

COPO: CONSISTENCY-AWARE POLICY OPTIMIZATION

Anonymous authors

Paper under double-blind review

ABSTRACT

Reinforcement learning has significantly enhanced the reasoning capabilities of Large Language Models (LLMs) in complex problem-solving tasks. Recently, the introduction of DeepSeek R1 has inspired a surge of interest in leveraging rule-based rewards as a low-cost alternative for computing advantage functions and guiding policy optimization. However, a common challenge observed across many replication and extension efforts is that when multiple sampled responses under a single prompt converge to identical outcomes, whether correct or incorrect, the group-based advantage degenerates to zero. This leads to vanishing gradients and renders the corresponding samples ineffective for learning, ultimately limiting training efficiency and downstream performance. To address this issue, we propose a consistency-aware policy optimization framework that introduces a structured global reward based on outcome consistency, the global loss based on it ensures that, even when model outputs show high intra-group consistency, the training process still receives meaningful learning signals, which encourages the generation of correct and self-consistent reasoning paths from a global perspective. Furthermore, we incorporate an entropy-based soft-blending mechanism that adaptively balances local advantage estimation with global optimization, enabling dynamic transitions between exploration and convergence throughout training. Our method introduces several key innovations in both reward design and optimization strategy. We validate its effectiveness through substantial performance gains on multiple mathematical reasoning benchmarks, highlighting the proposed framework's robustness and general applicability. The code for this work has been open-sourced.

1 INTRODUCTION

Deepseek R1 Guo et al. (2025) has demonstrated remarkable potential of Reinforcement Learning (RL) in enhancing the reasoning capabilities of Large Language Models (LLMs) Radford et al. (2018); Achiam et al. (2023); Bai et al. (2023); Touvron et al. (2023); Liu et al. (2024) when tackling complex tasks such as mathematical problem solving and code generation. Previous RL applications Song et al. (2024); Ji et al. (2023a;b) based on methods such as Proximal Policy Optimization (PPO) Schulman et al. (2017), Direct Policy Optimization (DPO) Rafailov et al. (2023), and Reinforcement Learning Human Feedback (RLHF) Christiano et al. (2017), which primarily focus on aligning model’s responses with human preferences. To better support LLMs in the exploration and prioritization of optimal reasoning paths (Chain-of-Thought, CoT Wei et al. (2022)) during training, recent works such as Qwen2.5 Yang et al. (2024) and DeepSeek R1 have shifted their attention toward outcome-based reward mechanisms and have emphasized the potential of leveraging group-relative advantage (GRA) Shao et al. (2024a) strategies for effective policy optimization.

However, despite the remarkable practical effectiveness demonstrated by these works, a growing body of studies Yu et al. (2025); Liu et al. (2025) has revealed inherent flaws in Group-relative Policy Optimization (GRPO)-based methods. Specifically, when an objective is either too trivial or too challenging for the current policy model, the reward distribution over the model's responses tends to converge, causing most relative advantages to collapse towards zero. This leads to gradient collapse and sample wastage, hindering effective optimization of the challenging objective.

054 DAPO Yu et al. (2025) attempts to mitigate this problem by employing dynamic batch-size sam-
 055 pling to improve training efficiency and stability. Nevertheless, it fails to fundamentally address the
 056 underlying sample wastage problem.

057 To tackle the above challenges, we propose a novel consistency-entropy-based policy optimization
 058 framework, **COP**O, that theoretically addresses the sample wastage and gradient vanishing problem
 059 under extreme samples observed in GRPO methods. Specifically, we introduce a structured global
 060 reward based on outcome consistency and a global optimization mechanism, and we incorporate
 061 an entropy-based soft-blending mechanism that adaptively balances local advantage estimation with
 062 global optimization. We not only demonstrate the performance improvement of COP over GRPO
 063 methods in mathematical reasoning tasks, but also conduct extensive ablation studies on various ex-
 064 isting improvements to GRPO training schemes, aiming to provide deeper insight into GRPO-based
 065 post-training methods for this domain. Our main contributions are summarized as follows:

- 066 • We analyze the problem of advantage vanishing in GRPO and propose a global advantage
 067 formulation to extract batch-level advantage signals, thereby enabling effective utilization
 068 of data samples that would otherwise be discarded due to vanishing advantages.
- 069 • We propose a novel consistency-entropy-based policy optimization method named COP, 070 introducing the concept of joint optimization across both intra-group and inter-group sam-
 071 ples to fully leverage available training data.
- 072 • We develop an entropy-aware soft-blending mechanism that adaptively balances global
 073 optimization and local optimization objectives throughout training.

075 2 PRELIMINARY

076 2.1 GROUP-RELATIVE POLICY OPTIMIZATION, GRPO

077 GRPO, as a policy optimization algorithm, adopts a more streamlined approach by leveraging
 078 reward-based advantage estimation. The core idea of GRPO is to eliminate the need for an ad-
 079 ditional value network by computing advantages through intra-group reward comparisons under the
 080 same input. Specifically, given an input prompt q , the old policy $\pi_{\theta_{\text{old}}}$ generates a set of G candidate
 081 output sequences: $\mathcal{O}_q = \{o_1, o_2, \dots, o_G\}$. These sequences are then evaluated by a task-specific re-
 082 ward function r_ϕ , designed according to the optimization objective, yielding a corresponding reward
 083 set: $\{r_1, r_2, \dots, r_G\}$. The direction of policy update is determined by the relative ranking of rewards
 084 within the group: samples receiving higher rewards than the group average are encouraged by in-
 085 creasing their likelihood under the policy, while those with below-average rewards are suppressed
 086 by reducing their associated policy probabilities.

087 From this, the advantage of GRPO is calculated as:

$$088 \hat{A}_i = \frac{r_i - \mu_r}{\sigma_r}, \quad (1)$$

089 where $\mu_r = \text{mean}(\{r_i\}_{i=1}^G)$, $\sigma_r = \mu_r = \text{std}(\{r_i\}_{i=1}^G)$. Substituting the new advantage \hat{A}_i , group
 090 B , and the responses $\{o_{i=1}^G\}$ sampled by the policy model into the objective function of the PPO,
 091 we can obtain the objective function of the GRPO:

$$092 J_{\text{GRPO}}(\theta) = \mathbb{E}_{q, \{o_i\} \sim \pi_{\theta_{\text{old}}}} \left[\frac{1}{G} \sum_{i=1}^G \frac{1}{|o_i|} \sum_{t=1}^{|o_i|} \min \left(\frac{\pi_\theta(o_{i,t} \mid q, o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t} \mid q, o_{i,<t})} \hat{A}_i, \text{clip}(\cdot) \hat{A}_i \right) - \beta \mathbb{D}_{\text{KL}}[\pi_\theta \parallel \pi_{\text{ref}}] \right]. \quad (2)$$

101 2.2 ADVANTAGE DEGENERATION AND GRADIENT VANISHING OF GRPO

102 The internal mechanism of GRPO, which relies on reward mean and variance to estimate the advan-
 103 tage function, exhibits an inherent fragility during training. By computing advantages based on the
 104 mean and standard deviation of rewards, GRPO encourages the model to shift its output distribution
 105 toward those that match the expectation. This training strategy inevitably leads to a gradual collapse
 106 of reward variance when a given prompt q becomes either too easy or too difficult relative to the
 107 current policy π_θ . Formally, for any group \mathcal{O}_q , by definition Equation 1, as $\text{Var}(r) \rightarrow 0$, we have

108 $std(r) \rightarrow 0$, and all $r_i \approx \bar{r}$, thus $A_i \approx 0$. As a direct consequence, the gradient of the GRPO objective
 109 vanishes: $\nabla_{\theta} L_{GRPO} \rightarrow 0$. The degeneration of advantages and subsequent gradient vanishing
 110 substantially reduces the contribution of affected samples to policy updates, leading to a notable
 111 decline in training efficiency. As training progresses, this phenomenon tends to intensify.

112 During GRPO training, we observe that challenging training examples frequently lead to all G roll-
 113 out trajectories producing incorrect answers, which results in the vanishing of the advantage signal.
 114 Figure 1 presents the distribution of reward lists per prompt when training the 3B model with the
 115 GRPO method. Notably, prompts for which all sampled answers are incorrect constitute the largest
 116 proportion, accounting for 56% of the training data. When including the all-correct cases, 59.9% of
 117 the training samples exhibit zero inter-group reward variance, implying that only 40.1% of the data
 118 contribute effective advantage signals during GRPO training.

119 To address sparse advantage signals, DAPO
 120 uses dynamic sampling to exclude all-1 or all-0
 121 reward samples. However, this approach leads
 122 to a significant waste of training samples, espe-
 123 cially in the above case of small-scale LLMs,
 124 where samples with all-0 accuracy make up the
 125 majority. Given the same amount of inference
 126 data, small LLMs under the DAPO training
 127 framework discard a large portion of samples,
 128 thereby slowing down the model’s performance
 129 improvement. We believe that samples with
 130 zero in-group advantage still hold value, as they
 131 can provide global perspectives on optimization
 132 directions that support the overall training of
 133 the model. Therefore, extracting the effective
 134 advantage signals from the data where advantages
 135 vanish in GRPO is crucial for further improving
 136 model performance.

