PACPO: POLARITY-AWARE CONTRIBUTION POLICY OPTIMIZATION FOR MULTI-REWARD RL

Anonymous authors
Paper under double-blind review

ABSTRACT

As the deployment of large language models expands to complex reasoning, tool-use, and interactive applications, post-training increasingly involves multiple behavioral objectives. To jointly optimize these objectives, multi-reward reinforcement learning has become a key paradigm in language model post-training. Existing methods typically preserve reward-wise information during advantage estimation but merge these signals before applying probability-ratio clipping, causing all reward dimensions to share the boundary response determined by the aggregated advantage. When the policy ratio falls outside the clipping interval, the shared clipping response may mask corrective gradients from reward contributions of the opposite polarity. To address this issue, we propose **P**olarity-**A**ware **C**ontribution **P**olicy **O**ptimization (**PACPO**), which separates reward contributions by polarity and applies polarity-specific clipping responses. This design allows corrective signals from contributions of opposite polarity to remain active, rather than being suppressed by clipping after aggregation. By preserving these otherwise masked signals, PACPO mitigates clipping-boundary gradient masking, thereby promoting more balanced joint optimization across multi-reward objectives. Experiments across diverse multi-reward post-training settings show that PACPO consistently outperforms existing multi-reward optimization methods, demonstrating its effectiveness for multi-reward policy optimization.

1 INTRODUCTION

With the widespread deployment of large language models (LLMs) across real-world scenarios Kumar et al. (2026); Wu et al. (2024); Qin et al. (2024); Lou et al. (2024); Xu et al. (2025), models increasingly need to accommodate diverse task-specific requirements and human preferences Shao et al. (2024); Qian et al. (2026); Zhou et al. (2024). These diverse demands translate into a broad set of behavioral objectives, spanning task efficiency Lou et al. (2024), response coherence Liu et al. (2024) and logical consistency Yang et al. (2025), safety Bai et al. (2022), fairness Zhang et al. (2025b), and other aspects of model behavior. To jointly optimize these objectives, LLM post-training is increasingly shifting toward multi-reward reinforcement learning (RL) Li et al. (2025), which assigns distinct reward signals to different behavioral objectives Zhu et al. (2025). Despite the promise of multi-reward optimization, it remains a fundamental challenge to preserve the distinct optimization effects of heterogeneous reward signals within a unified policy update.

To address the challenges mentioned above, recent methods coordinate heterogeneous reward signals at the advantage level Shrivastava et al. (2026); Lu et al. (2026), producing advantage estimates that directly drive policy updates Liu et al. (2026b); Shao et al. (2024); Pavlenko et al. (2026); Zhang et al. (2025a). By normalizing reward feedback during advantage estimation, this formulation helps reconcile differences in reward scale and distribution. One direct approach is to follow GRPO Shao et al. (2024) by first aggregating multiple rewards and then constructing group-relative advantages. However, in multi-reward settings, it aggregates rewards before normalization, obscuring reward-specific information and weakening the corresponding learning signals. GDPO Liu et al. (2026b) computes group-relative advantages separately for each reward dimension, preserving reward-specific information until aggregation. However, GDPO still combines these advantages into a single scalar before policy optimization. Consequently, all reward-wise contributions share the clipping-boundary response determined by the sign of the aggregate advantage when cross-reward disagreement persists.

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.

This shared boundary response arises because standard PPO-style updates aggregate reward-wise advantages before clipping Schulman et al. (2017); Liu et al. (2026b;a), compressing distinct reward-specific update requirements into a single aggregate advantage. When different reward dimensions favor different update directions Yu et al. (2020); Kim et al. (2025); Li et al. (2025), their respective contributions therefore correspond to different boundary responses Lu et al. (2023); Navon et al. (2022). We refer to this directional sign as *polarity*. After aggregation, however, only the aggregate advantage enters the clipping operation, and its sign determines the shared boundary response for all reward-wise contributions. Once the policy ratio crosses the boundary corresponding to the update direction of the aggregate advantage, its policy-gradient contribution vanishes, causing corrective gradients from reward-wise contributions with different update direction to be suppressed. We refer to this phenomenon as *clipping-boundary gradient masking*. Addressing this issue requires reward-wise contributions to remain distinguishable under their clipping responses until the policy objective determines the effective gradient.

To this end, we propose **Polarity-Aware Contribution Policy Optimization (PACPO)**[1], which extends reward-wise information preservation from advantage estimation to policy optimization. PACPO constructs explicit reward-wise contributions on a shared scale while preserving the positive or negative direction of each contribution as its polarity, reflecting its directional effect on the policy update. To prevent drastic gradient changes caused by clipping the full polarity contributions, PACPO decomposes them into the net preference and the cancelable disagreement, and introduces a conflict-retention mechanism to control the fraction of disagreement retained. It then applies clipping separately to the adjusted positive and negative polarity components, allowing each polarity to retain its corresponding boundary response before aggregation. Together, these mechanisms preserve the original aggregate advantage within a single policy objective while allowing otherwise masked corrective gradients to remain active under clipping.

We evaluate PACPO in three multi-reward settings, covering tool calling on API-Bank, scientific question answering on ScienceQA and GPQA, and summarization on CNN/DailyMail. In API-Bank and scientific QA, PACPO consistently achieves the highest overall scores across all evaluated backbones, and in CNN/DailyMail it achieves the best mean UniEval score with the highest coherence and consistency. These results highlight PACPO's potential for multi-reward optimization.

Our contributions are summarized as follows:

- We identify **clipping-boundary gradient masking**, showing that scalarizing reward-wise contributions before clipping can subject distinct reward dimensions to a unified boundary response and suppress corrective gradients.
- We propose **PACPO**, which preserves explicit reward-wise contributions and applies polarity-aware clipping separately by polarity, with a retention coefficient controlling the conflict retained before aggregation, thereby maintaining reward-specific update effects.
- We evaluate PACPO in multiple multi-reward post-training settings and observe consistent performance gains over strong baselines across diverse tasks.

2 PRELIMINARIES

2.1 MULTI-REWARD REINFORCEMENT LEARNING

Traditional policy optimization assumes that model behavior can be evaluated by a scalar reward. Let $\mathcal{D}$ denote a distribution of prompts and let π_θ denote the policy model. Given a prompt $x \sim \mathcal{D}$, the policy generates a response $y \sim \pi_\theta(\cdot \mid x)$. For a scalar reward function $r(x, y)$, policy optimization seeks to maximize the expected reward Schulman et al. (2017):

$$\mathcal{J}(\theta) = \mathbb{E}_{x\sim\mathcal{D},\, y\sim\pi_\theta(\cdot|x)}\left[r(x, y)\right]. \tag{1}$$

Multi-reward reinforcement learning extends this setting by evaluating each response along K distinct reward dimensions Roijers et al. (2013). The feedback assigned to a response is represented by a reward vector

$$\mathbf{r}(x, y) = \left(r^1(x, y), \ldots, r^K(x, y)\right), \tag{2}$$

[1]Anonymous code repository: `https://anonymous.4open.science/r/PACPO-1244/`

where each reward function captures a particular property of the response, such as task correctness, safety, response format, or generation efficiency.

Let $\{w_k\}_{k=1}^K$ denote the weights assigned to the individual reward dimensions, which determine their relative importance in optimizing the shared policy. The resulting multi-reward objective is:

$$\mathcal{J}_{\mathbf{w}}(\theta) = \mathbb{E}_{x\sim\mathcal{D},\, y\sim\pi_\theta(\cdot|x)}\left[\sum_{k=1}^{K} w_k r^k(x,y)\right]. \tag{3}$$

This process is complicated by scale sensitivity, calibration mismatch, and preference conflicts.

2.2 Group-Based Policy Optimization

Group Relative Policy Optimization (GRPO) Shao et al. (2024) avoids training an additional value model by estimating advantages relative to a group of responses sampled for the same prompt. For each prompt x, the old policy $\pi_{\theta_{\text{old}}}$ samples G responses $y_n \sim \pi_{\theta_{\text{old}}}(\cdot \mid x)$, where $n = 1, \ldots, G$. Under the multi-reward objective defined above, GRPO first combines the reward dimensions and then standardizes the resulting scalar reward within the group:

$$r_n^{\text{sum}} = \sum_{k=1}^{K} w_k r_n^k, \qquad A_n^{\text{GRPO}} = \frac{r_n^{\text{sum}} - \text{mean}_j[r_j^{\text{sum}}]}{\text{std}_j[r_j^{\text{sum}}] + \epsilon_{\text{adv}}}. \tag{4}$$

To mitigate off-policy discrepancy during optimization, GRPO applies importance sampling at the token level and defines the policy ratio as:

$$\rho_{n,t}(\theta) = \frac{\pi_\theta(y_{n,t} \mid x, y_{n,<t})}{\pi_{\theta_{\text{old}}}(y_{n,t} \mid x, y_{n,<t})}. \tag{5}$$

Following PPO Schulman et al. (2017), GRPO constrains policy updates by applying PPO clipping to the importance-weighted advantage. Given a response-level advantage A, the corresponding token-level clipped surrogate term is

$$\Psi_\epsilon(\rho_{n,t}(\theta), A) = \min\left(\rho_{n,t}(\theta)A,\ \text{clip}\left(\rho_{n,t}(\theta), 1-\epsilon, 1+\epsilon\right) A\right), \tag{6}$$

where ϵ denotes the PPO clipping threshold. The GRPO objective averages this surrogate term over all sampled responses and their tokens.

Because GRPO performs group-wise standardization after reward aggregation, different reward configurations may be mapped to the same group-relative advantage. Group reward-Decoupled Policy Optimization (GDPO) Liu et al. (2026b) changes this order by first computing a separate group-relative advantage for each reward dimension:

$$A_n^k = \frac{r_n^k - \text{mean}_j[r_j^k]}{\text{std}_j[r_j^k] + \epsilon_{\text{adv}}}, \qquad k = 1, \ldots, K, \tag{7}$$

These reward-wise advantages are then aggregated and standardized over the training batch $\mathcal{B}$:

$$A_n^{\text{sum}} = \sum_{k=1}^{K} w_k A_n^k, \qquad \widehat{A}_n^{\text{GDPO}} = \frac{A_n^{\text{sum}} - \text{mean}_{\mathcal{B}}[A^{\text{sum}}]}{\text{std}_{\mathcal{B}}[A^{\text{sum}}] + \epsilon_{\text{batch}}}. \tag{8}$$

GDPO therefore preserves reward-wise relative variation during advantage estimation while maintaining a stable scale for the final policy update.

3 Methods

Although prior methods have made progress in preserving dimension-wise feedback during advantage estimation, they do not resolve the gradient masking caused by aggregating reward-wise contributions into a single scalar before PPO clipping. To address this issue, our key idea is to explicitly maintain reward-wise contributions on a shared scale, preserving their respective polarities. We then use a retention mechanism to highlight the net preference and apply PPO clipping separately by polarity, allowing each polarity to retain its corresponding boundary response before aggregation. Building on this, we propose **Polarity-Aware Contribution Policy Optimization (PACPO)**. In this section, we first introduce the construction of reward-wise contributions, then present polarity-aware conflict retention, and finally formulate whole policy optimization process.

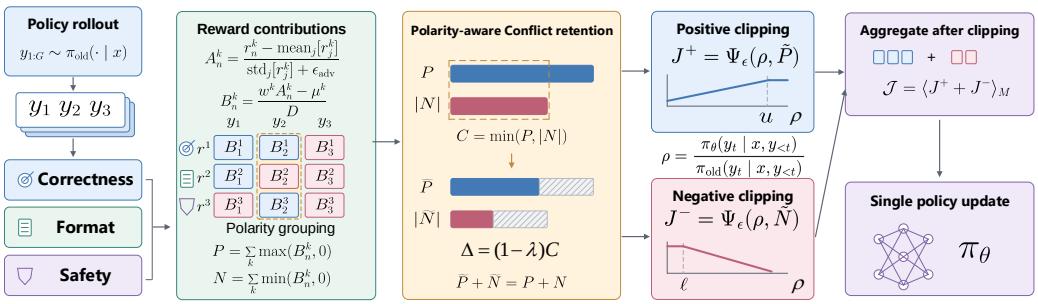


Figure 1: Method overview of PACPO.

