo, Jiamu Zhou, Cheng Cheng, Yin Zhao, et al. Hammer: Robust function-calling for on-device language models via function masking. *arXiv preprint arXiv:2410.04587*, 2024.

Haotian Liu, Yihao Liu, Jingwei Ni, Siyuan Huang, Xinpeng Liu, Pengyu Cheng, Jiajun Song, Ruijin Ding, Junfeng Li, Zhechao Yu, et al. GD^{2}PO: Mitigating multi-reward conflicts via group-dynamic reward-decoupled policy optimization. *arXiv preprint arXiv:2606.16771*, 2026a.

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*, 2026b.

Weiwen Liu, Xu Huang, Xingshan Zeng, Shuai Yu, Dexun Li, Shuai Wang, Weinan Gan, Zhengying Liu, Yuanqing Yu, Zezhong WANG, et al. Toolace: Winning the points of llm function calling. In *International conference on learning representations*, volume 2025, pp. 41359–41381, 2025a.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding r1-zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*, 2025b.

Yining Lu, Zilong Wang, Shiyang Li, Xin Liu, Changlong Yu, Qingyu Yin, Zhan Shi, Zixuan Zhang, and Meng Jiang. Learning to optimize multi-objective alignment through dynamic reward weighting. *Transactions of the Association for Computational Linguistics*, 14:1051–1073, 2026.

Kirill Pavlenko, Alexander Golubev, Simon Karasik, and Boris Yangel. Blockwise advantage estimation for multi-objective rl with verifiable rewards. *arXiv preprint arXiv:2602.10231*, 2026.

Cheng Qian, Emre Can Acikgoz, Qi He, Hongru Wang, Xiusi Chen, Dilek Hakkani-Tur, Gokhan Tur, and Heng Ji. Toolrl: Reward is all tool learning needs. In *Advances in Neural Information Processing Systems*, volume 38, pp. 105523–105553, 2026.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

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

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, pp. 1279–1297, 2025.

Hongyan Xie, Yikun Ban, Ruiyu Fang, Di Wu, Zixuan Huang, Deqing Wang, Jianxin Li, and Shuangyong Song. Direct preference optimization with pareto dominance constraint for online multi-objective alignment. *Neurocomputing*, pp. 134604, 2026.

Zhihao Xu, Yongqi Tong, Xin Zhang, Jun Zhou, and Xiting Wang. Understanding conflicts in multi-objective alignment through reward consistency. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 5450–5472, 2026.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

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

Yu Yuan, Yaoyou Fan, Lili Zhao, Guangting Zheng, Kai Zhang, Lu Pan, Ke Zeng, and Qi Liu. Uncovering and mitigating aggregation-induced reward hacking in multi-reward reinforcement learning. *arXiv preprint arXiv:2609.00213*, 2026.

Jianguo Zhang, Tian Lan, Ming Zhu, Zuxin Liu, Thai Hoang, Shirley Kokane, Weiran Yao, Juntao Tan, Zhiwei Liu, Yihao Feng, et al. xlam: A family of large action models to empower ai agent systems. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 11583–11597, 2025.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, Jingren Zhou, and Junyang Lin. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*, 2025.

Zhanhui Zhou, Jie Liu, Jing Shao, Xiangyu Yue, Chao Yang, Wanli Ouyang, and Yu Qiao. Beyond one-preference-fits-all alignment: Multi-objective direct preference optimization. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 10586–10613, 2024.

A Implementation Details

A.1 Detailed Settings

We implement GPDPO with verl and use vLLM for rollout generation. Experiments use Qwen3-1.7B and Qwen3-4B on RLLA-4K, whose training file contains 3,920 examples; overlength prompts are filtered during loading. Tables 4 and 5 summarize the settings. Unless stated otherwise, all reward dimensions have unit aggregation weights, and GPDPO uses uniform pair weights with $\lambda = 0.1$ and $\beta = 0.1$.

Table 4: Training settings for GPDPO. Batch sizes are measured in prompts.

Setting	Value
Training hardware	One node, $4\times$ NVIDIA RTX 5880 Ada (48 GB each)
Base models	Qwen3-1.7B, Qwen3-4B
Training data	RLLA-4K, 3,920 records before prompt filtering
Max prompt / response length	2,048 / 1,024 tokens
Train batch / mini-batch size	512 / 128 prompts
Rollout group size G	4 (2,048 responses per rollout step)
Training steps / PPO epochs	200 rollout steps / 1 epoch per step
Optimizer / learning rate	AdamW / 10^{-6}, constant, no warmup
Temperature / top-p / top-k	1.0 / 1.0 / disabled
PPO clip range / dual-clip factor	$(\epsilon_{\text{low}}, \epsilon_{\text{high}}) = (0.2, 0.2)$ / 3.0
Policy-loss aggregation	Token mean
Dynamic batching	Enabled; response groups remain intact
vLLM GPU memory utilization	0.60
KL penalty / entropy bonus	Neither used
Rewards	2R: correctness + length; 3R: additionally format
Pairwise loss parameters	$\lambda = 0.1$, $\beta = 0.1$; uniform pair weights
Reference policy	Sampling policy; refreshed every rollout step
Checkpoint interval / main results	Every 50 steps / selected step-200 checkpoint