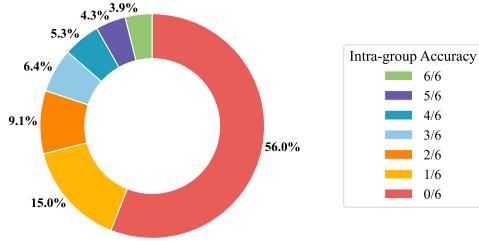


Figure 1: The distribution of intra-group accuracy for the Qwen2.5-3B-instruct model after 60 steps of GRPO training with rollout G=6. Over half of the problems yield all-zero outputs during inference.

3 COPO

139 In this paper, we proposed Consistency-Aware Policy Optimization(COPO), an RL framework that
 140 addresses the limitations of GRPO-like methods. Figure 2 shows the demonstration of COPO. To
 141 enable the effective use of samples with high consistency that would otherwise yield vanishing gradi-
 142 ents under group-relative training, the COPO framework calculates global rewards at the batch level
 143 and yields inter-group loss. Moreover, COPO introduces a consistency-entropy-based hybrid mech-
 144 anism to effectively integrate intra-group local optimization with inter-group global optimization to
 145 guide model updates.

146 Specifically, given a batch of prompts $Q = \{q_1, q_2, \dots, q_B\}$, the training objective of COPO is
 147 defined as:

$$J_{\text{COPO}}(\theta) = \mathbb{E}_{q \sim \mathcal{D}} \left[w(H_q) \cdot \mathcal{L}_{\text{local}}(q) + (1 - w(H_q)) \cdot \mathcal{L}_{\text{global}}(q) \right], \quad (3)$$

150 where $\mathcal{L}_{\text{local}}$ denotes the local policy loss, $\mathcal{L}_{\text{global}}$ denotes the global policy loss and $w \in (0, 1)$ is
 151 an entropy-based blending weight that adjusts the relative importance of two optimization. In the
 152 following subsections, we will describe each component of COPO in detail.

3.1 INTRA-GROUP LOCAL OPTIMIZATION

156 As shown in the upper part of Figure 2, the intra-group local optimization approach follows the
 157 principles of GRPO, where rewards and advantages are computed based on responses to one prompt.
 158 For each generated response, the local reward is calculated by the rule-based reward function $R(\cdot)$
 159 mentioned in Equation 13. The local optimization objective is expressed as:

$$J_{\text{local}}(\theta) = \mathbb{E}_{q, \{o_i\} \sim \pi_{\theta_{\text{old}}}} \left[\frac{1}{\sum_{i=1}^G |o_i|} \sum_{i=1}^G \sum_{t=1}^{|o_i|} \min \left(\frac{\pi_{\theta}(o_{i,t} | q, o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t} | q, o_{i,<t})} \hat{A}_{o_i}^{\text{local}}, \text{clip}(\cdot) \hat{A}_{o_i}^{\text{local}} \right) \right], \quad (4)$$

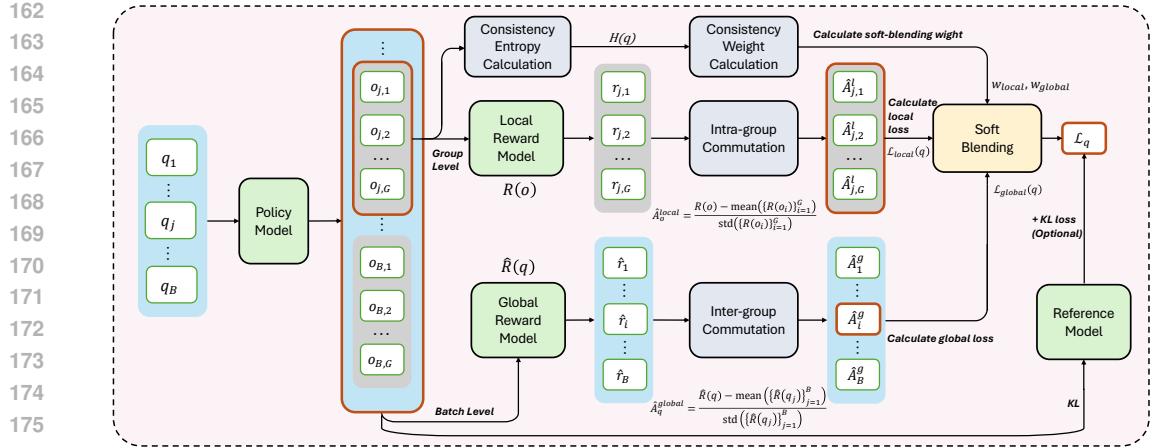


Figure 2: Demonstration of our COPO methods. COPO incorporates global optimization into the GRPO foundation to mitigate gradient vanishing caused by intra-group consistency.

where

$$\hat{A}_o^{\text{local}} = \frac{R(o) - \text{mean}(\{R(o_i)\}_{i=1}^G)}{\text{std}(\{R(o_i)\}_{i=1}^G)} \quad (5)$$

3.2 INTER-GROUP GLOBAL OPTIMIZATION

When reasoning outcomes are consistently all correct or all incorrect, the group-relative local objective collapses, causing vanishing advantages and ineffective gradients. To address this, we introduce inter-prompt global optimization, using cross-prompt reward variability to drive updates even when local signals vanish.

Given a prompt q , we sample G responses $o_{1:G} \sim \pi_\theta(\cdot \mid q)$, and define a prompt-level reward function $\hat{R}(q)$. Our goal is to optimize the policy such that it increases the likelihood of all sampled tokens in proportion to the prompt-level reward.

Under the framework of Proximal Policy Optimization(PPO), our objective remains to maximize the expected return of all sampled tokens, which is the same as intra-group local optimization. PPO calculates advantages based on Generalized Advantage Estimation(GAE), while advantage functions in traditional RL are typically computed as: $A(s_t, a_t) = G_t - V(s_t)$, where G_t denotes the cumulative return from timestep t , and $V(s_t)$ is the estimated value function. Because training an additional value head is computationally expensive, we drop it and approximate $\hat{A}_i = \hat{R}(q) - b$, where we treat $\hat{R}(q) = \frac{1}{G} \sum_{i=1}^G r_i$ as the return G_t to quantify the model's performance on prompt q , and use a baseline b as a surrogate for the value function. A fixed constant baseline cannot track the reward shift that occurs during training. Instead, we use the mean reward of the current mini-batch as b : $b \approx \mathbb{E}_{q \sim \mathcal{B}}[\hat{R}(q)]$.

In order to keep the local and global gradient magnitudes close to each other and avoid oscillations or mode collapse, we apply standardization so that the way to calculate global advantage is the same as Equation 1. The global advantage is ultimately computed as:

$$\hat{A}_q^{\text{global}} = \frac{\hat{R}(q) - \text{mean}(\{\hat{R}(q_j)\}_{j=1}^B)}{\text{std}(\{\hat{R}(q_j)\}_{j=1}^B)}, \text{ for } \forall o_i \in \mathcal{O}_q, \quad (6)$$

where $\text{mean}(\{\hat{R}(q_j)\}_{j=1}^B)$ and $\text{std}(\{\hat{R}(q_j)\}_{j=1}^B)$ are the mean and standard deviation of prompt-level rewards within the current mini-batch. The global optimization objective is expressed as:

$$J_{\text{global}}(\theta) = \mathbb{E}_{q, \{o_i\} \sim \pi_\theta} \left[\frac{1}{\sum_{i=1}^G |o_i|} \sum_{i=1}^G \sum_{t=1}^{|o_i|} \min \left(\frac{\pi_\theta(o_{i,t} \mid q, o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t} \mid q, o_{i,<t})} \hat{A}_q^{\text{global}}, \text{clip}(\cdot) \hat{A}_q^{\text{global}} \right) \right]. \quad (7)$$

216 While this formulation bears superficial resemblance to local advantage computation, the semantics
 217 are fundamentally different. Here, both $\text{mean}(\{\hat{R}(q_j)\}_{j=1}^B)$ and $\text{std}(\{\hat{R}(q_j)\}_{j=1}^B)$ are calculated
 218 from different actions and states; they can be viewed as trajectory-independent constants when the
 219 gradient is taken, which do not introduce bias in the policy gradient.
 220

221 In the local case, samples $o_{1:G}$ within the same prompt q share the same state, and the difference
 222 $R(o) - \text{mean}(\{R(o_i)\}_{i=1}^G)$ reflects a relative ranking among actions in that specific state. As a
 223 result, the gradient explicitly pushes the model to shift probability mass from less preferred incor-
 224 rect responses toward higher-rewarding responses. In contrast, global optimization operates across
 225 different prompts q_1, q_2, \dots , each representing a distinct state. The rewards $\hat{R}(q_j)$ are therefore
 226 not semantically comparable. The mean reward $\text{mean}(\{\hat{R}(q_j)\}_{j=1}^B)$ functions purely as a baseline
 227 to normalize the learning signal across diverse environments. Importantly, this does not cause the
 228 model to shift probability from actions in complex prompts toward those in simpler prompts. This
 229 is because the gradient in policy optimization still applies locally at each state-action pair (s, a) ,
 230 and a constant baseline across prompts is treated as a variance-reducing term in the policy gradient,
 231 without altering the expected optimization direction.
 232