3.1 Reward-wise Contribution

PACPO extends reward-wise information preservation to policy optimization by reconstructing each reward dimension contribution on a shared scale. This representation preserves the polarity of each contribution in the clipping objective while exactly recovering the original aggregate advantage. Specifically, denoting the weighted reward-wise signal by $X_{i,j}^{(k)} = w_k A_{i,j}^{(k)}$, we define

$$B_{i,j}^{(k)} = \frac{X_{i,j}^{(k)} - \mu_k}{D}, \tag{9}$$

where i indexes the prompt group within the training batch, j is the sampled response, and k is the reward dimension. The term μ_k and D represent the batch mean of the k-th weighted reward-wise advantage and the denominator of the final GDPO advantage in Eq.8, respectively.

With this construction, the reward-wise contributions sum exactly to the normalized advantage used by the aggregate objective:

$$\sum_{k=1}^{K} B_{i,j}^{(k)} = A_{i,j}^{\mathrm{G}}. \tag{10}$$

where $A_{i,j}^{\mathrm{G}}$ denotes the final GDPO advantage $\widehat{A}_n^{\mathrm{GDPO}}$ in Eq.8. Eq.10 provides the foundation for PACPO, as it preserves the GDPO-estimated preference while revealing the internal reward structure obscured by scalar aggregation. This implies that two different rollouts can share the same aggregate advantage $A_{i,j}^{\mathrm{G}}$ while being supported by very different reward-wise contribution profiles. However, this scalar identity mixes positive and negative contributions during summation, and thus does not reveal the directional structure of the advantage. To retain this directional information, we group the contributions according to their polarity:

$$P_{i,j} = \sum_{k=1}^{K} \left[B_{i,j}^{(k)}\right]_+, \qquad N_{i,j} = \sum_{k=1}^{K} \left[B_{i,j}^{(k)}\right]_-, \tag{11}$$

where $[z]_+ = \max(z, 0)$ and $[z]_- = \min(z, 0)$. The positive component $P_{i,j}$ aggregates the reward contributions that favor the current rollout, whereas the negative component $N_{i,j}$ aggregates those that oppose it. Importantly,

$$P_{i,j} + N_{i,j} = A_{i,j}^{\mathrm{G}}, \tag{12}$$

so the pair $(P_{i,j}, N_{i,j})$ preserves the original GDPO preference while leveraging polarity information to reveal whether that preference results from consistent reward evidence or from cancellation between competing objectives.

3.2 Polarity-Aware Conflict Retention

Section 3.1 reconstructs the aggregate advantage as reward-wise contributions with positive or negative polarity, explicitly exposing the contribution of each polarity and the disagreement between them. However, the decomposed polarity-aware contributions are not equivalent to the net preference along each polarity, and directly using them would amplify the boundary contribution of a

single polarity. Here, we decouple the net preference from the disagreement between positive and negative polarities, allowing this disagreement to influence the nonlinear clipping stage under explicit control. This mechanism quantifies the mutually canceling portion of positive and negative evidence via the cancelable disagreement mass, and regulates the visibility of this conflict through a retention coefficient $\lambda \in [0, 1]$. This construction preserves the aggregate advantage while allowing the retained disagreement to influence the subsequent clipped objective.

The key observation is that the scalar advantage $A^{\mathrm{G}}_{i,j} = P_{i,j} + N_{i,j}$ only records the net outcome, while the cancelable disagreement remains invisible. PACPO separates this cancelable disagreement from the net preference. Specifically, we define the *cancelable disagreement mass* as:

$$C_{i,j} = \min\left(P_{i,j}, |N_{i,j}|\right). \tag{13}$$

It quantifies the maximal amount of positive and negative evidence that can annihilate each other without reversing the polarity of either component. Intuitively, $C_{i,j}$ measures the overlap between supporting and opposing reward signals in aggregation, and thus serves as a direct proxy for the severity of cross-reward disagreement within a rollout.

Simply preserving all of this disagreement, however, would allow both polarity components to independently influence the clipped objective, amplifying their effects on the policy gradient. Under full retention, clipping either component can cause a large drop in its gradient contribution and thereby exacerbate update fluctuations. PACPO therefore uses a fixed retention coefficient $\lambda \in [0, 1]$ to control how much of the cancelable disagreement remains visible to policy optimization. The retained polarity components are

$$\begin{aligned} \widetilde{P}_{i,j} &= P_{i,j} - (1-\lambda)C_{i,j}, \\ \widetilde{N}_{i,j} &= N_{i,j} + (1-\lambda)C_{i,j}. \end{aligned} \tag{14}$$

The amount $(1-\lambda)C_{i,j}$ is canceled between the positive and negative components before policy optimization, while the remaining $\lambda C_{i,j}$ is retained as opposing-polarity reward evidence. The coefficient λ controls how much of the cancelable disagreement remains explicit in $\widetilde{P}_{i,j}$ and $\widetilde{N}_{i,j}$, thereby regulating the visibility of cross-reward conflict in the clipped objective.

Importantly, the retained components still satisfy $\widetilde{P}_{i,j} + \widetilde{N}_{i,j} = A^{\mathrm{G}}_{i,j}$. This preservation property ensures that PACPO keeps the same advantage as GDPO while exposing the polarity information concealed by scalar aggregation.

3.3 Polarity-Aware Contribution Policy Optimization

In the preceding section, we constructed the retained polarity components $\widetilde{P}_{i,j}$ and $\widetilde{N}_{i,j}$ to preserve a controlled amount of the disagreement otherwise canceled by scalar aggregation. These components still sum to the aggregate advantage $A^{\mathrm{G}}_{i,j}$. To ensure that this retained disagreement influences policy optimization, PACPO keeps the two components separate and applies PPO clipping to each of them independently. This design preserves the GDPO aggregate preference within the linear region, while allowing each component to respond separately when the probability ratio crosses the clipping boundary. Consequently, PACPO retains reward disagreement precisely where nonlinear clipping makes it consequential.

Using the token-level policy ratio $\rho_{i,j,t}(\theta)$ defined in Eq.5 and the standard PPO clipped surrogate $\Psi_\epsilon(\rho, A)$ in Eq.6, PACPO computes the surrogate separately for $\widetilde{P}_{i,j}$ and $\widetilde{N}_{i,j}$:

$$\Gamma^{\mathrm{PACPO}}_{i,j,t} = \Psi_\epsilon\left(\rho_{i,j,t}, \widetilde{P}_{i,j}\right) + \Psi_\epsilon\left(\rho_{i,j,t}, \widetilde{N}_{i,j}\right). \tag{15}$$

This change is important because reward aggregation and PPO clipping are not interchangeable for opposite-polarity contributions. When the policy ratio remains inside the clipping interval, the surrogate is linear in the advantage, and therefore

$$\Gamma^{\mathrm{PACPO}}_{i,j,t} = \rho_{i,j,t}\left(\widetilde{P}_{i,j} + \widetilde{N}_{i,j}\right) = \rho_{i,j,t} A^{\mathrm{G}}_{i,j}. \tag{16}$$

Taking the derivative with respect to the probability ratio, we obtain:

$$\frac{\partial \Gamma_{i,j,t}^{\mathrm{PACPO}}}{\partial \rho_{i,j,t}} = \begin{cases} \widetilde{P}_{i,j}, & \rho_{i,j,t} < 1-\epsilon, \\ A_{i,j}^{\mathrm{G}}, & 1-\epsilon < \rho_{i,j,t} < 1+\epsilon, \\ \widetilde{N}_{i,j}, & \rho_{i,j,t} > 1+\epsilon. \end{cases} \tag{17}$$

Under polarity-specific clipping, crossing a clipping boundary does not eliminate all gradient signals. It suppresses only the polarity component that has already moved excessively in its preferred direction, while the opposite-polarity component remains active as a corrective signal. Therefore, when $\rho_{i,j,t} < 1-\epsilon$, the negative-polarity component has already crossed the lower boundary in its preferred direction and is clipped. The positive-polarity component remains active, supporting the response and providing the only effective signal to prevent the probability ratio from decreasing further and correct the excessive update. When $\rho_{i,j,t} > 1+\epsilon$, the roles of the two polarity components are reversed. If these components are aggregated before clipping, as in GDPO, the sign of the aggregate advantage may render the entire surrogate objective flat, thereby masking the boundary correction that should otherwise emerge. PACPO instead keeps $\widetilde{P}_{i,j}$ and $\widetilde{N}_{i,j}$ separate, so that cross-reward disagreement does not vanish under scalar clipping. When the probability ratio crosses a clipping boundary, this disagreement is transformed into a corrective effect directed toward the clipping interval.

The final PACPO objective is obtained by substituting the polarity-aware surrogate $\Gamma_{i,j,t}^{\mathrm{PACPO}}$ into the same response- and token-level averaging objective used by GDPO, together with a KL penalty and an entropy bonus:

$$\mathcal{J}_{\mathrm{PACPO}}(\theta) = \mathbb{E}_{x_i \sim \mathcal{D},\, \mathcal{G}(x_i)} \left[\frac{1}{G} \sum_{j=1}^{G} \frac{1}{|y_{i,j}|} \sum_{t=1}^{|y_{i,j}|} \left(\Gamma_{i,j,t}^{\mathrm{PACPO}} - \beta \mathcal{L}_{\mathrm{KL},i,j,t} + \eta \mathcal{H}_{i,j,t} \right) \right]. \tag{18}$$

Here, $\mathcal{L}_{\mathrm{KL},i,j,t}$ and $\mathcal{H}_{i,j,t}$ denote the token-level KL divergence and entropy, following the same convention as GDPO. PACPO requires only two contribution tensors and reuses the same forward and backward passes, introducing no additional parameters or inference cost.

4 Experiments

In this section, we first report comparisons of PACPO with common methods across multiple multi-reward scenarios to demonstrate its effectiveness in multi-reward alignment and conflict mitigation. We then conduct additional ablation studies on key factors of PACPO, including parameter λ and conflict retention mechanism.

4.1 Experiment Setup

We evaluate the effectiveness of PACPO compared with GRPO Shao et al. (2024), GDPO Liu et al. (2026b), and GD^{2}PO Liu et al. (2026a) on three different tasks, including tool calling, scientific question answering, and text summarization.

For the tool calling task, we evaluate the trained models on API-Bank Li et al. (2023a) and use the RLLA training set for policy optimization following ToolRL Qian et al. (2026). This task involves three rewards: correctness, format, and length. For scientific reasoning, we train models on SciKnowEval Feng et al. (2024) and evaluate them on ScienceQA Saikh et al. (2022); Lu et al. (2022) and GPQA Rein et al. (2023) benchmarks, optimizing two rewards: answer correctness and format compliance. We also report joint success, the percentage of responses that are both correct and properly formatted. For text summarization, we use the official train and test splits of the CNN/DailyMail dataset Nallapati et al. (2016), which involve five reward dimensions: coherence, consistency, fluency, relevance, and conciseness.

Backbones and Baselines We evaluate PACPO on instruction-tuned backbones across three tasks. For API-Bank, we use Qwen2.5-1.5B-Instruct, Qwen2.5-3B-Instruct Qwen Team (2024), and Llama-3.2-3B-Instruct Grattafiori et al. (2024). Scientific question answering uses Qwen2.5-3B-Instruct and Llama-3.1-8B-Instruct, while CNN/DailyMail summarization uses Qwen2.5-7B-Instruct. We compare PACPO with GRPO Shao et al. (2024), GDPO Liu et al. (2026b), and GD^{2}PO-Hard Liu et al. (2026a). And we introduce Hypervolume (HV) to assess joint performance across

Table 1: Main results on API-Bank across three backbones. Level 1–3 Acc. denotes accuracy at each task level. Correct and Format denote overall task accuracy and format-compliance rate, respectively, both reported as percentages. Bold indicates the best result within each backbone.