Each rollout step generates 512×4 responses and performs four mini-batch updates. Correctness assigns partial credit for matching tool names, parameter names, and values, scaled to $[-3, 3]$. Length is $\min(\text{round}(n_{\text{think}}/512, 2), 1)$, where n_{think} counts whitespace-separated words in the `think` block; a missing complete block receives zero. Format is binary and checks balanced, nonrepeated, correctly ordered blocks, with `think` always required and the other required blocks determined by GT.

Table 5: Evaluation settings for the selected step-200 checkpoints.

Setting	Value
Evaluation hardware	One NVIDIA RTX A6000 (48 GB)
Dataset	Processed API-Bank, 597 questions
Level 1 / 2 / 3 questions	399 / 67 / 131
Decoding	Greedy, one response per question
Temperature / top-p	0.0 / 1.0
Max new tokens / total sequence length	4,096 / 4,096
Generation batch / tensor parallel size	1 / 1
Evaluation seed	42
Metrics	Level accuracy, Correct Acc., Length Rew.; Format Acc. for 3R

Correct Acc. is accuracy over all 597 questions. Overall is Correct Acc./100 + Length Rew. for 2R, with Format Acc./100 added for 3R. Each main-table result is a single evaluation of a selected run. The 4,096-token total sequence limit includes the prompt and can reduce the available generation length.

A.2 Detailed Algorithm

Algorithm 1 follows the three components in Section 3.3. Dominance relations and preference pairs are built once per rollout batch. Scores and joint updates are evaluated per mini-batch using the sampling policy fixed at the start of the rollout step.

Algorithm 1 Group Pareto-Dominance Policy Optimization (GPDPO)

Input: Initial policy π_θ, prompt distribution $\mathcal{P}$, reward functions $\{r_i\}_{i=1}^M$, group size G, reward weights $\{w_i\}_{i=1}^M$, loss weight λ, margin scale β
Output: Trained policy π_θ