233 Intuitively, the global advantage function evaluates the model’s performance across different
 234 prompts within the same batch by assigning rewards or penalties accordingly. Although the specific
 235 prompts vary from batch to batch, the global advantage consistently provides positive reinforcement
 236 to trajectories that are more likely to yield correct answers. In cases where all rollout trajectories
 237 associated with a prompt are correct, the global advantage assigns the highest level of positive re-
 238inforcement to strengthen such paths. In contrast, it applies negative reinforcement to prompts for
 239 which all trajectories are incorrect. By supplying accuracy-based reward signals to data instances
 240 where the local advantage is zero, the global advantage alleviates the issue of sample inefficiency
 241 that arises when relying solely on local advantage.
 242

243 3.3 ENTROPY-BASED SOFT BLENDING

244 While the global optimization strategy effectively mitigates the gradient vanishing problem inherent
 245 to local group-relative methods, it could introduce a new challenge: the global optimization assigns
 246 the same advantage value, derived from the prompt-level reward, to all sampled responses $o_i \in$
 247 \mathcal{O}_q . Consequently, lower-quality responses may undesirably receive higher advantages than they
 248 inherently merit, thereby weakening the precision of credit assignment and diluting learning signals
 249 from truly optimal responses. Therefore, the global optimization is more suitable for prompts with
 250 high response consistency.
 251

252 To address this trade-off, we propose adaptively selecting between local and global optimiza-
 253 tion strategies based on the consistency entropy of the current policy’s responses. Formally,
 254 given the set generated responses \mathcal{O}_q , the set of outcomes extracted from \mathcal{O}_q are defined as q :
 $T_q = \{\tau_1, \tau_2, \dots, \tau_k\}$, where k denotes the number of unique outcomes from \mathcal{O}_q .
 255

256 we define the consistency entropy as:
 257

$$H(q) = - \sum_{\tau \in T_q} p(\tau) \cdot \log p(\tau), p(\tau) = \frac{\text{count}(\tau)}{G}, \quad (8)$$

258 where $\text{count}(\tau)$ denotes the number of occurrences of τ . The consistency entropy evaluates the
 259 consistency of the model’s responses to a given prompt, serving as an indicator of the determinism
 260 in its output behavior.
 261

262 To ensure all samples participate in both global and local optimization paths without discarding any
 263 sample entirely, we propose a soft-blending mechanism that smoothly interpolates between the two
 264 objectives:
 265

$$\mathcal{L}_q = w_{\text{local}}(H(q)) \cdot \mathcal{L}_{\text{local}}(q) + w_{\text{global}}(H(q)) \cdot \mathcal{L}_{\text{global}}(q), \quad (9)$$

266 where the weighting functions are defined as:
 267

$$w_{\text{local}}(H) = \sigma(\gamma(H - \rho)), w_{\text{global}}(H) = 1 - w_{\text{local}}(H), \quad (10)$$

268 with $\sigma(\cdot)$ denoting the sigmoid function for smooth interpolation, γ as a temperature hyperparam-
 269 eter controlling the sharpness of transition, and ρ the central entropy threshold around which the
 270 optimization focus transitions.
 271

Algorithm 1 COPO Training

270
 271
 272 **Require:** Policy model π_θ , old policy $\pi_{\theta_{\text{old}}}$, local reward function $R(\cdot)$, global reward function $\hat{R}(\cdot)$,
 273 blending parameters (γ, ρ) , clip parameter ϵ , batch size B , samples per prompt G
 274 1: Initialize π_θ from pre-trained LM; copy $\pi_{\theta_{\text{old}}} \leftarrow \pi_\theta$
 275 2: **while** not converged **do**
 276 3: Sample a batch of prompts $\{q_1, \dots, q_B\} \sim \mathcal{D}$
 277 4: **for** each prompt q in batch **do**
 278 5: Sample G responses $\mathcal{O}_q = \{o_1, \dots, o_G\} \sim \pi_{\theta_{\text{old}}}(\cdot | q)$
 279 6: Compute final answers $T_q = \{\tau_1, \tau_2, \dots, \tau_k\}$ and entropy: (Equation 8)
 280 7: Compute blending weights: (Equation 10)
 281 8: Compute individual rewards $\{r_i\}_{i=1}^G$
 282 9: Compute group-level local advantage: (Equation 1)
 283 10: Compute batch-level global reward $\{\hat{r}_i\}_{i=1}^B$ for each prompt q
 284 11: Compute global advantage: (Equation 6)
 285 12: **for** each $o_i \in \mathcal{O}_q$, and token t **do**
 286 13: Update the policy model π_θ by maximizing the COPO objective: (Equation 11)
 287 14: **end for**
 288 15: **end for**
 289 16: Aggregate losses over all tokens in the batch and update π_θ using gradient descent
 290 17: Periodically update $\pi_{\theta_{\text{old}}} \leftarrow \pi_\theta$
 291 18: **end while**

292 Thus, when consistency entropy $H(q)$ is high, indicating high diversity in responses, the local op-
 293 timization dominates, encouraging the model to differentiate and reinforce higher-quality responses
 294 within the group. Conversely, when $H(q)$ is low, indicating high response uniformity, global op-
 295 timization dominates, pushing the model toward maintaining correctness and consistency across
 296 prompts. This mechanism enables each sample to adaptively determine its contribution intensity to
 297 both optimization pathways, mitigating potential pitfalls such as optimization precision loss result-
 298 ing from relying solely on global optimization, and diminishing advantage and vanishing gradients
 299 caused by exclusively employing local optimization. Accordingly, the COPO training procedure
 300 follows Algorithm 1, with the overall optimization objective formulated as:

301
 302
$$J_{\text{COPO}}(\theta) = \mathbb{E}_{q \sim \mathcal{D}} \left[\frac{1}{\sum_{i=1}^G |o_i|} \sum_{i=1}^G \sum_{t=1}^{|o_i|} \cdot \left(w(H_q) \cdot \min \left(r_{i,t}^{(q)}(\theta) A_{o_i}^{\text{Local}}, \text{clip}(\cdot) A_{o_i}^{\text{Local}} \right) \right. \right. \\ \left. \left. + (1 - w(H_q)) \cdot \min \left(r_{i,t}^{(q)}(\theta) \hat{A}_q^{\text{Global}}, \text{clip}(\cdot) \hat{A}_q^{\text{Global}} \right) \right) - \beta \mathbb{D}\text{KL} [\pi_\theta \| \pi_{\text{ref}}] \right], \quad (11)$$

 303
 304
 305
 306

307 where $r_{i,t}^{(q)}(\cdot) = \frac{\pi_\theta(o_{i,t} | q, o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t} | q, o_{i,<t})}$. This normalized advantage is then uniformly applied to all log-
 308 probabilities associated with prompt q .
 309

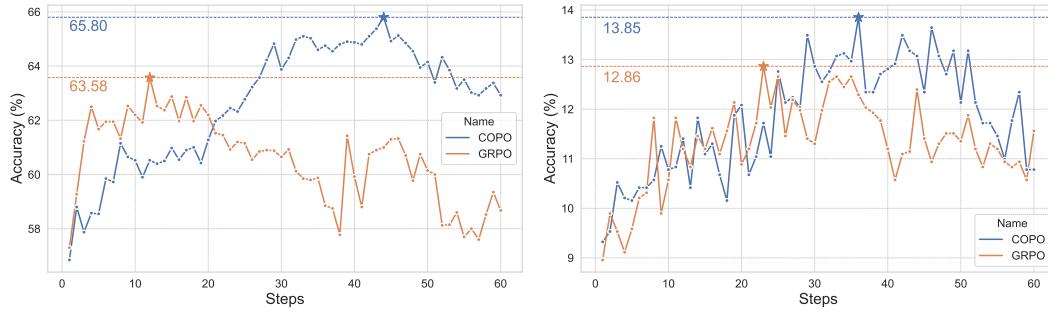
4 TRAINING

310
 311 To ensure fair comparisons, all experiments are conducted using the DAPO-MATH-17k Yu et al.
 312 (2025) dataset as the training set. Evaluation is performed on a suite of benchmarks, including
 313 MATH-500 Lightman et al. (2023), AIME 2024 Jia (2024), GSM8k Cobbe et al. (2021), and AIME
 314 2025 Lin (2025), which together span a broad range of mathematical reasoning difficulties. A rule-
 315 based reward incorporating solely correctness-based signals is employed as the reward model. All
 316 training and testing experiments are conducted through the VERL framework Sheng et al. (2024).
 317
 318

319 We sample 512 prompts per batch, generating 6 responses each. The data are split into 32 mini-
 320 batches for gradient updates. Both Qwen2.5-Instruct-3B and 7B are trained for 60 optimization
 321 steps. We adopt the AdamW optimizer with no weight decay and a constant learning rate of 1×10^{-6} .
 322 For the PPO clipping objective, we apply an asymmetric clipping strategy, setting $\epsilon = 0.2$. The
 323 maximum length for both prompt and generated response is set to 2048 tokens. During inference,
 we use nucleus sampling with temperature 1.0 and top-p 1.0.