Model	Method	level1 Acc	level2 Acc	level3 Acc	Correct	Format	Length	Overall
Qwen2.5-1.5B Instruct	GRPO	47.78±9.25	36.13±7.39	33.61 ±8.67	43.36 ±8.52	83.21 ±16.75	1.00±0.00	2.266
	GDPO	58.22±6.42	44.24±6.28	35.93 ±5.81	51.76 ±6.13	84.86 ±11.54	0.99 ±0.01	2.356
	GD^{2}PO-Hard	59.65 ±5.17	43.28 ±7.29	41.22 ±5.08	53.77 ±5.19	87.44 ±9.82	0.99±0.01	2.402
	PACPO	**64.40**±4.33	**55.22**±5.96	**45.04**±2.14	**59.12**±3.63	**98.50** ±1.65	**1.00**±0.00	**2.576**
Qwen2.5-3B Instruct	GRPO	67.67 ±2.70	55.22±7.71	**47.33** ±7.36	61.81±3.75	98.16 ±1.07	1.00 ±0.00	2.600
	GDPO	67.17 ±3.37	58.21±7.58	40.46±5.19	60.30 ±4.13	99.33±0.60	**1.00**±0.00	2.596
	GD^{2}PO-Hard	67.17 ±3.13	58.21 ±8.95	41.22 ±7.26	60.47±4.32	99.66 ±0.42	1.00 ±0.00	2.601
	PACPO	**68.42**±2.19	**59.70**±4.89	46.56±3.64	**62.65** ±2.57	**99.83** ±0.23	1.00 ±0.00	**2.625**
Llama-3.2-3B Instruct	GRPO	59.60 ±2.54	34.72 ±7.71	50.61 ±5.46	54.84 ±3.41	93.16 ±4.97	0.99 ±0.01	2.470
	GDPO	60.40 ±4.95	37.31 ±6.39	50.38 ±5.90	55.61 ±4.03	94.30±4.82	**1.00**±0.00	2.499
	GD^{2}PO-Hard	59.15 ±4.61	43.28 ±7.88	**55.73**±8.49	56.62 ±5.25	**100.00**±0.00	**1.00**±0.00	2.566
	PACPO	**61.65**±3.47	**46.27** ±7.67	**55.73**±8.24	**58.63** ±4.13	**100.00** ±0.00	**1.00** ±0.00	**2.586**

Table 2: Main results on scientific question answering across two backbones. Fmt denotes strict format compliance, Acc denotes answer accuracy, and Joint requires both on the same example. Avg is the mean of Fmt, Acc, and Joint. Bold marks the best value within each backbone.

	ScienceQA				GPQA-Main				
Model / Method	Fmt	Acc	Joint	Avg	Fmt	Acc	Joint	Avg	Overall
Qwen2.5-3B-Instruct	61.67	75.22	50.14	62.34	80.36	27.46	23.21	43.68	53.01
GRPO	**99.37**	**81.57**	**81.28**	**87.41**	89.58	28.51	25.55	47.88	67.65
GDPO	99.23	81.36	81.27	87.29	**91.74**	29.69	28.57	50.00	68.64
GD^{2}PO-Hard	98.37	80.02	79.73	86.04	88.84	28.57	26.79	48.07	67.05
PACPO	99.23	81.28	80.99	87.17	91.52	**32.37**	**30.80**	**51.56**	**69.37**
Llama-3.1-8B-Instruct	96.06	90.20	87.03	91.10	70.98	28.79	24.33	41.37	66.23
GRPO	99.53	90.26	90.17	93.32	85.34	27.35	26.11	46.27	69.80
GDPO	99.62	90.30	90.11	93.34	**87.72**	27.90	26.12	47.25	70.29
GD^{2}PO-Hard	99.60	**90.68**	**90.68**	93.65	85.16	29.30	26.69	47.05	70.35
PACPO	**99.81**	90.59	90.59	**93.66**	85.94	**30.13**	**29.70**	**48.59**	**71.13**

Table 3: Main results on CNN/DailyMail summarization with Qwen2.5-7B-Instruct. The five dimension scores and Overall are multiplied by 100 (↑), while HV is reported in units of 10^{-3} (↑). Overall averages all five dimensions. Bold marks the best displayed value in each column.

Method	Coherence	Consistency	Fluency	Relevance	Conciseness	HV score	Overall
Qwen2.5-7B-Instruct	94.42	83.22	94.80	89.19	80.18	0.100	88.36
GRPO	96.71	91.82	95.65	**93.76**	**90.88**	0.361	93.76
GDPO	97.13	92.91	95.76	93.75	89.16	0.399	93.74
GD2PO-Hard	94.74	92.83	**95.98**	92.63	82.87	0.304	91.81
PACPO	**98.52**	**93.49**	95.91	93.73	90.72	**0.462**	**94.47**

reward objectives in this task. HV measures the volume dominated by each method's reward vector relative to a shared reference point, with higher values indicating stronger joint performance.

4.2 Main results

Table 1, 2 and 3 summarize the main evaluation results for PACPO and Comparisons. We set $\lambda = 0.15$ in all three experiments. See the Appendix B.4 for details.

Tool calling. Table 1 reports the results on the API-Bank benchmark across three backbones. As we can see, PACPO achieves the highest Overall score on these backbones. Summed across the

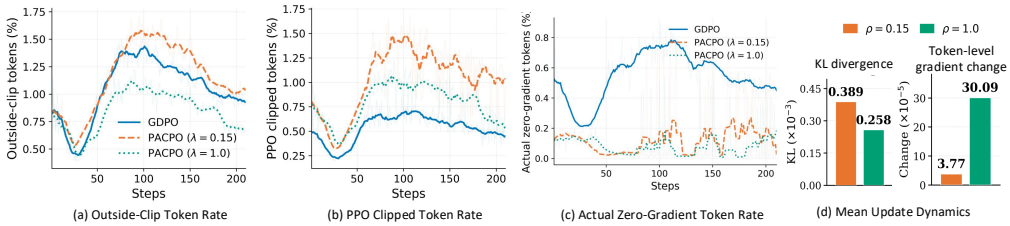


Figure 2: Clipping and token-level gradient dynamics on API-Bank.

backbones, its Overall score exceeds that of GD^{2}PO-Hard, the strongest baseline, by 2.88% and that of GDPO by 4.51%, demonstrating its effectiveness in multi-reward optimization. All methods achieve Length scores close to 1.00. For accuracy, PACPO achieves the highest accuracy at most task levels and the highest overall correct score on every backbone. Its aggregate correct score is higher than GD^{2}PO-Hard and GDPO, indicating PACPO can provide more accurate policy responses for tool calling. Most importantly, the improvement of PACPO does not rely solely on accuracy; it also achieves the highest score in format, which demonstrates PACPO's ability to handle multi-rewards.

Scientific question answering. For scientific question answering, as shown in Table 2, PACPO attains the best overall scores across the two backbones. PACPO achieves Overall scores on both Qwen2.5-3B-Instruct and Llama-3.1-8B-Instruct. A notable observation is that PACPO yields larger gains on more challenging tasks. The improvement is less pronounced on the relatively easier ScienceQA, but more substantial on the more challenging GPQA. On GPQA-Main, PACPO improves the Avg score over GDPO, the strongest baseline, by 3.12% and 2.84% on the two backbones, respectively. This suggests that PACPO is more sensitive to multi-reward signals and can effectively capture the direction of policy updates even in scenarios where rewards are difficult to obtain.

Text summarization. Based on the results from Table 3, PACPO outperforms all methods on CNN/DailyMail, achieving the highest HV score and overall score. GRPO places more emphasis on relevance and conciseness, thus achieving slightly higher scores than PACPO on these two dimensions, but it pays less attention to consistency and coherence, resulting in lower scores on these two aspects. In contrast, PACPO maintains strong performance across all five reward dimensions, which is consistent with its leading HV score. And its highest overall score further demonstrates PACPO's adaptability and effectiveness on multi-reward tasks.

4.3 Mechanism Analysis of Clipping-Boundary Gradient Masking

To validate the polarity-specific clipping mechanism, we compare GDPO with PACPO under $\lambda = 1.0$ (full-PAC) and $\lambda = 0.15$, with details in Appendix B.6. In Fig. 2 (a), all methods exhibit policy ratios outside the clipping interval during training. Since GDPO aggregates reward contributions before clipping, it produces relatively conservative and flattened updates and therefore exhibits the lowest clipping rate (Fig. 2(b)). However, its aggregate clipping response can suppress the entire token-local policy gradient, resulting in the highest actual zero-gradient token rate (Fig. 2(c)). In contrast, full-PAC retains the cancelable disagreement. Compared with $\lambda = 0.15$, which retains only part of this disagreement, $\lambda = 1.0$ produces larger polarity-specific corrections to the gradients. Although this reduces both the fraction of policy ratios outside the clipping interval and the zero-gradient rate, retaining all opposing-polarity signals may intensify gradient conflict and interfere with net-preference optimization, resulting in lower final performance than $\lambda = 0.15$. Overall, PACPO replaces the shared clipping response of the aggregated advantage with polarity-specific responses, which preserve corrective gradients and mitigate gradient masking.

4.4 Ablation Study

Hyperparameter λ. To analyze conflict retention, we vary λ from 0 to 1 on the same model. As shown in Table 4, $\lambda = 0$ removes all cancelable positive and negative contributions, retaining only the net preference but discarding reward-specific corrective information. In contrast, $\lambda = 1$ retains all cancelable disagreement, which preserves such information but may intensify gradient conflict and interfere with net-preference optimization. Intermediate values balance these extremes

Table 4: Effect of the conflict-retention coefficient λ on API-Bank.

λ	Correct.	Length	Format	Overall
0.00	61.31	1.00	99.50	2.608
0.10	61.58	1.00	99.65	2.612
0.15	**62.92**	1.00	**99.82**	**2.627**
0.20	**62.92**	1.00	95.64	2.586
1.00	61.85	1.00	99.66	2.615

Table 5: Effect of polarity-specific clipping across two backbones on API-Bank.

Model	Clipping	Correctness	Length	Format
Qwen2.5-3B Instruct	w/o polarity	60.64	1.00	**99.99**
	PACPO	**62.23**	1.00	99.82
Llama-3.2-3B Instruct	w/o polarity	54.44	1.00	99.50
	PACPO	**58.82**	**1.00**	**100.00**

by retaining useful corrective signals without introducing the full disagreement. Among them, $\lambda = 0.15$ performs best, indicating that moderate conflict retention provides a better balance between the net preference and reward-specific optimization requirements.

Polarity-specific clipping We further ablate polarity-specific clipping across two models, with the experimental settings described in Appendix B.5. As shown in Table 5, the full PACPO achieves better results on both models. Without this mechanism, positive and negative contributions are aggregated before clipping and share the boundary response determined by the sign of the aggregated advantage. When the policy ratio crosses the corresponding boundary, corrective gradients from opposite-polarity contributions may also be masked. In contrast, PACPO clips polarity contributions separately before aggregation, allowing one branch to retain effective corrective gradients when the other is clipped, thereby mitigating gradient masking.

5 Related Work

Reinforcement Learning for LLMs. Reinforcement learning adapts language models through reward feedback Christiano et al. (2017); Stiennon et al. (2020); Ramamurthy et al. (2022); Li et al. (2023b), with PPO-based RLHF providing an established approach to instruction alignment (Ouyang et al., 2022; Schulman et al., 2017). GRPO estimates advantages from sampled response groups without a separate critic (Shao et al., 2024). Subsequent methods refine policy updates from different perspectives: SAPO uses smooth, sign-dependent gates (Gao et al., 2025), and DHPO combines token- and sequence-level ratios with separate clipping (Min et al., 2026). These methods develop complementary approaches to advantage estimation and update regulation in language-model reinforcement learning, while their extension to multi-reward settings remains an active direction.