for each training iteration **do**
  $\theta_{\text{old}} \leftarrow \theta$, $\mathcal{B} \leftarrow \text{SampleBatch}(\mathcal{P})$
  Stage 1: Pareto-dominance identification
  for each prompt $x \in \mathcal{B}$ **do**
    $\mathcal{G}(x) \leftarrow \text{Sample}(\pi_{\theta_{\text{old}}}(\cdot \mid x), G)$
    $\ell_{n,t}^{\text{old}} \leftarrow \log \pi_{\theta_{\text{old}}}(y_{n,t} \mid x, y_{n,<t})$ $(\forall n, t)$
    $\mathbf{r}_n \leftarrow (r_1(x, y_n), \ldots, r_M(x, y_n))$ $(\forall n)$
    $\mu_i \leftarrow G^{-1} \sum_{n=1}^G r_n^i$, $A_n^i \leftarrow (r_n^i - \mu_i)/s_i$ $(\forall n, i)$
    $g(y_n) \leftarrow \mathbf{1}\{\text{sign}(A_n^i)\,\text{sign}(A_n^j) \geq 0,\ \forall i, j\}$ $(\forall n)$
    $\mathcal{F}(x) \leftarrow \{y_n \in \mathcal{G}(x) : g(y_n) = 0\}$, $\mathcal{R}(x) \leftarrow \mathcal{G}(x) \setminus \mathcal{F}(x)$
    $\hat{\kappa}(x) \leftarrow G^{-1} \sum_{n=1}^G g(y_n)$, $\widetilde{A}_n \leftarrow \hat{\kappa}(x) g(y_n) \sum_{i=1}^M w_i A_n^i$
      // Apply the GD^{2}PO conflict filter and retain its standard advantages.
    $\mathcal{D}(x) \leftarrow \{(y_w, y_l) \in \mathcal{G}(x)^2 : \mathbf{r}_w \succ_{\text{P}} \mathbf{r}_l\}$
  end for
  Stage 2: Preference-loss construction
  $\mathcal{E}(x) \leftarrow \{(y_w, y_l) \in \mathcal{D}(x) : y_w \in \mathcal{F}(x) \vee y_l \in \mathcal{F}(x)\}$ $(\forall x \in \mathcal{B})$
    // Select dominance edges involving at least one filtered response.
  $A \leftarrow \text{MaskedWhiten}(\widetilde{A}, g)$, $N_{\mathcal{E}} \leftarrow \sum_{x \in \mathcal{B}} |\mathcal{E}(x)|$
  $\{\mathcal{M}_k\} \leftarrow \text{GroupPartition}(\mathcal{B})$ // Keep response groups intact.
  for each mini-batch $\mathcal{M}_k$ **do**
    $\ell_{n,t}^{\theta} \leftarrow \log \pi_\theta(y_{n,t} \mid x, y_{n,<t})$ $(\forall n, t \text{ in } \mathcal{M}_k)$
    $u_\theta(y_n \mid x) \leftarrow |y_n|^{-1} \sum_{t=1}^{|y_n|} (\ell_{n,t}^{\theta} - \ell_{n,t}^{\text{old}})$
    $\widetilde{u}_\theta(y \mid x) \leftarrow (1 - g(y)) u_\theta(y \mid x) + g(y)\,\text{sg}[u_\theta(y \mid x)]$
      // Retained endpoints serve as fixed anchors in the pairwise loss.
    $\ell_{\text{pair}}(y_w, y_l \mid x) \leftarrow -\log \sigma(\beta[\widetilde{u}_\theta(y_w \mid x) - \widetilde{u}_\theta(y_l \mid x)])$
  Stage 3: Joint policy optimization
    $\mathcal{L}_{\text{pair}}^{(k)} \leftarrow 0$ // An empty pair set contributes zero.
    if $N_{\mathcal{E}} > 0$ **then**
      $\mathcal{L}_{\text{pair}}^{(k)} \leftarrow N_{\mathcal{E}}^{-1} \sum_{x \in \mathcal{M}_k} \sum_{(y_w, y_l) \in \mathcal{E}(x)} \ell_{\text{pair}}(y_w, y_l \mid x)$
    end if
    $\mathcal{L}_{\text{standard}}^{(k)} \leftarrow \text{ClippedPolicyLoss}(\ell^\theta, \ell^{\text{old}}, A; \mathcal{M}_k)$
    $\mathcal{L}_{\text{GPDPO}}^{(k)} \leftarrow \mathcal{L}_{\text{standard}}^{(k)} + \lambda \mathcal{L}_{\text{pair}}^{(k)}$
    $\theta \leftarrow \text{OptimizerStep}(\theta, \nabla_\theta \mathcal{L}_{\text{GPDPO}}^{(k)})$
  end for
end for
return π_θ

Pair construction and gradient flow. The implementation compares the reward-wise normalized advantages before filtering, with tolerance 10^{-8}; positive per-dimension normalization preserves Pareto ordering in exact arithmetic. Sequence scores average log-probability ratios over valid response tokens. Retained endpoints are detached only in the pairwise term; both endpoints receive gradients when both are filtered. Cached sampling-policy probabilities provide the reference without an additional fixed model.

Loss normalization and updates. We implement each pair loss as $\text{softplus}(-\beta m)$, where $m = \widetilde{u}_\theta(y_w \mid x) - \widetilde{u}_\theta(y_l \mid x)$. Every mini-batch and micro-batch uses the full rollout batch's pair count as its denominator. GroupPartition preserves group order; MaskedWhiten whitens retained-token advantages and leaves filtered advantages zero. ClippedPolicyLoss is GD^{2}PO's standard objective; OptimizerStep uses AdamW. Pairs reuse the rollout responses and rewards.

Table 6: Detailed results for Figure 2. Entries report count (percentage). The denominator is 65,536 paired label comparisons per setting for (a,b), and 35,937 / 58,689 filtered responses under 2R / 3R for (c,d). The four categories in (d) partition the filtered set.

Panel	Measurement	2R: count (%)	3R: count (%)
(a)	Expansion, $4 \to 64$: changed	10,608 (16.19)	17,756 (27.09)
	Expansion, $4 \to 64$: unchanged	54,928 (83.81)	47,780 (72.91)
(b)	Peer replacement, $G = 4$: disagreement	5,999 (9.15)	10,434 (15.92)
	Peer replacement, $G = 16$: disagreement	6,312 (9.63)	8,119 (12.39)
	Peer replacement, $G = 64$: disagreement	3,561 (5.43)	3,994 (6.09)
(c)	GT tool-call structure matches	8,345 (23.22)	16,039 (27.33)
(d)	Consistently better	4,900 (13.63)	8,905 (15.17)
	Both	11,122 (30.95)	16,163 (27.54)
	Consistently worse	15,752 (43.83)	25,797 (43.96)
	No consistent relation	4,163 (11.58)	7,824 (13.33)
	At least one consistent relation	31,774 (88.42)	50,865 (86.67)

B Limitations

Our evaluation is limited to two- and three-reward settings, as suitable benchmarks with a larger number of reward dimensions are currently unavailable. Future work will explore reward conflicts in higher-dimensional multi-reward settings and investigate scalable strategies for exploiting informative signals from conflicting responses.