324 Table 1: Comparison of GRPO, DAPO and our method across MATH-500 and AIME24 datasets.
325 We use mean@8 and maj@8 as metrics for MATH-500, and mean@64 and maj@64 for AIME24.
326 The COPO results report the best performance.* denotes the results are reproduced by ourselves.
327

328 Method	329 MATH 500		330 AIME 24		331 Mean Avg	332 Maj Avg
	333 mean@8	334 maj@8	335 mean@64	336 maj@64		
337 Qwen2.5-Instruct 3B*	48.35	56.11	2.45	8.36	45.75	53.41
338 GRPO	55.83	62.43	7.08	15.59	53.07	59.78
339 DAPO	55.93	61.81	5.47	13.74	53.07	59.09
340 COPO (ours)	60.38	65.06	6.67	14.48	57.34	62.2
341 Δ (vs best)	+4.55	+2.63	-0.41	-1.11	+4.27	+2.42
342 Qwen2.5-Instruct 7B*	58	61.73	9.38	14.7	55.25	59.07
343 GRPO	63.58	66.65	12.86	20.35	60.71	64.03
344 DAPO	62.15	65.76	11.77	17.94	59.3	63.05
345 COPO (ours)	65.8	69.27	13.85	21.07	62.86	66.54
346 Δ (vs best)	+2.22	+2.62	+0.99	+0.72	+2.15	+2.51



(a) Performance on MATH-500

(b) Performance on AIME24

349 Figure 3: Performance of GRPO and COPO on MATH-500 (mean@8) and AIME24 (mean@64)
350 using Qwen2.5 7B Instruct during training.
352

353 For the baseline experiments, we adopt the original GRPO method without any of the enhancements
354 introduced in DAPO, based on the experimental results presented in subsection 5.2. When applying
355 the COPO method, we set the value of w_{local} to zero for fully incorrect data, in order to prevent
356 $w_{global} < 1$ from reducing the overall loss. More details have been depicted in the appendix.
357

358 5 EXPERIMENT RESULTS AND DISCUSSION

360 5.1 MAIN RESULTS

362 Table 1 presents the performance comparison of our proposed COPO method against GRPO and
363 DAPO. Our method achieves superior inference accuracy over the GRPO approach with only a lim-
364 ited number of training steps. For Qwen2.5-Instruct 7B, COPO achieves a maximum mean@8 score
365 of 65.8% on the MATH-500 dataset, representing a 2.22% improvement over GRPO. Moreover,
366 COPO attains a mean@64 score of 13.85% on the AIME24 dataset, surpassing GRPO by 0.99%.
367 In terms of the majority voting (maj) metric, COPO also demonstrates consistent improvements,
368 achieving 69.27% (maj@8) on MATH-500 and 21.07% (maj@64) on AIME24, both outperforming
369 the results of GRPO and DAPO.

370 For Qwen2.5-Instruct 3B, COPO also demonstrates impressive performance. On the MATH-500
371 dataset, COPO achieves a peak mean@8 accuracy of 60.38%, marking a 4.55% improvement over
372 GRPO. When evaluated using the majority voting metric, COPO continues to show consistent gains,
373 achieving 2.63% (maj@8) improvement over GRPO on MATH-500. However, COPO underper-
374 forms the baseline on AIME24 by 0.41%, indicating that our method cannot achieve its full potential
375 when there is a large mismatch between model capacity and task difficulty. Additional experiments
376 and analyses are provided in the appendix.

377 Figure 3 presents a comparison of the test performance of the Qwen2.5-Instruct 7B under the GRPO
378 and COPO algorithms. Subfigures (a) and (b) show the evolution of mean@8 performance on

378 Table 2: Performance of different loss aggregation modes and KL divergence values on GSM8K
 379 (\dagger mean@8, \ddagger maj@8) and AIME25 (\dagger mean@64, \ddagger maj@64).

Method	token-level loss	KL	GSM8K \dagger	GSM8K \ddagger	AIME25 \dagger	AIME25 \ddagger
COPO*	✗	✓	86.10	89.56	3.82	10.00
	✓	✓	85.67	89.06	2.40	5.08
	✗	✗	85.62	88.83	3.02	8.29
	✓	✗	85.63	89.00	2.60	7.21

385 Table 3: Ablation study of COPO on Qwen2.5-Instruct 3B (MATH-500, \dagger mean@8, \ddagger maj@8). “Loss
 386 type” specifies the components of the optimization objective. “Hybrid strategy” denotes the method
 387 used to combine the local and global loss terms. “Zero control” indicates whether the local loss
 388 weight w_{local} is set to 0 for samples with completely incorrect outputs.

Method	Loss Type	Hybrid Strategy	Zero Control	MATH-500 \dagger	MATH-500 \ddagger
baseline	local	-	✗	55.83	62.43
+GO-Selective	local & global	binary	✓	58.88	64.51
+GO-Blended	local & global	soft blending	✗	59.80	64.32
+GO-Only	global	-	✗	60.35	64.60

395 MATH-500 and AIME24 during training. As shown, GRPO achieves a rapid accuracy increase
 396 in the early stages but suffers from a performance drop in later steps. In contrast, COPO maintains
 397 relatively stable performance and achieves the best results in later training stages. This suggests
 398 that COPO, by introducing inter-group rewards and a dynamic weighting strategy, is able to ex-
 399 tract meaningful learning signals from data with high intra-group consistency, thereby mitigating
 400 the impact of vanishing gradients caused by the zero advantage of some groups.

401 Notably, DAPO performs poorly compared to GRPO when trained with the same amount of data,
 402 achieving a maximum accuracy of only 5.47% on the AIME24 data set. On the 7B model, DAPO
 403 performs even worse, with its weighted mean@8 score decreasing by 1.39% relative to GRPO. These
 404 results suggest that DAPO’s advantages may not be effectively demonstrated on smaller models
 405 when reasoning and training are conducted with equivalent data volumes.

406 5.2 ANALYSIS OF COPO

408 **Ablation Study on Implementation Modifications** We initially investigate two common modi-
 409 fications to the GRPO framework: token-level loss and the KL term, aiming to establish a stronger
 410 experimental baseline. We extend GRPO with global optimization by adding a global loss signal
 411 for groups with zero advantage, and then evaluate different combinations of token-level loss and KL
 412 regularization, similar to DAPO. Table 2 reports results for Qwen2.5-Instruct 3B on GSM8K and
 413 AIME25. The model performs best without token-level loss but with KL regularization, improving
 414 mean scores by 0.47% on GSM8K and 1.42% on AIME25 compared to the opposite setting. Based
 415 on these findings, we retain the original GRPO configuration for subsequent COPO optimization
 416 without additional modifications.

417 To demonstrate the effectiveness of different modules of COPO, we investigate three key questions.
 418 **First**, we examine whether data with zero in-group advantage truly lacks learning value. **Second**,
 419 we explore whether utilizing the global optimization can improve performance. **Third**, we aim to
 420 determine how to balance global and local rewards to maximize the model’s capacity.

421 For the first question, we introduce the variant of GO-Selective (Global Optimization Selective),
 422 where the global optimization is applied to a prompt only when all of the extracted answers of this
 423 prompt are incorrect, and in all other cases, the local reward from GRPO is used without modi-
 424 fication. For the second question, we introduce the variant of GO-Only (Global Optimization Only),
 425 in which the model relies exclusively on the global optimization, with w_{local} in Equation 10 set to
 426 zero. Regarding the third question, we propose the variant of GO-Blended (Global Optimization
 427 Blended), which applies soft blending without any specific handling of all-zero cases. Addition-
 428 ally, we investigate the impact of the weight and threshold of soft blending on model performance.
 429 Table 3 presents the experimental results of these variants of COPO on the MATH-500 dataset.

431 **Effectiveness of “Ineffective” Data** Under the GO-Selective setting, the global optimization is
 432 utilized exclusively in cases where all sampled answers generated by the model are incorrect. The

432 GO-Selective experiment exclusively optimizes the fully incorrect paths that fail to receive effective
 433 advantage signals within GRPO, thereby providing targeted evidence of our method’s ability to
 434 extract effective signals from “ineffective data” deprecated by DAPO. On the MATH-500 dataset,
 435 GO-Selective achieves improvements of 3.05% and 2.08% over the baseline in terms of the mean
 436 and maj metrics, respectively. This demonstrates that training data with all-zero outcomes still holds
 437 learning value, and the incorporation of global optimization enables the model to effectively leverage
 438 useful information from those fully incorrect training examples.

439 **Impact of Global Signals** To evaluate whether the introduction of a global optimization mecha-
 440 nism leads to tangible performance improvements, we introduce the GO-Only experiment, in which
 441 the model is updated solely based on the advantage derived from the global reward. As shown in
 442 Table 3, the GO-Only setting achieves strong performance on both the mean@8 and maj@8 met-
 443 rrics, significantly outperforming the baseline with 4.52% improvements in mean@8 and 2.17% in
 444 maj@8, consistently outperforming the baseline. This result indicates that our global optimization
 445 formulation allows the model to capture both positive signals from correct trajectories and penalties
 446 from incorrect ones, thereby improving overall performance.

447 **Influence of the Hybrid Strategy** Under the GO-
 448 Blended setting, the model achieves performance im-
 449 provements of 3.97% and 1.89% on the mean@8 and
 450 maj@8 metrics compared to the baseline, demon-
 451 strating that our soft-blending approach effectively integrates the
 452 two optimization strategies. The method of combining
 453 the global optimization with the original local optimiza-
 454 tion in GRPO also leads to different impacts on the final
 455 results. As shown in Equation 10, higher consistency en-
 456 tropy of the answer list corresponding to greater weight
 457 assigned to the local loss, and lower entropy results in
 458 greater weight for the global loss. The weight allocation
 459 is controlled by the parameters γ and ρ in the equation.

460 The manner of integrating global loss with GRPO’s original local loss also significantly influences
 461 performance. With an increasing slope, the weight distribution becomes more binary, indicating
 462 a preference for using either global or local optimization exclusively. In contrast, when the slope
 463 is smaller, the weight distribution tends to be more linear, suggesting that the loss computation
 464 incorporates both types of loss.