Multi-Reward Alignment. Multi-reward alignment methods address heterogeneous feedback through reward-space optimization or policy composition. GDPO normalizes each reward separately before aggregating and batch-normalizing advantages, preserving distinctions obscured by aggregate reward normalization (Liu et al., 2026b). GD^{2}PO further addresses reward disagreement using hard or signal-to-noise-based rollout filtering and query-level reweighting (Liu et al., 2026a). Policy-composition approaches instead retain reward-specialized behaviors: Rewarded Soups interpolates parameters of models trained on different rewards (Rame et al., 2023), while PRISM learns reward-specific positive policies and a global negative policy, composing their logits to support adjustable preference trade-offs (Liang et al., 2026). In contrast, PACPO moves reward-wise preservation from advantage estimation to the clipping stage. Through polarity-aware clipping, it keeps opposing reward contributions active, thereby mitigating cross-reward conflict while preserving the net advantage within a single-policy objective.

6 Conclusion

In this paper, we identify clipping-boundary gradient masking in multi-reward policy optimization, where aggregating reward-wise advantages before clipping can suppress corrective signals from opposing-polarity contributions. We propose PACPO, which reconstructs reward-wise contributions on a shared scale, controls the retained disagreement between polarities, and clips positive and negative components separately. PACPO preserves the aggregate advantage while allowing otherwise masked corrective gradients to remain active beyond clipping boundaries. Experiments across tool calling, scientific question answering, and text summarization demonstrate consistent improvements over strong baselines.

AI USE STATEMENT

(This section is **required** and does not count toward the page limit.)

In this work, we used generative AI tools for providing feedback on ablation design, organizing experimental records, selecting illustrative cases for appendix analyses, and assisting with the presentation of experimental results. We have not used generative AI tools to generate synthetic datasets, execute experiments, produce numerical results, and the remaining required-disclosure tasks are not applicable to this work. Additionally, we used generative AI tools for language revisiting and translation, literature search. We have reviewed all AI-assisted work. All numerical results and illustrative cases were manually verified against the raw experimental records, and all suggested citations were checked against the original publications. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

(This section is **recommended** and does not count toward the page limit.)

This work studies multi-reward policy optimization using publicly available datasets and pretrained language models. It does not involve new human-subject studies or the collection of private user data.

REPRODUCIBILITY STATEMENT

(This section is **recommended** and does not count toward the page limit.)

We support reproducibility by providing the complete formulation of PACPO in Section 3 and the corresponding derivations and gradient analysis in Appendix A. The training configurations, model backbones, and optimization hyperparameters are summarized in Table 7, with additional ablation and diagnostic settings provided in Appendix B. Dataset construction, prompt templates, reward definitions, and evaluation protocols are documented in Appendix C.

REFERENCES

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.

Kehua Feng, Xinyi Shen, Weijie Wang, Xiang Zhuang, Yuqi Tang, Qiang Zhang, and Keyan Ding. Sciknoweval: Evaluating multi-level scientific knowledge of large language models. *arXiv preprint arXiv:2406.09098*, 2024.

Chang Gao, Chujie Zheng, Xiong-Hui Chen, Kai Dang, Shixuan Liu, Bowen Yu, An Yang, Shuai Bai, Jingren Zhou, and Junyang Lin. Soft adaptive policy optimization. *arXiv preprint arXiv:2511.20347*, 2025.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Dohyeong Kim, Mineui Hong, Jeongho Park, and Songhwai Oh. Conflict-averse gradient aggregation for constrained multi-objective reinforcement learning. In *International Conference on Learning Representations*, volume 2025, pp. 36498–36527, 2025.

Komal Kumar, Tajamul Ashraf, Omkar Thawakar, Rao Muhammad Anwer, Hisham Cholakkal, Mubarak Shah, Ming-Hsuan Yang, Phillip HS Torr, Fahad Shahbaz Khan, and Salman Khan.

Llm post-training: A deep dive into reasoning large language models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2026.

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, 2025.

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, 2023a.

Ziniu Li, Tian Xu, Yushun Zhang, Zhihang Lin, Yang Yu, Ruoyu Sun, and Zhi-Quan Luo. Remax: A simple, effective, and efficient reinforcement learning method for aligning large language models. *arXiv preprint arXiv:2310.10505*, 2023b.

Ruiming Liang, Yi Zhong, Yizhen Yuan, Yinan Zheng, Tianyi Tan, Tianyue Wang, Haiyun Guo, Jinqiao Wang, and Xianyuan Zhan. Don't mix rewards, mix policies: Policy decomposition and optimization for multi-reward rl. *arXiv preprint arXiv:2607.29246*, 2026.

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.

Yinhong Liu, Zhijiang Guo, Tianya Liang, Ehsan Shareghi, Ivan Vulić, and Nigel Collier. Aligning with logic: Measuring, evaluating and improving logical preference consistency in large language models. *arXiv preprint arXiv:2410.02205*, 2024.

Renze Lou, Kai Zhang, and Wenpeng Yin. Large language model instruction following: A survey of progresses and challenges. *Computational Linguistics*, 50(3):1053–1095, 2024.

Haoye Lu, Daniel Herman, and Yaoliang Yu. Multi-objective reinforcement learning: Convexity, stationarity and pareto optimality. In *The Eleventh International Conference on Learning Representations*, 2023.

Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. *Advances in neural information processing systems*, 35:2507–2521, 2022.

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.

Zijun Min, Bingshuai Liu, Ante Wang, Long Zhang, Anxiang Zeng, Haibo Zhang, and Jinsong Su. Orchestrating tokens and sequences: Dynamic hybrid policy optimization for rlvr. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 37782–37795, 2026.

Ramesh Nallapati, Bowen Zhou, Cicero Dos Santos, Çağlar Gulçehre, and Bing Xiang. Abstractive text summarization using sequence-to-sequence rnns and beyond. In *Proceedings of the 20th SIGNLL conference on computational natural language learning*, pp. 280–290, 2016.

Aviv Navon, Aviv Shamsian, Idan Achituve, Haggai Maron, Kenji Kawaguchi, Gal Chechik, and Ethan Fetaya. Multi-task learning as a bargaining game. *arXiv preprint arXiv:2202.01017*, 2022.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. In *Advances in neural information processing systems*, volume 35, pp. 27730–27744, 2022.

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.

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, et al. Toolllm: Facilitating large language models to master 16000+ real-world apis. In *International Conference on Learning Representations*, volume 2024, pp. 9695–9717, 2024.

Qwen Team. Qwen2.5: A party of foundation models! `https://qwenlm.github.io/blog/qwen2.5/`, September 2024.

Rajkumar Ramamurthy, Prithviraj Ammanabrolu, Kianté Brantley, Jack Hessel, Rafet Sifa, Christian Bauckhage, Hannaneh Hajishirzi, and Yejin Choi. Is reinforcement learning (not) for natural language processing: Benchmarks, baselines, and building blocks for natural language policy optimization. *arXiv preprint arXiv:2210.01241*, 2022.

Alexandre Rame, Guillaume Couairon, Corentin Dancette, Jean-Baptiste Gaya, Mustafa Shukor, Laure Soulier, and Matthieu Cord. Rewarded soups: towards pareto-optimal alignment by interpolating weights fine-tuned on diverse rewards. In *Advances in Neural Information Processing Systems*, volume 36, pp. 71095–71134, 2023.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. Gpqa: A graduate-level google-proof q&a benchmark. *arXiv preprint arXiv:2311.12022*, 2023.

Diederik M Roijers, Peter Vamplew, Shimon Whiteson, and Richard Dazeley. A survey of multi-objective sequential decision-making. *Journal of Artificial Intelligence Research*, 48:67–113, 2013.

Tanik Saikh, Tirthankar Ghosal, Amish Mittal, Asif Ekbal, and Pushpak Bhattacharyya. Scienceqa: a novel resource for question answering on scholarly articles: T. saikh et al. *International Journal on Digital Libraries*, 23(3):289–301, 2022.

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, Y.K. Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Vaishnavi Shrivastava, Ahmed H Awadallah, Vidhisha Balachandran, Shivam Garg, Harkirat Behl, and Dimitris Papailiopoulos. Sample more to think less: Group filtered policy optimization for concise reasoning. In *International Conference on Learning Representations*, volume 2026, pp. 98439–98462, 2026.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. Learning to summarize with human feedback. *Advances in neural information processing systems*, 33:3008–3021, 2020.

Xuansheng Wu, Wenlin Yao, Jianshu Chen, Xiaoman Pan, Xiaoyang Wang, Ninghao Liu, and Dong Yu. From language modeling to instruction following: Understanding the behavior shift in llms after instruction tuning. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 2341–2369, 2024.

Weikai Xu, Chengrui Huang, Shen Gao, and Shuo Shang. Llm-based agents for tool learning: A survey. *Data Science and Engineering*, 10(4):533–563, 2025.

Yunqiao Yang, Houxing Ren, Zimu Lu, Ke Wang, Weikang Shi, Aojun Zhou, Junting Pan, Mingjie Zhan, and Hongsheng Li. Probability-consistent preference optimization for enhanced llm reasoning. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 6435–6448, 2025.

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

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, 2025a.

Tao Zhang, Ziqian Zeng, Yuxiang Xiao, Huiping Zhuang, Cen Chen, James Foulds, and Shimei Pan. Genderalign: An alignment dataset for mitigating gender bias in large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 11293–11311, 2025b.

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.

Mingkang Zhu, Xi Chen, Bei Yu, Hengshuang Zhao, and Jiaya Jia. Stratified grpo: Handling structural heterogeneity in reinforcement learning of llm search agents. *arXiv preprint arXiv:2510.06214*, 2025.

A Analysis of Polarity-Aware Contribution Policy Optimization

This appendix derives the effect of contribution reconstruction, conflict retention, and polarity-wise clipping in Section 3. During actor optimization, sampled responses, reward weights, normalization statistics, and λ are fixed. We analyze the token policy surrogate before loss averaging.

A.1 Notation and shared-scale reconstruction

Let $i = 1, \ldots, I$ index prompts, $j = 1, \ldots, G$ their responses, and $t = 1, \ldots, |y_{i,j}|$ the tokens of response $y_{i,j}$. The reward index $k = 1, \ldots, K$ is a superscript (k) in r, A, X, and B, and a subscript in w_k and μ_k. The response index n in Section 2 becomes j when i is explicit.

Table 6: Notation used throughout the method analysis. Parentheses in (k) distinguish a reward index from a power.

Symbol	Meaning
$r_{i,j}^{(k)},\ A_{i,j}^{(k)}$	Reward and its group-normalized advantage for dimension k.
$w_k,\ X_{i,j}^{(k)}$	Reward weight and weighted advantage $w_k A_{i,j}^{(k)}$.
$\mu_k,\ D$	Batch mean of $X^{(k)}$ and the shared GDPO normalization denominator.
$B_{i,j}^{(k)},\ A_{i,j}^{\mathrm{G}}$	Reconstructed contribution and final normalized GDPO advantage.
$P_{i,j},\ N_{i,j},\ C_{i,j}$	Positive sum, nonpositive negative sum, and cancelable disagreement mass.
$\widetilde{P}_{i,j}, \widetilde{N}_{i,j}, \lambda$	Retained components and retention coefficient $\lambda \in [0, 1]$.
$\rho_{i,j,t}, \epsilon$	Token policy ratio and PPO clip width, with $0 < \epsilon < 1$.
$\Psi_\epsilon, \Gamma_{i,j,t}^{\mathrm{PACPO}}$	Standard PPO surrogate and the sum of retained polarity surrogates.

For $G \geq 2$ and $\epsilon_{\mathrm{adv}} > 0$, the advantage of response j is computed with i and k fixed; the mean and standard deviation range over $j' = 1, \ldots, G$ responses to the same prompt:

$$A_{i,j}^{(k)} = \frac{r_{i,j}^{(k)} - \mathrm{mean}_{j'}[r_{i,j'}^{(k)}]}{\mathrm{std}_{j'}[r_{i,j'}^{(k)}] + \epsilon_{\mathrm{adv}}}, \qquad X_{i,j}^{(k)} = w_k A_{i,j}^{(k)}. \tag{A.1}$$

Define $A_{i,j}^{\mathrm{sum}} = \sum_k X_{i,j}^{(k)}$. All batch statistics use the GDPO population and sample weights. With $\epsilon_{\mathrm{batch}} > 0$, let

$$\mu_k = \mathrm{mean}_{\mathcal{B}}[X^{(k)}], \qquad D = \mathrm{std}_{\mathcal{B}}[A^{\mathrm{sum}}] + \epsilon_{\mathrm{batch}} > 0, \qquad B_{i,j}^{(k)} = \frac{X_{i,j}^{(k)} - \mu_k}{D}. \tag{A.2}$$

The shared denominator D uses the GDPO standard-deviation convention and stabilizer.