C Additional Details for the Conflict-Filtering Analysis

We supplement Figure 2 with settings and results for Qwen3-1.7B trained with GD^{2}PO under correctness and length rewards (2R), and with the additional format reward (3R).

C.1 Sensitivity to the Reference Group

Experimental setup. We use the step-50 checkpoint from each of the separate 2R and 3R training runs. Both settings use a common set of 128 RLLA-4K training prompts. Each model generates four target responses and a pool of 128 additional responses per prompt, with temperature 1.0 and at most 1,024 response tokens.

Experimental procedure. Target responses and rewards remain fixed. For panel (a), we add 60 pool responses to the original four-response group and compare the four targets' filtering labels at $G = 4$ and $G = 64$. For panel (b), each target is combined with two sets of $G - 1$ peers from the pool, for $G \in \{4, 16, 64\}$. Sampling is without replacement within each set, with overlap allowed between sets. We repeat each comparison 128 times.

Results. Each rate uses $128 \times 4 \times 128 = 65{,}536$ paired label comparisons per reward setting. Expansion changes 10,608/65,536 (16.19%) labels under 2R and 17,756/65,536 (27.09%) under 3R. At $G = 64$, peer replacement still produces 3,561/65,536 (5.43%) and 3,994/65,536 (6.09%) inconsistent labels, respectively (Table 6). Thus, a fixed response's filtering decision remains sensitive to group composition.

C.2 Information Retained in Filtered Responses

Experimental setup. For panels (c,d), we analyze rollout logs from steps 1–200 of the two separate GD^{2}PO training runs, each with $G = 4$. Conflict filtering identifies 35,937 responses under 2R and 58,689 under 3R.

Experimental procedure. Panel (c) measures GT matches of ordered tool-call lists, including call order, tool names, and parameter names and values. Panel (d) compares each filtered response with its three original peers using Pareto dominance, assigning it to one of the four mutually exclusive

Table 7: Comparison-edge statistics under 2R. All edge counts involve at least one filtered response. Percentages for added edges use the Pareto-edge count as the denominator.

Measurement	GD^{2}PO	GPDPO ($\lambda = 0.1$)
All-dimension strict-improvement edges	6,374	6,338
Pareto-dominance edges	52,421	52,631
Edges added by allowing ties	46,047 (87.84%)	46,293 (87.96%)
Tied correctness, improved length	29,229	29,578
Tied length, improved correctness	16,818	16,715

categories in Table 6. "Both" means better than some peers and worse than others. All percentages use the full filtered set as the denominator.

Results. GT matches account for 8,345/35,937 (23.22%) and 16,039/58,689 (27.33%) of the 2R and 3R filtered sets, respectively. At least one consistent relation exists for 31,774/35,937 (88.42%) and 50,865/58,689 (86.67%), leaving 4,163 and 7,824 responses without such relations. These results show that filtered responses can retain task-relevant content and consistent comparisons, supporting the recovery of pairwise supervision.

D Experiments on Pareto-Dominance Edges

We examine how allowing ties in individual reward dimensions affects the comparison edges available for recovering pairwise supervision from filtered responses.

Experimental setup. We analyze steps 1–200 of historical GD^{2}PO and GPDPO ($\lambda = 0.1$) training runs with Qwen3-1.7B on RLLA-4K, using correctness and length rewards (2R) and response groups of size $G = 4$.

Experimental procedure. We label responses using the training-time conflict filter and compare two criteria: *all-dimension strict improvement*, which requires higher rewards in every dimension; and *Pareto dominance*, which requires no lower reward in any dimension and a higher reward in at least one. Both criteria use the same candidate pairs: within-group pairs involving at least one filtered response. Retained–retained pairs are excluded, and each edge is counted once in the better-to-worse direction. We count edges under each criterion and identify the tied dimension in edges admitted by Pareto dominance but excluded by strict improvement.

Results. Table 7 shows that Pareto dominance retains 52,421 edges under GD^{2}PO, compared with 6,374 under all-dimension strict improvement. The additional 46,047/52,421 (87.84%) edges include 29,229 ties in correctness and 16,818 ties in length, with improvement in the other reward. The GPDPO run shows a similar pattern, with 46,293/52,631 (87.96%) Pareto edges involving such ties.

Pareto dominance allows ties in some reward dimensions while using improvements in others to provide supervision. It retains more comparison edges than all-dimension strict improvement, thereby providing richer pairwise learning signals.