465 Table 4 shows that as the threshold γ increases from 3 to 10, model accuracy on the benchmark
 466 gradually improves. This effect may arise from partial signal cancellation between the loss types,
 467 where the global term reduces inter-sample differences and weakens contrastive effectiveness.

468 In our soft-blending strategy, the proportion of global loss is controlled via the threshold parameter
 469 ρ in Equation 10. With a smaller threshold, more trajectories upper to the threshold are assigned a
 470 high w_{local} , meaning a larger portion of the training data relies mainly on local rewards. Conversely,
 471 a higher threshold results in w_{local} approaching zero, indicating a greater reliance on global loss.

472 We examined the impact of varying threshold ρ values on model performance. The accuracy curves
 473 for GRPO and small thresholds ($\rho = 0.5$) show a declining trend in later training stages. As the
 474 threshold increases, the accuracy on the MATH-500 dataset improves progressively, suggesting that
 475 greater use of global optimization enhances the model’s performance on mathematical reasoning
 476 tasks. From the above results, it can be observed that setting a larger slope γ and a higher threshold
 477 ρ in COPO training leads to better reasoning performance (e.g., $\gamma = 20$, $\rho = 1.5$). Lower parameter
 478 values, in contrast, result in diminished performance gains.

479 6 CONCLUSIONS

480 In this paper, we propose a novel consistency-aware policy optimization framework that incorporates
 481 a structured global reward mechanism based on outcome consistency, while employing an entropy-
 482 based soft-blending strategy to effectively integrate local and global optimization objectives. By
 483 effectively leveraging the information embedded in challenging training data, COPO achieves an
 484 important improvement over GRPO, suggesting that fully utilizing intra-group data with zero ad-
 485 vantage values contributes positively to the training process. More details will be discussed in the
 486 appendix.

Table 4: Ablation study of soft-blending weights γ and ρ on Qwen2.5-Instruct 3B.

γ	ρ	MATH-500 [†]	MATH-500 [‡]
3	1	55.18	60.81
5	1	59.05	63.75
10	1	59.40	63.97
20	0.5	56.23	61.97
20	1.2	59.30	64.12
20	1.5	60.38	65.06

486 7 REPRODUCIBILITY STATEMENT
487488 We have made every effort to ensure the reproducibility of our results. The code, developed on top of
489 the VERL framework, has been anonymized and included in the supplementary materials. Detailed
490 descriptions of the experimental setup, including model configurations and hardware specifications,
491 are provided in section 4 and section A.2.492 493 REFERENCES
494495 Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Ale-
496 man, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical
497 report. *arXiv preprint arXiv:2303.08774*, 2023.498 Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge,
499 Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.500 Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep
501 reinforcement learning from human preferences. *Advances in neural information processing sys-
502 tems*, 30, 2017.503 Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser,
504 Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to
505 solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.506 Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu,
507 Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms
508 via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.509 Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec
510 Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv
511 preprint arXiv:2412.16720*, 2024.512 Jiaming Ji, Mickel Liu, Josef Dai, Xuehai Pan, Chi Zhang, Ce Bian, Boyuan Chen, Ruiyang Sun,
513 Yizhou Wang, and Yaodong Yang. Beavertails: Towards improved safety alignment of llm via
514 a human-preference dataset. *Advances in Neural Information Processing Systems*, 36:24678–
515 24704, 2023a.516 Jiaming Ji, Tianyi Qiu, Boyuan Chen, Borong Zhang, Hantao Lou, Kaile Wang, Yawen Duan,
517 Zhonghao He, Jiayi Zhou, Zhaowei Zhang, et al. Ai alignment: A comprehensive survey. *arXiv
518 preprint arXiv:2310.19852*, 2023b.519 Maxwell Jia. Aime 2024 dataset. https://huggingface.co/datasets/Maxwell-Jia/AIME_2024, 2024. Accessed: 2025-05-06.520 Yue Jiang, Jiawei Chen, Dingkang Yang, Mingcheng Li, Shunli Wang, Tong Wu, Ke Li, and Lihua
521 Zhang. Comt: Chain-of-medical-thought reduces hallucination in medical report generation. In
522 *ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing
523 (ICASSP)*, pp. 1–5, 2025. doi: 10.1109/ICASSP49660.2025.10887699.524 Zhong-Zhi Li, Duzhen Zhang, Ming-Liang Zhang, Jiaxin Zhang, Zengyan Liu, Yuxuan Yao, Haotian
525 Xu, Junhao Zheng, Pei-Jie Wang, Xiuyi Chen, et al. From system 1 to system 2: A survey of
526 reasoning large language models. *arXiv preprint arXiv:2502.17419*, 2025.527 Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan
528 Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let’s verify step by step. *arXiv preprint
529 arXiv:2305.20050*, 2023.530 Yenting Lin. Aime 2025 dataset. https://huggingface.co/datasets/yentinglin/aime_2025, 2025. Accessed: 2025-05-06.531 Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao,
532 Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint
533 arXiv:2412.19437*, 2024.

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

543 Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language under-
 544 standing by generative pre-training, 2018.

545 Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea
 546 Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances*
 547 *in Neural Information Processing Systems*, 36:53728–53741, 2023.

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

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

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

556 Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng,
 557 Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. *arXiv preprint*
 558 *arXiv: 2409.19256*, 2024.

559 Feifan Song, Bowen Yu, Minghao Li, Haiyang Yu, Fei Huang, Yongbin Li, and Houfeng Wang.
 560 Preference ranking optimization for human alignment. In *Proceedings of the AAAI Conference*
 561 *on Artificial Intelligence*, volume 38, pp. 18990–18998, 2024.

562 Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée
 563 Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and
 564 efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

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

568 An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li,
 569 Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2. 5 technical report. *arXiv preprint*
 570 *arXiv:2412.15115*, 2024.

571 Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Tiantian Fan, Gaohong
 572 Liu, Lingjun Liu, Xin Liu, et al. Dapo: An open-source llm reinforcement learning system at
 573 scale. *arXiv preprint arXiv:2503.14476*, 2025.

574
 575
 576
 577
 578
 579
 580
 581
 582
 583
 584
 585
 586
 587
 588
 589
 590
 591
 592
 593

594 **A APPENDIX**595 **A.1 ALGORITHM EXPLANATION**596 **A.1.1 PROXIMAL POLICY OPTIMIZATION, PPO**

597 The objective function for conventional PPO is defined as:

601
$$J_{\text{PPO}}(\theta) = \mathbb{E}_{q, o \sim \pi_{\theta_{\text{old}}}} \left[\sum_{t=1}^{|o|} \min \left(\frac{\pi_{\theta}(o_t \mid q, o_{<t})}{\pi_{\theta_{\text{old}}}(o_t \mid q, o_{<t})} A_t, \text{clip}(r_t(\cdot)) A_t \right) \right] \quad (12)$$
 602
603

604 where θ represents the parameters of the current policy π_{θ} ; o_t is the token generated at step t , $o_{<t}$ 605 represents the preceding token sequence. A_t is the advantage function, which captures the relative 606 value of taking action o_t at state s_t and is computed by $A_t = r_t + \gamma V(s_{t+1}) - V(s_t)$, where $V(s_t)$ 607 is the value of state s_t that is usually estimated by a value network. $\gamma \in [0, 1]$ is the discount factor, 608 controlling the trade-off between immediate and future rewards. PPO uses the clipping operator:

609
$$\text{clip}(\cdot) = \text{clip}(r_t(\theta), 1 - \epsilon, 1 + \epsilon)$$
 610

611 to restrict the update ratio $r_t(\theta)$ within the interval $[1 - \epsilon, 1 + \epsilon]$, and ϵ is the clip range of the 612 importance sampling ratio. By doing so, PPO prevents excessively large policy updates that could 613 destabilize training, ensuring that the new policy does not deviate too far from the previous one 614 while still allowing sufficient flexibility for improvement.615 **A.1.2 GROUP RELATIVE POLICY OPTIMIZATION, GRPO**

616 GRPO simplifies the training process compared to PPO by utilizing reward-based advantage esti- 617 mation. Instead of relying on a separate value network, GRPO calculates advantages by directly 618 comparing rewards among samples generated from the same input, streamlining the overall archi- 619 tecture.

620 Given an input prompt q , the previous policy $\pi_{\theta_{\text{old}}}$ produces a set of G candidate output sequences, 621 denoted as $\mathcal{O}q = \{o_1, o_2, \dots, o_G\}$. Each sequence is subsequently evaluated using a task-specific 622 reward function $r\phi$, constructed in accordance with the optimization objective, resulting in a cor- 623 responding reward set $\{r_1, r_2, \dots, r_G\}$. The direction of the policy update is determined by the 624 relative ranking of rewards within the group. Samples that receive rewards above the group aver- 625 age are encouraged by increasing their likelihood under the policy, while those with below-average 626 rewards are discouraged by reducing their corresponding policy probabilities.