Preservation of the aggregate advantage. Linearity of the common batch mean gives $\sum_k \mu_k = \mathrm{mean}_{\mathcal{B}}[\sum_k X^{(k)}] = \mathrm{mean}_{\mathcal{B}}[A^{\mathrm{sum}}]$. Since D is shared by every contribution,

$$\begin{aligned} \sum_{k=1}^{K} B_{i,j}^{(k)} &= \frac{\sum_k X_{i,j}^{(k)} - \sum_k \mu_k}{D} \\ &= \frac{A_{i,j}^{\mathrm{sum}} - \mathrm{mean}_{\mathcal{B}}[A^{\mathrm{sum}}]}{D} = A_{i,j}^{\mathrm{G}}. \end{aligned} \tag{A.3}$$

Thus the reconstructed contributions sum exactly to the GDPO advantage. □

The sign of $B_{i,j}^{(k)}$ gives the direction of reward k's contribution to the final batch-normalized advantage. Contribution conflict therefore means opposing signs among the reconstructed $B_{i,j}^{(k)}$.

A.2 Polarity grouping and controlled conflict retention

This subsection formalizes our idea of separating net preference from cancelable disagreement in the reward-wise contributions. By grouping contributions by polarity and retaining a λ-controlled

portion of the disagreement, we keep opposing contributions explicit while preserving the aggregate GDPO advantage. For one response, suppress (i, j) and write $A = A^{\mathrm{G}}_{i,j}$ and $B^{(k)} = B^{(k)}_{i,j}$. With $[z]_+ = \max(z, 0)$ and $[z]_- = \min(z, 0)$, define

$$P = \sum_k [B^{(k)}]_+, \qquad N = \sum_k [B^{(k)}]_-, \qquad A = P + N, \qquad C = \min(P, -N). \tag{A.4}$$

Here $P \geq 0$, $N \leq 0$, and $C \geq 0$; $|N| = -N$ is the negative component's magnitude.

Separating net preference from cancelable disagreement. We first show that

$$P = [A]_+ + C, \qquad N = [A]_- - C. \tag{A.5}$$

If $A \geq 0$, then $P \geq -N$, so $C = -N$, $P = A + C$, and $N = -C$. If $A < 0$, then $P < -N$, so $C = P$ and $N = A - C$. These cases establish Eq. A.5, including $A = 0$.

For a fixed $\lambda \in [0, 1]$, the retained components therefore satisfy

$$\begin{aligned} \widetilde{P} &= P - (1 - \lambda)C = [A]_+ + \lambda C, \\ \widetilde{N} &= N + (1 - \lambda)C = [A]_- - \lambda C. \end{aligned} \tag{A.6}$$

Since $[A]_+ \geq 0$, $[A]_- \leq 0$, and $\lambda C \geq 0$, the retained signs are preserved. Adding the expressions gives A, and taking their smaller magnitude gives $\lambda C + \min([A]_+, -[A]_-) = \lambda C$. Hence

$$\widetilde{P} + \widetilde{N} = A, \qquad \min(\widetilde{P}, -\widetilde{N}) = \lambda C. \tag{A.7}$$

Retention therefore preserves the net advantage and scales the cancelable disagreement to λC. □

At $\lambda = 0$, the retained components reduce to $([A]_+, [A]_-)$; $\lambda = 1$ retains the full pair (P, N).

A.3 Exact relation to aggregate PPO clipping

This subsection derives the exact effect of controlled conflict retention on PPO clipping. The retained disagreement leaves the surrogate unchanged within the clipping interval and introduces a correction of $\lambda C h_\epsilon(\rho)$ outside it, making its effect on the boundary response explicit. Let $\ell = 1 - \epsilon$ and $u = 1 + \epsilon$, with $0 < \epsilon < 1$. For $\rho > 0$, the standard PPO surrogate has the sign-specific form

$$\Psi_\epsilon(\rho, a) = \begin{cases} a \min(\rho, u), & a \geq 0, \\ a \max(\rho, \ell), & a < 0. \end{cases} \tag{A.8}$$

For $a \geq 0$, the minimum of ρa and $\mathrm{clip}(\rho, \ell, u)a$ selects $a \min(\rho, u)$. For $a < 0$, multiplication reverses the ordering and gives $a \max(\rho, \ell)$. At $a = 0$, the surrogate is zero.

Why two polarity components preserve reward-wise clipping. At a common ratio and clipping interval, the factors $\min(\rho, u)$ and $\max(\rho, \ell)$ are shared by all positive and all negative contributions, respectively. Hence

$$\begin{aligned} \sum_k \Psi_\epsilon(\rho, B^{(k)}) &= \min(\rho, u) \sum_k [B^{(k)}]_+ + \max(\rho, \ell) \sum_k [B^{(k)}]_- \\ &= \Psi_\epsilon(\rho, P) + \Psi_\epsilon(\rho, N). \end{aligned} \tag{A.9}$$

The two polarity components thus exactly represent reward-wise clipping before retention. □

A boundary correction determined by retained disagreement. To expose the dependence on λ, write $\Gamma_\lambda(\rho) = \Psi_\epsilon(\rho, \widetilde{P}) + \Psi_\epsilon(\rho, \widetilde{N})$; this is $\Gamma^{\mathrm{PACPO}}_{i,j,t}$ with the response and token indices suppressed. Define the distance of the ratio from the clipping interval as

$$h_\epsilon(\rho) = [\ell - \rho]_+ + [\rho - u]_+ = \begin{cases} \ell - \rho, & 0 < \rho < \ell, \\ 0, & \ell \leq \rho \leq u, \\ \rho - u, & \rho > u. \end{cases} \tag{A.10}$$

Then the exact surrogate identity is

$$\boxed{\Gamma_\lambda(\rho) = \Psi_\epsilon(\rho, A) - \lambda C h_\epsilon(\rho).} \tag{A.11}$$

To prove it, substitute Eq. A.6 into the sign-specific surrogate:

$$\begin{aligned}\Gamma_\lambda(\rho) &= ([A]_+ + \lambda C)\min(\rho, u) + ([A]_- - \lambda C)\max(\rho, \ell) \\ &= \Psi_\epsilon(\rho, A) - \lambda C\{\max(\rho, \ell) - \min(\rho, u)\}.\end{aligned} \tag{A.12}$$

Using $\max(\rho, \ell) = \rho + [\ell - \rho]_+$ and $\min(\rho, u) = \rho - [\rho - u]_+$, the difference in braces equals $h_\epsilon(\rho)$. This proves Eq. A.11 for every $\rho > 0$, including both boundaries. □

Equation A.11 identifies how retained disagreement shapes the clipping-boundary response. Inside $[\ell, u]$, the surrogate remains ρA. Outside this interval, the correction scales jointly with retained disagreement λC and ratio excursion $h_\epsilon(\rho)$. This links the reward-contribution structure to the boundary gradients derived next.

A.4 Boundary gradients and exact masking conditions

This subsection shows how retained disagreement changes the gradients outside the clipping interval and identifies the exact conditions under which aggregate GDPO clipping masks a correction. On these masked tokens, PACPO retains a nonzero log-ratio gradient of magnitude $\lambda C\rho$, directed toward the clipping interval. Differentiating the affine branches away from ℓ and u gives the two ratio derivatives

	$0 < \rho < \ell$	$\ell < \rho < u$	$\rho > u$
$\dfrac{\partial \Psi_\epsilon(\rho, A)}{\partial \rho}$	$[A]_+$	A	$[A]_-$
$\dfrac{\partial \Gamma_\lambda(\rho)}{\partial \rho}$	$[A]_+ + \lambda C$	A	$[A]_- - \lambda C$

(A.13)

Below ℓ, the positive component remains active; above u, the negative component remains active. At either clipping boundary, the adjacent columns give the left and right derivatives.

Additional gradient in the log-ratio coordinate. Let $v = \log \rho$. Since $\partial\rho/\partial v = \rho$, the difference in log-ratio derivatives is

$$\Delta g_\lambda(\rho) \equiv \frac{\partial \Gamma_\lambda}{\partial v} - \frac{\partial \Psi_\epsilon(\rho, A)}{\partial v} = \begin{cases} \lambda C\rho, & 0 < \rho < \ell, \\ 0, & \ell < \rho < u, \\ -\lambda C\rho, & \rho > u. \end{cases} \tag{A.14}$$

For the maximized surrogate and $\lambda C > 0$, the lower-side increment is positive and the upper-side increment is negative, pointing toward the clipping interval in the log-ratio coordinate. Both have magnitude $\lambda C\rho$. The minimized policy loss $-\Gamma_\lambda$ reverses these derivative signs.

When aggregate clipping actually masks a correction. For $\rho \in (0, \ell) \cup (\ell, u) \cup (u, \infty)$, define strict masking as a zero aggregate policy-surrogate derivative paired with a nonzero PACPO derivative on the same token. Because $\rho > 0$, this criterion is equivalent in the ratio and log-ratio coordinates. Eq. A.13 gives the necessary and sufficient conditions

$$\begin{aligned} &\text{lower masking:} \quad \rho < \ell, \quad A \le 0, \quad \lambda C > 0, \\ &\text{upper masking:} \quad \rho > u, \quad A \ge 0, \quad \lambda C > 0. \end{aligned} \tag{A.15}$$

Indeed, below ℓ the aggregate derivative $[A]_+$ vanishes exactly when $A \le 0$, and the PACPO derivative then equals λC. Above u the aggregate derivative $[A]_-$ vanishes exactly when $A \ge 0$, and the PACPO derivative then equals $-\lambda C$. Each retained derivative is nonzero exactly when $\lambda C > 0$. Inside the interval both derivatives equal A. These cases establish the equivalence. □

At $A = 0$ and $\lambda C > 0$, both exterior regions satisfy the masking criterion, while the interior remains neutral. On every strictly masked token, $\partial\Gamma_\lambda/\partial v = \Delta g_\lambda$, so the restored log-ratio gradient has magnitude $\lambda C\rho$.

A.5 Update procedure and implementation requirements

PACPO computes contributions once per rollout and reuses two retained polarity tensors throughout actor optimization:

1. Draw G responses for each prompt using $\pi_{\theta_{\text{old}}}$. Store the response masks and old log-probabilities, and evaluate the K rewards.
2. Compute the weighted reward advantages and shared-scale contributions using Eqs. A.1–A.2.
3. Form $P_{i,j}$, $N_{i,j}$, and $C_{i,j}$, and compute the retained components for λ. Detach and reuse them throughout the rollout's actor updates.
4. Evaluate $\rho_{i,j,t}$ from the current and old policy log-probabilities. Apply Ψ_ϵ to each retained component and add the two surrogates before the loss reduction.
5. Average valid-token terms within each response, then average over responses and prompts as in Eq. A.16. Include the common KL and entropy terms once and apply the optimizer update.

For I prompts and valid response lengths $L_{i,j} = |y_{i,j}| > 0$, the sampled loss corresponding to the main objective is

$$\widehat{\mathcal{L}}_{\text{PACPO}} = -\frac{1}{IG}\sum_{i=1}^{I}\sum_{j=1}^{G}\frac{1}{L_{i,j}}\sum_{t=1}^{L_{i,j}}\left(\Gamma_{i,j,t}^{\text{PACPO}} - \beta\mathcal{L}_{\text{KL},i,j,t} + \eta\mathcal{H}_{i,j,t}\right). \tag{A.16}$$

Here $L_{i,j}$ counts valid response tokens. Each response receives weight $1/(IG)$, and its valid tokens share that weight equally. PACPO and GDPO use this common reduction and the same gradient-accumulation scaling.

Numerical and degenerate cases. A sample-standard-deviation estimator uses at least two observations; the positive stabilizers keep denominators positive at zero variance. A reward constant within every prompt group yields zero group advantages. If A^{sum} is constant across the batch, centering gives $A^{\text{G}} = 0$; opposing centered contributions can still yield $C > 0$, recovering the balanced-conflict case above. Finite-precision checks verify the reconstruction and retention identities in Eqs. A.3 and A.7 to the chosen numerical tolerance.

B Experimental Details

B.1 Tool-Use Reasoning

We evaluate PACPO in three multi-reward post-training settings, including structured tool use, scientific multiple-choice question answering, and abstractive summarization. These settings involve different forms of reward interaction. Tool use combines task execution with structural and reasoning requirements. Scientific QA considers answer correctness and response-format compliance separately. Summarization jointly optimizes four quality dimensions and a length-related objective. Within each task, all compared methods use the same backbone, data split, reward functions, rollout budget, and evaluation protocol. All tool-calling results are averaged over five runs with standard deviations

Dataset. We use the processed RLLA dataset from ToolRL (Qian et al., 2026) for policy training. The training split contains 3,920 tool-use examples, each providing a dialogue context and a set of available API definitions. The model is required to generate a structured response containing a reasoning trace and, when necessary, tool invocations with valid arguments. For evaluation, we use 597 API-Bank examples (Li et al., 2023a) to assess next-call prediction across three task levels.

Reward design. The tool-use objective contains three equally weighted rewards. *Correctness* compares the predicted call with the reference tool name, parameter names, and parameter values. During training it is a shaped score in $[-3, 3]$, with malformed tool-call blocks receiving the minimum score. *Format* is a binary reward for the required ordering and validity of the `<think>`,

<tool_call>, and <response> blocks. *Length* encourages a sufficiently developed reasoning trace and is computed from the word count n_{think} inside the reasoning block:

$$r_{\text{len}} = \min\Big\{1, \text{round}\Big(\frac{n_{\text{think}}}{512}, 2\Big)\Big\}. \tag{A.17}$$

A missing reasoning block receives zero length reward.

Evaluation protocol. We evaluate all methods on the three API-Bank task levels using vLLM with a maximum generation length of 4,096 tokens. We report accuracy for each level and compute Correct Acc. over all evaluation examples. We also report Format Acc. and the mean Length Reward. Accuracy-based metrics are reported as percentages, while Length Reward retains its original $[0, 1]$ scale. The Overall score combines the three evaluation dimensions and is computed as

$$\text{Overall}_{\text{Api}} = \frac{\text{Correct}}{100} + \text{Length} + \frac{\text{Format}}{100}. \tag{A.18}$$

B.2 Scientific Multiple-Choice Question Answering

Dataset and task. The policy is trained on a curated 18,450-record SciKnowEval-v2 scientific-QA pool (Feng et al., 2024). Each prompt asks the model to reason about a multiple-choice science problem and return its final choice inside an <answer> block. Transfer is evaluated natural-science questions from ScienceQA (Saikh et al., 2022; Lu et al., 2022) and GPQA-Main (Rein et al., 2023). ScienceQA image-dependent questions are excluded, and GPQA choices are deterministically permuted per record before evaluation.

Reward design. Training uses two equally weighted binary rewards for correctness and format. The *correctness reward* evaluates whether the option extracted from the <answer> block matches the reference answer. The *format reward* verifies that the response contains a nonempty <think> block followed by an <answer> block with a single option label. These rewards evaluate answer correctness and format compliance separately.

Evaluation protocol. We evaluate all methods on ScienceQA and GPQA using greedy decoding. We report answer accuracy, format compliance, joint success, and Avg. Joint success requires both a correct answer and valid format on the same example. All methods use the same prompt format, answer extraction procedure, and displayed option order. For each benchmark, Avg. is the arithmetic mean of correctness, format compliance and joint success. The overall scientific-QA score is computed as

$$\text{Overall}_{\text{QA}} = \frac{\text{ScienceQA Avg.} + \text{GPQA Avg.}}{2}. \tag{A.19}$$

B.3 Multi-Dimensional Summarization

Dataset and task. We study abstractive summarization on CNN/DailyMail (Nallapati et al., 2016) using Qwen2.5-7B-Instruct. We sample 2,000 articles from the training split and evaluate all methods on the same held-out test set. The policy receives the source article and an instruction to generate a summary.

Reward design. Five equally weighted rewards score each generated summary. UniEval provides *coherence*, *consistency*, *fluency*, and *relevance*. Consistency and fluency are averaged over generated sentences, whereas coherence and relevance score the complete summary. The fifth reward encourages a reference-like compression ratio:

$$r_{\text{concise}} = \left[1 + \left(\frac{|a/b - \tau|}{16}\right)^2\right]^{-1}, \qquad \tau = 15.0094, \tag{A.20}$$

where a and b are the source-article and generated-summary character counts. This term rewards proximity to the target ratio rather than monotonically favoring shorter outputs.

Table 7: Training configurations across three experimental settings.

Setting	Tool use	Scientific QA	CNN/DailyMail
Batch size	512	128	64
Learning rate	1×10^{-6}	5×10^{-6}	5×10^{-7}
Responses per prompt	4	16	8
Maximum prompt length	2,048	512	4,096
Rollout temperature	1.0	1.0	1.0
Top-p	1.0	1.0	1.0
Maximum response length	1,024	1,024	1,000
Clipping parameters	$\epsilon = 0.2$	$\epsilon = 0.2$	$\epsilon = 0.2$
KL coefficient	$\beta = 0$	$\beta = 0.04$	$\beta = 0$
Reward weights	$(1, 1, 1)$	$(1, 1)$	$(1, 1, 1, 1, 1)$
λ	$\lambda = 0.15$	$\lambda = 0.15$	$\lambda = 0.15$

Evaluation protocol. We report all five reward dimensions and U-Mean, defined as the arithmetic mean of the four UniEval dimensions; conciseness is excluded from U-Mean so that the reported quality aggregate is not dominated by the explicit compression target. All methods use greedy decoding and the same 4,096/1,000-token prompt/completion limits. The CNN/DailyMail Overall score is the arithmetic mean of the five reported reward dimensions:

$$\text{Overall}_{\text{CNN}} = \frac{\text{Coh.} + \text{Cons.} + \text{Flu.} + \text{Rel.} + \text{Conc.}}{5}. \tag{A.21}$$

B.4 Implementation and Training Configuration

We conduct all experiments on 8 NVIDIA A800 GPUs, each with 80GB of memory. The key training configurations for tool use, scientific QA, and CNN/DailyMail summarization are summarized in Table 7. The backbone, batch size, rollout configuration, token limits, and learning rate are adjusted to the computational requirements of each setting but remain fixed across all compared methods within the same setting. Unless otherwise specified, all reward dimensions are equally weighted, and PACPO uses a conflict-retention coefficient of $\lambda = 0.15$. We apply the same clipping, KL regularization, and training budget to all methods within each setting to ensure a fair comparison.

B.5 Additional Details on Ablation Studies

Effect of λ. The conflict-retention coefficient $\lambda \in [0, 1]$ controls how much cancelable disagreement remains visible to polarity-specific clipping. Given positive and negative contributions $P \geq 0$ and $N \leq 0$, we define $C = \min(P, |N|)$ and obtain

$$\widetilde{P} = P - (1 - \lambda)C, \qquad \widetilde{N} = N + (1 - \lambda)C. \tag{A.22}$$

This preserves the net preference because $\widetilde{P} + \widetilde{N} = P + N$. When $\lambda = 0$, all cancelable disagreement is removed, leaving only the net preference. When $\lambda = 1$, the original positive and negative contributions are fully retained. Unlike GDPO, which clips their aggregate as $\Psi_\epsilon(\rho, P+N)$, PACPO clips them separately as

$$\Psi_\epsilon(\rho, P) + \Psi_\epsilon(\rho, N). \tag{A.23}$$

The two objectives agree within the clipping interval but differ after the relevant boundary is crossed, where PACPO can retain an opposite-polarity corrective signal suppressed by aggregate clipping. Table 8 presents a finer-grained sweep of λ on API-Bank. Although $\lambda = 0.30$ achieves the highest correctness, $\lambda = 0.15$ obtains the best overall score by maintaining competitive correctness and the highest format accuracy. These results further support moderate conflict retention. As shown in Table 8, $\lambda = 0.15$ provides the best balance between preserving such corrective information and limiting excessive cross-reward disagreement.

Polarity-specific clipping. We compare full PACPO with an aggregate-clipping variant on the two backbones reported in Table 5. Both variants use the same reward-wise contribution construction

Table 8: Effect of the conflict-retention coefficient λ on API-Bank.

λ	Correct.	Length	Format	Overall
0.00	61.31	1.00	99.50	2.608
0.10	61.58	1.00	99.65	2.612
0.15	62.92	1.00	**99.82**	**2.627**
0.20	62.92	1.00	95.64	2.586
0.50	61.45	1.00	99.20	2.607
1.00	61.85	1.00	99.66	2.615

Table 9: Detailed comparison of GDPO, conflict-attenuated GDPO (CA-GDPO), and PACPO on API-Bank across two backbones. CA-GDPO modifies the aggregate advantage using Eq. A.25 with $\alpha = 0.05$.

Model	Method	Level 1 Acc.	Level 2 Acc.	Level 3 Acc.	Correct	Length	Format	Overall
Qwen2.5-3B Instruct	GDPO	67.17	53.73	43.51	60.47	**99.92**	1.00	2.6039
	CA-GDPO	66.17	**58.21**	45.04	60.64	99.80	0.99	2.6027
	PACPO	**68.06**	51.51	**49.62**	**62.16**	99.91	1.00	2.6207
Llama-3.2-3B Instruct	GDPO	60.71	40.36	49.89	56.05	95.10	1.00	2.5115
	CA-GDPO	59.87	**47.34**	50.61	55.93	96.24	1.00	2.5217
	PACPO	**62.85**	46.33	**53.16**	**58.87**	**99.97**	1.00	2.5884

and conflict-retention coefficient $\lambda = 0.15$. The full model applies clipping separately as

$$\Psi_\epsilon(\rho, \widetilde{P}) + \Psi_\epsilon(\rho, \widetilde{N}), \tag{A.24}$$

whereas the ablated variant first aggregates the two components and applies a shared clipping response as $\Psi_\epsilon(\rho, \widetilde{P}+\widetilde{N})$. All other settings, including the data, reward weights, rollout configuration, training budget, and evaluation protocol, remain unchanged. This comparison therefore isolates the effect of polarity-specific clipping.

Scalar conflict attenuation. To determine whether the improvement of PACPO can be achieved by simply incorporating the conflict magnitude into the aggregate advantage, we construct a conflict-attenuated GDPO baseline. Specifically, we modify the aggregate advantage as

$$A^{\mathrm{CA}}_{i,j} = \mathrm{sign}\big(A^{\mathrm{G}}_{i,j}\big)\max\big(\big|A^{\mathrm{G}}_{i,j}\big| - \alpha C_{i,j}, 0\big)\,, \qquad \alpha = 0.05, \tag{A.25}$$

and apply standard aggregate clipping to $A^{\mathrm{CA}}_{i,j}$, while keeping all other settings unchanged. This baseline preserves the direction of the original aggregate advantage but reduces its magnitude according to the cancelable disagreement. As shown in Table 9, compared with GDPO, CA-GDPO produces mixed changes across different difficulty levels but does not improve the overall performance. This suggests that the cancelable disagreement contains useful information for adjusting policy updates, while directly attenuating the scalar advantage may also weaken useful net-preference signals. Although CA-GDPO and PACPO both incorporate $C_{i,j}$ while preserving the direction of the net preference, CA-GDPO still aggregates all contributions before clipping and therefore retains the shared boundary response. PACPO instead keeps the adjusted polarity components separate through clipping, allowing corrective gradients to remain active and achieving better overall performance.

B.6 Additional Details of the Clipping-Boundary Analysis

Figure 3 compares GDPO with PACPO under $\lambda = 1.0$ and $\lambda = 0.15$ using three token-level rates and two mean update diagnostics. All rates are computed over valid response tokens and exclude padding tokens. Figures 3(a)–(c) show the three rates over training, while Fig. 3(d) reports the mean KL divergence and token-local gradient change.