627 Based on this, the GRPO advantage is computed as:

628
$$\hat{A}_i = \frac{r_i - \mu_r}{\sigma_r},$$
 629
630

631 where $\mu_r = \text{mean}(\{r_i\}_{i=1}^G)$, $\sigma_r = \text{std}(\{r_i\}_{i=1}^G)$. By substituting the new advantage \hat{A}_i , the 632 group B, and the response set $\{o_1, o_2, \dots, o_G\}$ sampled by the policy model into the PPO objective, 633 we derive the objective function of GRPO:

634
$$J_{\text{GRPO}}(\theta) = \mathbb{E}_{q, \{o_i\} \sim \pi_{\theta_{\text{old}}}} \left[\frac{1}{G} \sum_{i=1}^G \frac{1}{|o_i|} \sum_{t=1}^{|o_i|} \min \left(\frac{\pi_{\theta}(o_{i,t} \mid q, o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t} \mid q, o_{i,<t})} \hat{A}_i, \text{clip}(\cdot) \hat{A}_i \right) - \beta \mathbb{D}\text{KL}[\pi_{\theta} \parallel \pi_{\text{ref}}] \right].$$
 635
636
637

638 **A.1.3 RULE-BASED REWARD**

639 Rule-based reward assigns scores to model outputs based on predefined rules. In our setting, cor- 640 rectness is the only evaluation criterion, which helps reduce the risk of reward hacking. Specifically, 641 the model is prompted to generate responses in a required format, and the final answer is extracted 642 and directly compared with the ground truth to assign the reward:

643
$$R(o) = \begin{cases} 1, & \text{is_equivalent}(\tau, \hat{\tau}) \\ 0, & \text{otherwise} \end{cases} \quad (13)$$
 644
645
646

647 where τ is the predicted answer extracted from response o and $\hat{\tau}$ is the ground truth.

648 A.1.4 DEMONSTRATIVE EVALUATION OF COPO FRAMEWORK
649

650 **Demonstration of the COPO computation procedure** To illustrate the operational mechanism
651 of COPO, we present a concrete example. Consider a batch consisting of 5 data instances, where the
652 model generates 6 candidate responses for each instance. For demonstration purposes, we take one
653 example from the batch: ***the question “1 + 1 = ?”***. The model produces 6 reasoning-based responses
654 to this question, such as: ***“The answer is 2. Answer: \$2.”*** From each response, the final predicted
655 answer is extracted. Suppose the extracted answers are: **[2, 2, 2, 3, 3, 4]**. Based on ground truth
656 comparison, the corresponding accuracy rewards are assigned as: **[1, 1, 1, 0, 0, 0]**. Subsequently,
657 COPO computes the local rewards and local advantages for each response according to GRPO using
658 5:
659

$$\hat{A}_o^{\text{local}} = \frac{R(o) - \text{mean}(\{R(o_i)\}_{i=1}^G)}{\text{std}(\{\hat{R}(o_i)\}_{i=1}^G)}.$$

660 For this reward list, the mean is 0.5 and the standard deviation is 0.5, resulting in a local advantage
661 list of **[1, 1, 1, -1, -1, -1]** for the corresponding sample. The final local loss is computed from this
662 advantage using Equation 4:
663

$$J_{\text{local}}(\theta) = \mathbb{E}_{q, \{o_i\} \sim \pi_{\theta_{\text{old}}}} \left[\frac{1}{\sum_{i=1}^G |o_i|} \sum_{i=1}^G \sum_{t=1}^{|o_i|} \min \left(\frac{\pi_{\theta}(o_{i,t} | q, o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t} | q, o_{i,<t})} \hat{A}_{o_i}^{\text{local}}, \text{clip}(\cdot) \hat{A}_{o_i}^{\text{local}} \right) \right],$$

664 Next, based on the Equation:
665

$$\hat{R}(q) = \frac{1}{G} \sum_{i=1}^G r_i, \quad (14)$$

666 the global reward for this data prompt is calculated to be **0.5**. For each of the 5 samples in the
667 batch, the global reward can be computed by following the procedure described above, resulting in
668 5 values. We set the global rewards for these samples as:
669

$$[\frac{1}{6}, \frac{1}{6}, \frac{2}{3}, \frac{1}{2}, \frac{1}{2}].$$

670 The global advantage is then calculated based on Equation 6:
671

$$\hat{A}_q^{\text{global}} = \frac{\hat{R}(q_j) - \text{mean}(\{\hat{R}(q_j)\}_{j=1}^B)}{\text{std}(\{\hat{R}(q_j)\}_{j=1}^B)}, \text{ for } \forall o_i \in \mathcal{O}_q.$$

672 For this global reward list, the mean is 0.4 and the standard deviation is 0.2, resulting in a local
673 advantage list of
674

$$[-1.167, -1.167, 1.333, 0.500, 0.500, 0.500]$$

675 for the corresponding sample. The final global loss is computed from this advantage using Equation
676 7.
677

$$J_{\text{global}}(\theta) = \mathbb{E}_{q, \{o_i\} \sim \pi_{\theta_{\text{old}}}} \left[\frac{1}{\sum_{i=1}^G |o_i|} \sum_{i=1}^G \sum_{t=1}^{|o_i|} \min \left(\frac{\pi_{\theta}(o_{i,t} | q, o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t} | q, o_{i,<t})} \hat{A}_q^{\text{global}}, \text{clip}(\cdot) \hat{A}_q^{\text{global}} \right) \right].$$

678 Subsequently, given the extracted answer list **[2, 2, 2, 3, 3, 4]** for this data prompt, the consistency
679 entropy is calculated using Equation 8,
680

$$H(q) = - \sum_{\tau \in T_q} p(\tau) \cdot \log p(\tau), \quad p(\tau) = \frac{\text{count}(\tau)}{G},$$

681 where $\text{count}(\tau)$ denotes the number of occurrences of τ . For this example, we have:
682

$$p('2') = 0.5, \quad p('3') = \frac{1}{3}, \quad p('4') = \frac{1}{6}.$$

683 The resulting consistency entropy H is 1.459.
684

685 By substituting the consistency entropy into the weight computation formula (Equation 10), where
686 we set $\gamma = 3$ and $\rho = 1$, the sigmoid function returns $w_{\text{local}} = 0.799$, and thus $w_{\text{global}} = 0.201$.
687

702
 703
 704
 705
 706
 707
 708
 709
 710
 711
 712
 713
 714
 715
 716
 717
 718
 719
 720
 721
 722
 723
 724
 725
 726
 727
 728
 729
 730
 731
 732
 733
 734
 735
 736
 737
 738
 739
 740
 741
 742
 743
 744
 745
 746
 747
 748
 749
 750
 751
 752
 753
 754
 755

System Prompt

You are Qwen, created by Alibaba Cloud. You are a helpful assistant.

User Prompt

Solve the following math problem step by step. The last line of your response should be of the form Answer: \$Answer (without quotes) where \$Answer is the answer to the problem.

Question

Remember to put your answer on its own line after "Answer:".

Figure 4: Training and Test Prompts

$$\mathcal{L}_q = w_{\text{local}}(H(q)) \cdot \mathcal{L}_{\text{local}}(q) + w_{\text{global}}(H(q)) \cdot \mathcal{L}_{\text{global}}(q),$$

and the weighting functions are defined as:

$$w_{\text{local}}(H) = \sigma(\gamma(H - \rho)), w_{\text{global}}(H) = 1 - w_{\text{local}}(H).$$

Finally, according to Equation 3, the local loss and global loss are combined using w_{local} and w_{global} to obtain the final loss value.

$$J_{\text{COPO}}(\theta) = \mathbb{E}_{q \sim \mathcal{D}} \left[w(H_q) \cdot \mathcal{L}_{\text{local}}(q) + (1 - w(H_q)) \cdot \mathcal{L}_{\text{global}}(q) \right].$$

Utilization of Data with Zero Local Advantage in COPO COPO provides optimization signals for data with zero advantages in GRPO, thereby preventing gradient vanishing and sample wastage. Specifically, consider a batch with five data instances, whose corresponding accuracy reward lists are **[0, 0, 0, 0, 0, 0]**, **[0, 0, 0, 0, 0, 0]**, **[1, 1, 1, 1, 1, 1]**, **[0, 0, 0, 1, 1, 1]**, and **[0, 0, 0, 1, 1, 1]**. For the first three instances, since all responses receive uniform local rewards, their local advantages are zero according to Eq.5.

Without incorporating the global advantage, their final advantages remain zero, resulting in zero gradients and thus gradient vanishing. COPO addresses this issue by assigning global advantages to the data. According to Eq.14, the global rewards of the five samples are computed as the mean accuracy of the model’s responses, yielding **[0, 0, 1, 0.5, 0.5]**. Based on Eq.7, the corresponding global advantages are **[-1.07, -1.07, 1.60, 0.27, 0.27]**, where the first three instances have nonzero global advantages that reflect the model’s accuracy on these data. Finally, by Eq.3, these nonzero global advantages contribute to the total advantage of the first three data points, thereby avoiding gradient vanishing.

A.1.5 REWARD HACKING IN MULTI-OBJECTIVE OPTIMIZATION

When applying multiple rewards for multi-objective optimization, advantage degeneration serves as a direct cause of reward hacking. When different reward signals have varying degrees of difficulty to achieve, the model tends to concentrate its strategy on optimizing the easier objective.

For example, when designing both a format reward and an outcome correctness reward, the initial policy finds it much easier to satisfy formatting requirements than to achieve correct reasoning. Consequently, the model rapidly shifts to producing outputs that conform to format specifications while ignoring reasoning quality. This leads to reward homogenization within the group, further degenerating the advantage estimation and causing the training process to collapse without further effective learning.