Outside-clip token rate. Figure 3(a) reports the proportion of tokens whose policy ratios fall outside the clipping interval $[1-\epsilon, 1+\epsilon]$. It measures how frequently the policy ratio crosses a clipping boundary, regardless of the advantage polarity or whether the clipped branch is ultimately activated. All three methods exhibit outside-clip tokens during training.

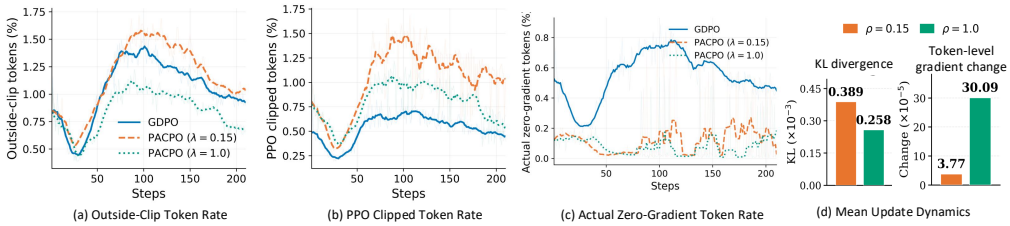


Figure 3: Clipping and token-level gradient dynamics on API-Bank.

Table 10: Mean training diagnostics for PACPO under different conflict-retention coefficients on API-Bank.

Metric	$\lambda = 0.15$	$\lambda = 1.0$
Outside-clip token rate (%)	1.126	0.844
Local gradient change ($\times 10^{-5}$)	3.77	30.09
Actor gradient norm	0.602	0.542
KL divergence ($\times 10^{-3}$)	0.389	0.258

Clipped token rate. Figure 3(b) reports the proportion of tokens that activate a clipped branch in the policy objective. This metric considers both the policy-ratio boundary and the update direction determined by the corresponding advantage or polarity component. GDPO exhibits the lowest clipped-token rate, whereas PACPO activates clipping separately for the two polarity components.

Actual zero-gradient token rate. Figure 3(c) reports the proportion of tokens whose token-local policy gradients are numerically zero. For GDPO, the gradient is computed from the clipped objective based on the aggregated advantage. For PACPO, it is computed after combining the separately clipped positive- and negative-polarity components. Therefore, clipping one PACPO component does not necessarily make the total token gradient zero, since the opposite-polarity component may remain active.

Mean gradient statistics

Local gradient change measures the absolute change in the token-level policy-gradient coefficient before and after the clipping-based gradient correction. A larger value indicates a stronger modification of the local optimization signal. This metric characterizes token-local gradients and is distinct from the parameter gradients of the entire actor network.

Actor gradient norm is the ℓ_2-norm of the gradients over all actor parameters after backpropagation. It reflects the overall magnitude of an actor update, with a larger value generally indicating a stronger parameter update.

KL divergence measures the distributional difference between the current policy and the reference policy. A larger KL divergence indicates a greater deviation from the reference policy, whereas a smaller value corresponds to a more conservative policy update.

Actual zero-gradient token rate is the proportion of valid response tokens whose final token-local policy-surrogate gradient is numerically zero after advantage construction, clipping, and loss computation. Specifically, a token is treated as zero-gradient when $|\partial \ell_t / \partial \log \pi_\theta(y_t \mid x, y_{<t})| \leq 10^{-12}$. A higher rate indicates that clipping completely suppresses the policy-gradient signals of more tokens.

Figure 3(d) further compares the mean token-local gradient change and KL divergence between the two PACPO settings. Full conflict retention at $\lambda = 1.0$ increases the gradient change from 3.77×10^{-5} to 30.09×10^{-5} while producing a lower KL divergence. This indicates that full retention more strongly reshapes token-local gradients, although it does not produce a larger average policy displacement in this run.

As shown in Table 10, $\lambda = 1.0$ produces a substantially larger change in the token-local surrogate gradient, together with a lower outside-clip token rate, actor gradient norm, and KL divergence. This result shows that retaining all cancelable disagreement more strongly reshapes local gradients and reduces the overall policy displacement. With $\lambda = 0.15$, PACPO retains part of the opposing-polarity contributions, moderating local gradient changes while preserving reward-specific corrective information. Its superior final performance demonstrates that moderate conflict retention better balances net-preference optimization and reward-specific optimization requirements.

Table 11: Effective token-gradient rates in the CIR and non-CIR regions on API-Bank. The difference is PACPO minus GDPO in percentage points.

Token region	GDPO	PACPO	Difference
CIR	47.57% $\pm$6.30	99.88% $\pm$0.11	+52.31 pp
Non-CIR	99.92% $\pm$0.18	99.92% $\pm$0.11	0.00 pp

B.7 Localized Gradient Recovery in the CIR Region

PACPO aims to recover corrective gradients that are suppressed when conflicting reward contributions are aggregated before PPO clipping. To examine whether this recovery occurs in the theoretically expected region, we define conflict-and-importance-ratio-out-of-range (CIR) tokens as those that contain both positive- and negative-polarity contributions and have policy ratios outside the clipping interval. We record 100 training steps of PACPO on API-Bank and compute both the PACPO gradient and the counterfactual native GDPO gradient on exactly the same tokens. Only the PACPO gradient is used for parameter updates, while the GDPO gradient is retained as a shadow diagnostic. This paired computation eliminates differences caused by separate rollouts or training trajectories. A gradient-recovery event occurs when the GDPO gradient is numerically zero but the PACPO gradient remains effective.

As shown in Table 11, the effective-gradient rate among CIR tokens is only 47.57% under the GDPO objective but reaches 99.88% under PACPO, an improvement of 52.31 percentage points. In contrast, both objectives achieve an effective-gradient rate of 99.92% on non-CIR tokens. We also observe no cases in which PACPO suppresses a gradient that remains effective under GDPO. Aggregating reward contributions before clipping causes the entire token-local gradient to vanish for approximately half of the CIR tokens. PACPO instead preserves nearly all effective gradients in this region by allowing the opposite-polarity component to remain active when the other component is clipped. Meanwhile, PACPO does not change whether gradients remain effective on non-CIR tokens. These results show that PACPO performs localized gradient recovery at conflict-and-clipping boundaries rather than globally amplifying policy gradients or uniformly weakening PPO clipping.

C Task Interfaces and Reward Definitions

C.1 Tool-use reasoning

Each API-Bank evaluation record provides a system message containing tool definitions and a user message containing the dialogue context. Both messages are passed to the policy through its chat interface. The system-message template below is taken from the processed API-Bank data. Example-specific tool definitions are replaced with a placeholder, and blank lines are compacted for display.

Tool use — Evaluation system-message template

```
You are a helpful multi-turn dialogue assistant capable of leveraging
    tool calls to solve user tasks and provide structured chat
   responses.
**Available Tools**
In your response, you can use the following tools:
{{ AVAILABLE_TOOL_DEFINITIONS }}
**Steps for Each Turn**
```

```
1. **Think:** Recall relevant context and analyze the current user
    goal.
2. **Decide on Tool Usage:** If a tool is needed, specify the tool
    and its parameters.
3. **Respond Appropriately:** If a response is needed, generate one
    while maintaining consistency across user queries.
**Output Format**
```plaintext
<think> Your thoughts and reasoning </think>
<tool_call>
{"name": "Tool name", "parameters": {"Parameter name": "Parameter
    content", "... ...": "... ..."}}
{"name": "... ...", "parameters": {"... ...": "... ...", "... ...":
    "... ..."}}
...
</tool_call>
<response> AI's final response </response>
```
Important Notes
1. You must always include the `<think>` field to outline your
    reasoning. Provide at least one of `<tool_call>` or `<response>`.
     Decide whether to use `<tool_call>` (possibly multiple times),
    `<response>`, or both.
2. You can invoke multiple tool calls simultaneously in the `<
    tool_call>` fields. Each tool call should be a JSON object with a
     "name" field and an "parameters" field containing a dictionary
    of parameters. If no parameters are needed, leave the "parameters
    " field an empty dictionary.
3. Refer to the previous dialogue records in the history, including
    the user's queries, previous `<tool_call>`, `<response>`, and any
     tool feedback noted as `<obs>` (if exists).
```

C.1.1 DIALOGUE CONTEXT AND SCORING

The user message contains the original request, preceding assistant turns, tool observations, and the current request. The schematic below illustrates the roles of these fields. The original message is read directly from the dataset without rewriting.

Tool use — Input-context schema (not a model output)

```
{{ DIALOGUE_HISTORY }}
  <user> {{ PREVIOUS_USER_REQUEST }} </user>
  {{ PREVIOUS_ASSISTANT_TURN }}
  <obs> {{ TOOL_OBSERVATION }} </obs>
  ...
<user> {{ CURRENT_USER_REQUEST }} </user>
```

Training uses the three equally weighted correctness, format, and reasoning-length rewards described in the experimental setup. Evaluation uses a separate parser. A prediction is considered correct if any parsed candidate matches the reference tool name and parameter dictionary. Correctness is averaged over records with scoreable reference calls, while format and length are averaged over all evaluated records. The evaluation format validator checks whether the required blocks are balanced, non-repeated, and correctly ordered. It does not require the exact newline patterns used by the training scorer.

C.1.2 CONCRETE API-BANK CASE

API-Bank `Level3_2` — One pending email reminder

```
Available tools:
  QueryMeeting(user_name)
  EmailReminder(content, time, location, recipient)

Dialogue state:
  QueryMeeting returned two meetings for John:
  (1) client meeting, Room 1, 2021-01-01 10:00:00.
  (2) new-project meeting, Room 2, 2021-01-02 10:00:00.
  An EmailReminder for meeting (1) has already succeeded.

Current user request:
  "Based on our conversation above, please only make one tool call
to solve my need."

Reference next call:
  EmailReminder(
    content="Meeting about the new project will be held at Room 2.
Please be there on time.",
    time="2021-01-02 10:00:00",
    location="Room 2",
    recipient="john@example.com"
  )

GDPO output excerpt (correctness=0, format=1, length=1.0):
  [702-word reasoning omitted.]
  <response>The previous tool calls have already retrieved the
meeting details for John and sent an email reminder to
John@example.com. However, the available tools do not provide a
single tool that can perform both tasks.</response>
  [No <tool_call> block was emitted.]

GD2PO-Hard parsed call (correctness=0, format=1, length=1.0):
  EmailReminder(
    content="Meeting with the client will be held at Room 1. Please
be there on time. Your email reminder has been sent successfully.",
    time="2021-01-01 10:00:00",
    location="Room 1",
    recipient="john@example.com"
  )

PACPO (lambda=0.15) parsed call (correctness=1, format=1,
length=1.0):
  EmailReminder(
    content="Meeting about the new project will be held at Room 2.
Please be there on time.",
    time="2021-01-02 10:00:00",
    location="Room 2",
    recipient="john@example.com"
  )
```

For comparison, we present one held-out API-Bank example. All methods satisfy the format and length criteria, but only PACPO selects the still-pending meeting and correctly fills all four reference arguments.

Figure 4 presents the training reward trajectories for correctness, format, and length across three backbones. All methods gradually converge as training proceeds, while correctness exhibits clearer differences than the other two rewards. PACPO maintains stable optimization and achieves competitive or higher correctness rewards on most backbones. Meanwhile, its format and length rewards

converge close to their respective upper bounds, indicating that improvements in correctness do not destabilize the other reward dimensions.

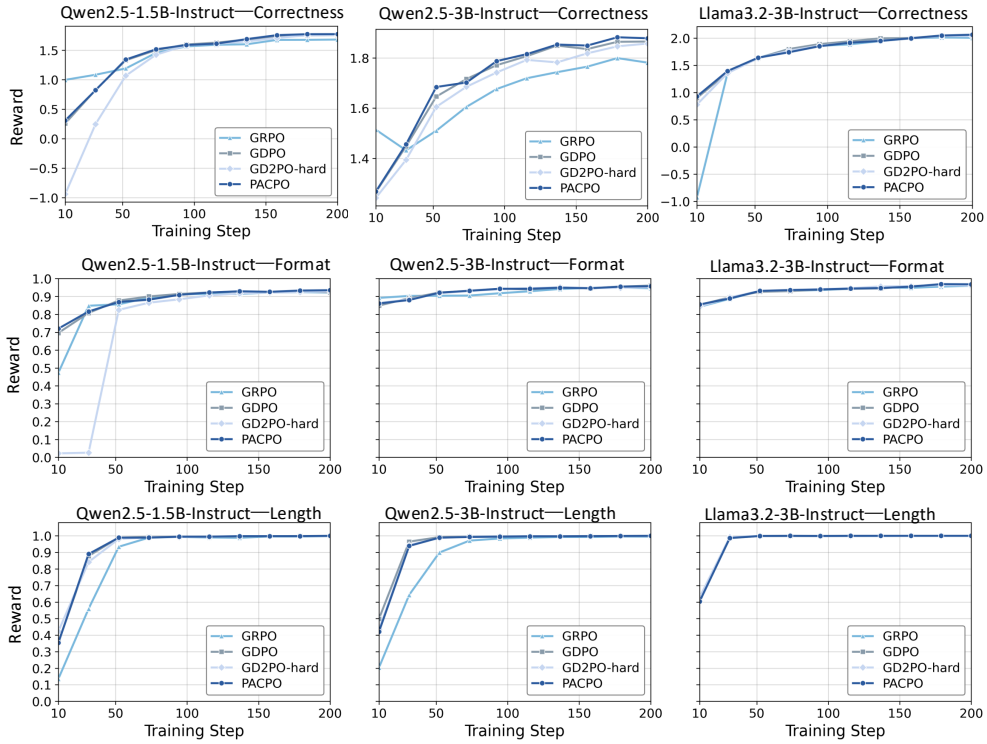


Figure 4: Training reward trajectories on API-Bank across three backbones.

C.2 Scientific Multiple-Choice Question Answering

The scientific-QA pipeline uses a two-message prompt: a fixed system instruction followed by a user message containing the question and answer choices.

Scientific QA — System message

```
You are a helpful AI assistant specialized in solving multiple-choice
     science questions.

For every request, solve the problem carefully and step by step
    before giving the final answer.

You must respond in exactly the following format:

<think>
Provide a clear, step-by-step solution.
</think>
<answer>
Provide only the final answer.
</answer>

Rules:

1. The <think> section must contain the complete reasoning process.
```

```
2. The <answer> section must appear after </think>.

3. The <answer> section must contain only one option letter: A, B, C,
    or D.

4. Do not include explanation, extra words, or additional formatting
   in <answer>.

5. Always complete the reasoning before writing the final answer.
```

Scientific QA — User-message template

```
Given a question and four options, please select the right answer.
Your answer should be "A", "B", "C" or "D".

{question}

Choices:
A. {choice_a}
B. {choice_b}
C. {choice_c}
D. {choice_d}
```

C.2.1 ANSWER EXTRACTION AND EVALUATION

The correctness parser searches sequentially for an answer tag, a boxed answer, an answer/option/choice phrase, and an isolated option letter within the final 256 characters. At the first successful stage, it selects the last match and compares the uppercased letter with the reference answer. The format parser requires a nonempty `<think>` block followed by an `<answer>` block containing only one letter from A–D. An answer may therefore be correct without satisfying the strict format requirement. For GPQA, the answer options are shuffled deterministically for each record using seed zero.

We sample one ScienceQA example from the test-set evaluation for qualitative comparison. PACPO produces a complete reasoning trace and a correctly formatted answer that agrees with the reference.

Question and displayed option order.

ScienceQA `test:3150` — Evaluation input

```
What do these two changes have in common?
mixing sand and water
sediment settling to the bottom of a muddy puddle

A. Both are chemical changes.
B. Both are caused by heating.
C. Both are caused by cooling.
D. Both are only physical changes.

Reference answer: D
```

Verbatim PACPO output.

PACPO — Greedy completion

```
<think>
To determine which option correctly describes the nature of the
given changes, let's analyze each one:

1. Mixing sand and water:
```

```
   - This is a physical change because the components (sand and
water) remain distinct and can be separated without changing their
fundamental properties.

2. Sediment settling to the bottom of a muddy puddle:
   - This is also a physical change because the sediment (which is
essentially small particles of sand and clay) settles due to
gravity, but the overall composition and state of the mixture (muddy
puddle) remains unchanged.

Now, let's evaluate the options:
A. Both are chemical changes: This is incorrect because neither
mixing nor settling involves any chemical reaction that alters the
substances involved.
B. Both are caused by heating: Neither mixing nor settling requires
heating to occur.
C. Both are caused by cooling: Neither mixing nor settling requires
cooling to occur.
D. Both are only physical changes: This is correct because both
processes involve no alteration of the substances' chemical
identities and simply rearrange them.

</think>
<answer>
D
</answer>
```

Extracted	Correctness	Strict format	Joint	Reasoning words	Image required
D	1	1	1	165	No

The answer parser extracts `D`, and the independent strict-format validator accepts the complete `<think>`–`<answer>` structure. This case illustrates the evaluated interface and scoring channels; it is not used as a causal comparison with another optimization method.

Scientific question answering. Figure 5 reports the correctness and format reward trajectories for scientific question answering across two backbones. Correctness progressively improves with moderate fluctuations, particularly during the later stages of training. In contrast, the format reward rapidly approaches saturation and remains stable thereafter. PACPO follows a stable optimization trajectory and achieves competitive correctness rewards while preserving near-perfect format compliance on both backbones.

C.3 Multi-dimensional summarization on CNN/DailyMail

The experiment uses CNN/DailyMail 3.0.0. The policy receives the source article through the template below. The article and reference summary are provided separately to the reward functions.

CNN/DailyMail — Policy prompt template

```
Text: {{ ARTICLE }}
Instruction: Summarize the Text without any Explanation.
Output:
```

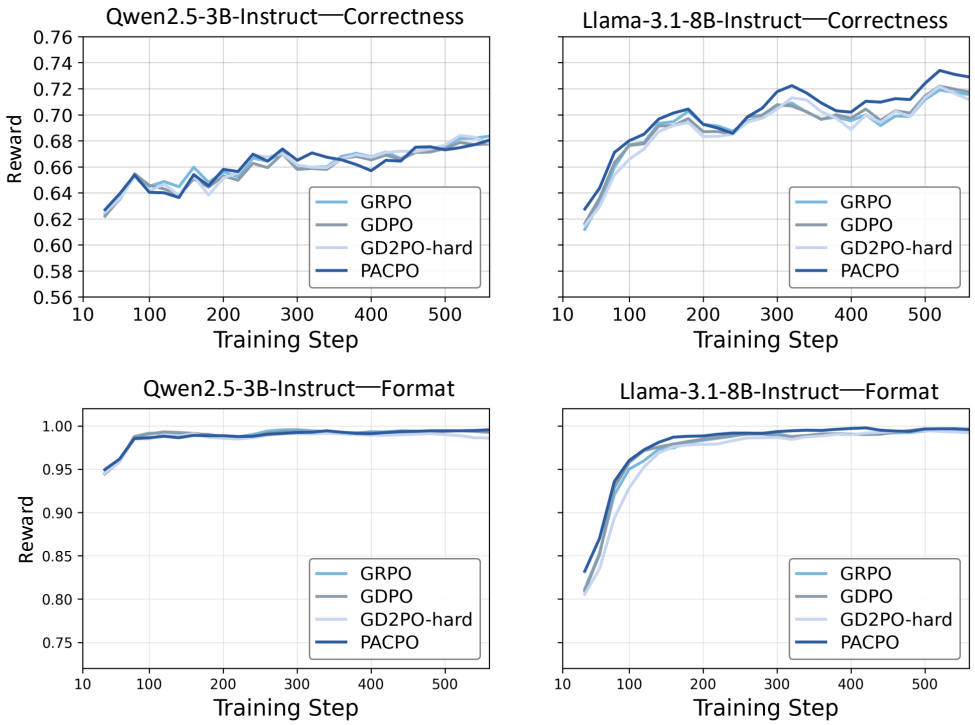


Figure 5: Training reward trajectories for scientific question answering with Qwen2.5-3B-Instruct and Llama-3.1-8B-Instruct.

C.3.1 Concrete CNN/DailyMail case

CNN/DailyMail `test/6427` — Obama in Jamaica and U.S.-Cuba relations

```
Input article excerpt (lightly normalized for display):
  President Barack Obama proved he is very popular in Jamaica on
Thursday. Crowds across the country cheered for President Obama as
he made multiple stops and became the first American president to
visit the island nation since President Ronald Reagan in 1982. It
seems that no one was more excited than Jamaican Prime Minister
Portia Simpson Miller. [The archived article continues. The complete
article contains 1,418 words.]

Reference summary (punctuation normalized for display):
  President Barack Obama became the first president to visit Jamaica
since President Ronald Reagan in 1982 on Thursday. Prime Minister
Portia Simpson Miller gushed over President Obama, at one point
telling him "I love you." President Obama signaled Thursday while
speaking in Jamaica that he will soon remove Cuba from the U.S. list
of state sponsors of terrorism. This came as he prepared to have
dinner with Cuban President Raul Castro at the Summit of the
Americas in Panama. The two shared a handshake at Nelson Mandela's
funeral in 2013, which angered some Americans. Secretary of State
John Kerry met Cuban Foreign Minister Bruno Rodriguez in a Panama
City hotel.

GRPO:
  President Barack Obama visited Jamaica and became the first U.S.
president to do so since Ronald Reagan in 1982. He was warmly
```

```
received by Jamaican citizens and officials, including Prime
Minister Portia Simpson Miller, who expressed her country's love for
Obama and asked him to consider exonerating Marcus Garvey, a
Jamaican national hero. During his visit, Obama also signaled that
he would soon remove Cuba from the U.S. list of state sponsors of
terrorism, a move that could improve relations between the two
countries.

GDPO:
  President Barack Obama visited Jamaica and became the first
American president to do so since Ronald Reagan in 1982. He was
warmly received by the Jamaican people and Prime Minister Portia
Simpson Miller, who expressed her country's love for him and asked
him to consider exonerating Marcus Garvey, a Jamaican national hero.
Additionally, Obama signaled during his visit that he will soon
remove Cuba from the U.S. list of state sponsors of terrorism, which
is expected to improve relations between the two countries.

GD2PO-Hard:
  President Barack Obama made a historic visit to Jamaica, becoming
the first U.S. president to do so since 1982, and expressed his
intention to remove Cuba from the U.S. list of state sponsors of
terrorism, signaling improved relations between the two countries.

PACPO :
  President Barack Obama visited Jamaica and became the first
American president to do so since Ronald Reagan in 1982. During his
visit, he met with Jamaican Prime Minister Portia Simpson Miller and
other Caribbean leaders. Prime Minister Simpson Miller expressed her
country's love for Obama and asked him to exonerate Marcus Garvey, a
Jamaican national hero. Obama also signaled that he will soon remove
Cuba from the U.S. list of state sponsors of terrorism, which is
expected to improve relations between the two countries.
```

We sample one CNN/DailyMail example for qualitative comparison. PACPO achieves the highest U-Mean and performs best in coherence, consistency, and relevance. It also maintains comparable fluency and matches the highest conciseness score among the compared methods.

CNN/DailyMail. Figure 6 shows the training dynamics of the five summarization rewards. Coherence and consistency generally improve throughout training, while fluency remains stable at a high level. Relevance increases during the early and middle stages before moderately declining near the end. Conciseness exhibits larger fluctuations across methods, reflecting the sensitivity of the compression objective. PACPO nevertheless maintains balanced and stable trajectories across all five reward dimensions.

Limitations. PACPO is currently developed and evaluated within PPO-style multi-reward policy optimization. Its extension to other policy objectives, longer-horizon interactive tasks, and broader model scales remains to be explored. Future work will examine the generality of polarity-aware optimization in these settings.



Figure 6: Training reward trajectories on CNN/DailyMail with Qwen2.5-7B-Instruct across coherence, consistency, fluency, relevance, and conciseness.