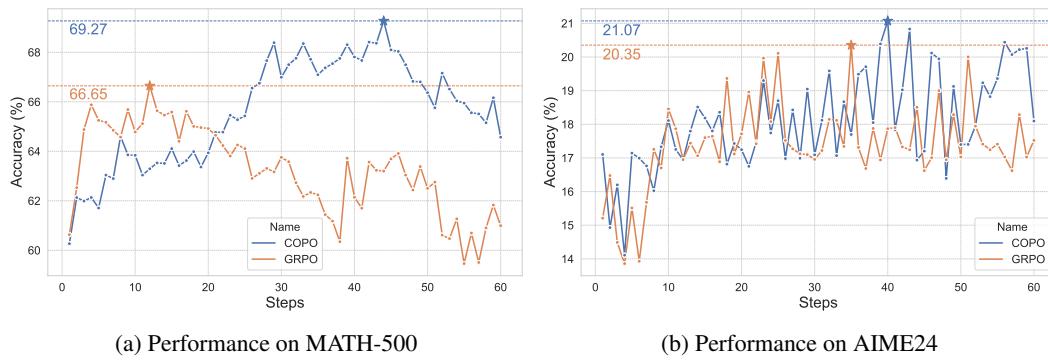


Figure 5: Performance of GRPO and COPO on MATH-500 and AIME24 (maj@8) using Qwen2.5-7B-Instruct during training

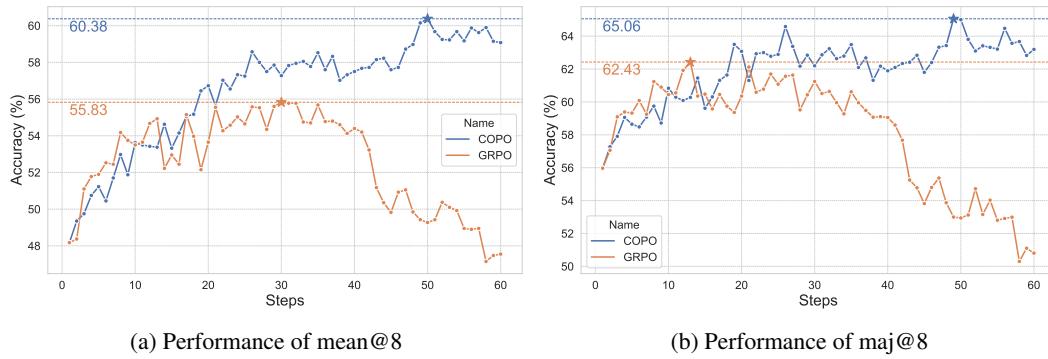


Figure 6: Performance of GRPO and COPO on MATH-500 with mean@8 and maj@8 using Qwen2.5-3B-Instruct during training

A.2 EXPERIMENTS SETTING DETAILS

All experiments for the 3B model were conducted on four GPUs with 80 GB of memory each, while those for the 7B model were carried out on four GPUs with 96 GB of memory each. During evaluation, the dataset used the same prompts as the training set (DAPO-MATH-17k) to ensure consistency. Figure 4 presents the detailed structure of the prompt. When obtaining rule-based rewards, we extract the final answer from the reasoning path in the required format and use the Python package *math_verify* to determine whether the answer matches the ground truth.

A.3 MORE EXPERIMENTS RESULTS

A.3.1 FIGURES OF MAIN EXPERIMENTS

The test performance of Qwen2.5-7B-Instruct under the GRPO and COPO algorithms is compared in Figure 5. The progression of maj performance over the course of training is shown in subfigures (a) and (b) for the MATH-500 and AIME24 datasets, respectively. COPO demonstrates more consistent gains in maj accuracy over GRPO on both datasets, suggesting that it enables the model to acquire more general and transferable problem-solving strategies.

The test performance of Qwen2.5-3B-Instruct under the GRPO and COPO algorithms on MATH-500 is compared in Figure 6. The COPO method demonstrates a consistent upward trend in both the mean@8 and maj@8 metrics.

Figure 7 shows the entropy dynamics of the 7B and 3B models during training with COPO and GRPO. For the 7B model, the entropy trends of COPO and GRPO are similar, but COPO maintains a more stable entropy level in the later steps. For the 3B model, COPO yields consistently higher entropy, indicating its ability to preserve response diversity throughout training.

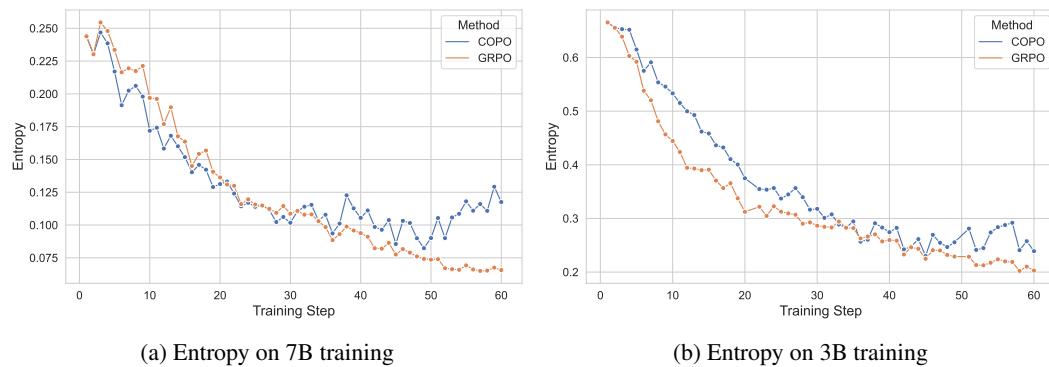


Figure 7: Entropy of COPO training on Qwen2.5-7B-instruct and Qwen2.5-3B-Instruct

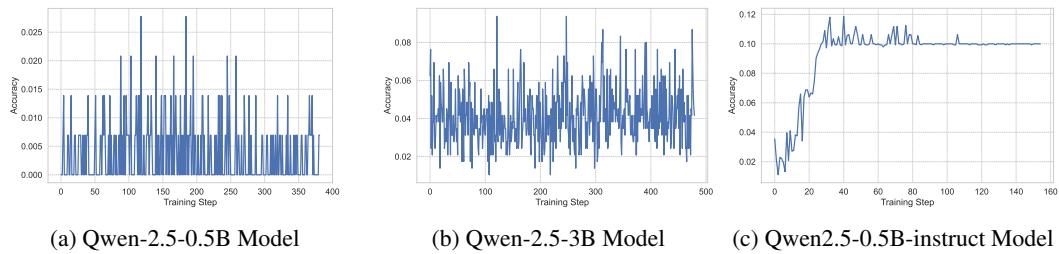


Figure 8: Training Accuracy Dynamics of GRPO on the Small and Base Models

A.3.2 EXPERIMENTS ON SMALL MODEL AND BASE MODEL

We conducted GRPO experiments on base models (Qwen2.5-0.5B, Qwen2.5-3B) as well. Due to their lack of instruction-following ability, these models struggled to produce correctly formatted outputs. To address this, we introduced a format reward:

$$R(\tau, \hat{\tau}) = \begin{cases} 0, & \text{is_null}(\hat{\tau}) \\ 1, & \text{is_equivalent}(\tau, \hat{\tau}) \\ 0.1, & \text{otherwise} \end{cases} \quad (15)$$

where τ is the formatted answer extracted from prediction and $\hat{\tau}$ is the ground-truth. The format reward is defined as 0 for incorrect formats, 0.1 for correct format but incorrect answers, and 1 for correct answers.

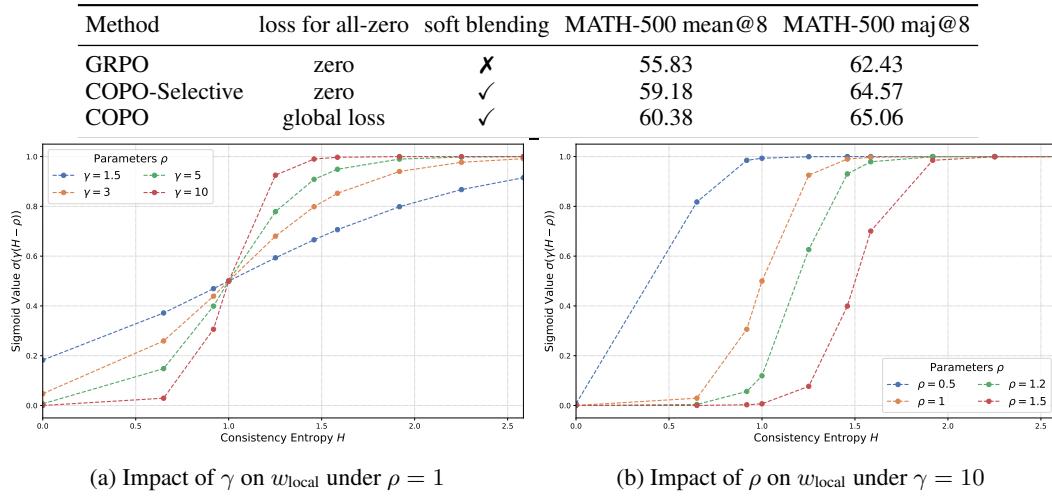
However, even with the format reward, the models failed to maintain proper output formatting. As shown in Figure 8a and 8b, the reward score remained below 0.1 after 300 training steps with no upward trend.

We also ran experiments on Qwen2.5-0.5B-Instruct model, applying the same format reward to regulate output. According to Figure 8c, under this scheme, the model achieved stable formatting within 40 steps. However, due to limited base capabilities, it was unable to sample correct answers on this dataset, with most prompts yielding zero advantage and no further learning progress.

These results suggest that RL methods cannot directly drive small models that lack instruction-following ability toward desired behaviors. Dataset difficulty calibration and cold-start strategies may be necessary prerequisites for RL training on small base models.

A.3.3 IMPACT OF LOSS MASKING ON FULLY INCORRECT SAMPLES

To further investigate whether fully incorrect samples contribute to model learning, we conducted an additional experiment called COPO-Selective, in which the loss corresponding to fully incorrect samples is set to zero, while the remaining samples still use soft blending to combine local and global losses. Compared to our main method, the only difference in COPO-Selective is how fully incorrect samples are handled. The main method applies global loss to these samples, while COPO-

864 Table 5: Comparison of COPO-Selective and other methods across MATH-500 datasets.
865882 Figure 9: Variation of w_{local} with Consistency Entropy H under Different Hyperparameter Settings
883

884 Selective excludes them from optimization by assigning a zero loss, effectively removing them from
885 weight updates.
886

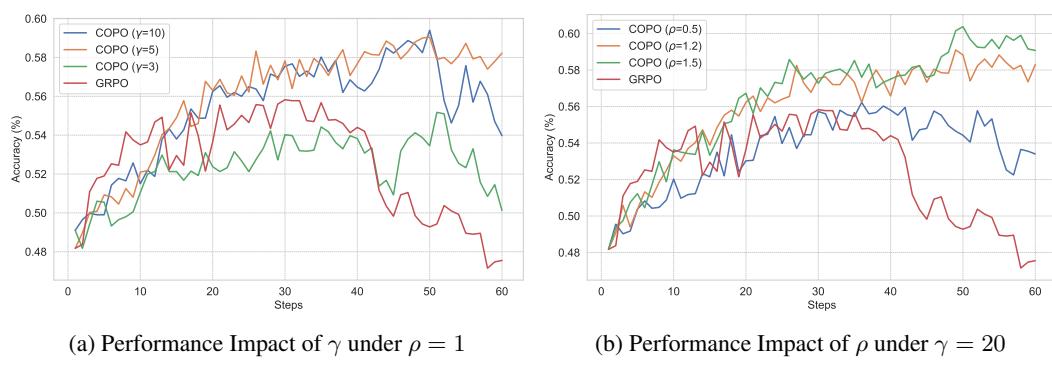
887 As shown in Table 5, COPO-Selective achieves a significant improvement over GRPO, but still
888 underperforms the main method by 1.2% and 0.49% in mean@8 and maj@8, respectively. This
889 suggests that incorporating loss signals for fully incorrect samples with zero intra-group advantage
890 helps the model extract useful information from them.

891 A.4 DISCUSSION

892 A.4.1 EFFECT OF HYPERPARAMETERS IN SOFT BLENDING

893 Figure 9 illustrates the effect of the soft-blending hyperparameters γ and ρ on the weight w_{local} .
894 From subfigure (a), we observe that with ρ fixed, γ controls the sharpness of the soft blending curve.
895 As γ increases, the curve transitions from linear to more binary-like, meaning that larger values of γ
896 push w_{local} closer to 0 or 1 for more data points. Subfigure (b) shows that with γ fixed, ρ determines
897 the horizontal shift of the blending curve. Smaller values of ρ shift the curve leftward toward the
898 y-axis, resulting in more data points receiving w_{local} values close to 1. In contrast, larger values of ρ
899 shift the curve rightward, assigning more data points with w_{local} values close to 0, which indicates a
900 greater reliance on global optimization.

901 Figure 10 shows how the performance of COPO on the MATH-500 test set varies under different
902 hyperparameter settings. It can be observed that higher values of γ and ρ result in the highest
903

916 Figure 10: Effect of different γ and ρ in Soft Blending with Qwen2.5-3B-Instruct on MATH-500
917

918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971

Question:

Twelve friends met for dinner at Oscar's Overstuffed Oyster House, and each ordered one meal. The portions were so large, there was enough food for 18 people. If they share, how many meals should they have ordered to have just enough food for the 12 of them?

GRPO:

To solve this problem, we need to determine how many meals the 12 friends should have ordered so that they have just enough food for themselves, given that the portions were enough for 18 people.

First, we need to find out how much food each person consumes. Since the portions are large and are enough for 18 people, we can say that the total portions of food for 18 people are enough for each of the 18 people to eat a full meal. Therefore, the food for one person is $\frac{1}{18}$ of the total food.

Now, we need to find out how much food is enough for 12 people. Since each person requires $\frac{1}{18}$ of the total food, for 12 people, the total amount of food needed would be:

$$12 \times \frac{1}{18} = \frac{12}{18} = \frac{2}{3}$$

This means that to have just enough food for 12 people, they should have ordered $\frac{2}{3}$ of the total portions. Since the portions they had were enough for 18 people, and they need just enough for 12 people, **they should have ordered the equivalent of what 6 people would have ordered (since $\frac{2}{3}$ of 18 people is 6 people).**

Therefore, they should have ordered meals for 6 people to have just enough food for 12 people.

So, the answer is:

Answer: \$6 → Wrong

COPO:

To solve this problem, we need to determine how many meals each person should have ordered so that the total amount of food is just enough for 12 people, given that the original portions are enough for 18 people.

First, let's determine the ratio of the number of people the original meals are enough for to the number of people they want to serve. The original meals are enough for 18 people, and they want to serve 12 people. The ratio of the desired number of people to the original number of people is:

$$\frac{12}{18} = \frac{2}{3}$$

This means that each person should receive $\frac{2}{3}$ of a meal to serve 12 people with the amount of food originally intended for 18 people.

Since each of the 12 friends should receive $\frac{2}{3}$ of a meal, the total number of meals needed is:

$$12 \times \frac{2}{3} = 8$$

Therefore, they should have ordered 8 meals to have just enough food for the 12 of them.

Answer: \$8 → Right

Figure 11: Case of GRPO and COPO reasoning results on MATH-500

accuracy. This suggests that a more binary-like blending curve, along with a greater reliance on global optimization, can more effectively improve model performance.

A.5 CASE STUDY

To better understand the difference between methods, we conduct a case study on selected examples from the MATH-500 dataset. Figure 11 illustrates a representative example where GRPO fails due to incorrect intermediate reasoning, while COPO provides a complete and correct derivation.

A.6 LIMITATION

As shown in Table 1, when using the relatively small 3B model, our method exhibits weaker performance on AIME24, with a difference of 0.41% compared to the baseline. However, it achieves greater improvements on the simpler MATH-500 dataset. We also conducted experiments with the

972 Table 6: Comparison of GRPO and our method across MATH-500 and AIME24 datasets.* denotes
 973 the results are reproduced by ourselves
 974

975 976 Method	977 MATH 500		978 AIME 24		979 Mean Avg	980 Maj Avg
	981 mean@8	982 maj@8	983 mean@64	984 maj@64		
985 Qwen2.5-instruct 1.5B*	66.88	71.00	8.80	18.14	63.59	68.01
986 GRPO	70.00	73.55	11.46	19.23	66.69	70.48
987 COPPO ($\gamma = 5, \rho = 1$)	68.93	72.85	10.78	19.46	65.64	69.83
988 COPPO ($\gamma = 10, \rho = 1$)	68.83	73.12	10.78	19.92	65.54	70.11

981
 982 COPPO method on the Qwen2.5-Math-1.5B-Instruct model. Table 6 presents a performance compar-
 983 ision between our method and the baseline on the MATH-500 and AIME24 datasets, using both the
 984 mean and maj metrics. As shown, our method still lags behind GRPO by approximately 1% on most
 985 metrics.

986 This observation suggests that the current COPPO method may not offer advantages when applied
 987 to smaller math-tuned models. On one hand, smaller models typically have weaker generalization
 988 capabilities, making it difficult to fully leverage the potential benefits of combining local and global
 989 losses. In some cases, the objectives of local and global optimization may even conflict, leading to
 990 degraded performance. On the other hand, the Qwen2.5-Math-1.5B-Instruct model is specifically
 991 fine-tuned for mathematical tasks. Introducing a composite loss function that is not fully aligned
 992 with its task-specific pretraining objectives may interfere with its learned structural representations
 993 or reasoning mechanisms, thereby weakening overall performance.

994 995 A.7 RELATED WORKS

996 A.7.1 LLM REASONING

1000 The ability of LLMs to directly generate answers through autoregressive decoding is often referred
 1001 to as their 'System 1' capability Li et al. (2025). In contrast, solving complex problems through
 1002 deliberate, logical reasoning—by first thinking and then generating—is considered the 'System 2'
 1003 mode. CoT prompting has emerged as one of the most effective approaches to endow LLMs with
 1004 human-like reasoning ability. Early CoT Wei et al. (2022); Jiang et al. (2025) methods relied on in-
 1005 context learning by inserting exemplar reasoning processes into prompts, but such methods struggle
 1006 to generalize across a wider range of task domains. An alternative and more scalable approach is
 1007 to let models autonomously generate reasoning paths depending on the specific question. By fine-
 1008 tuning LLMs on high-quality reasoning trajectories, models can quickly learn human-like thought
 1009 patterns for particular problems. However, the annotation cost of such data is often prohibitive for
 1010 most researchers. As a result, a series of RL-based methods have emerged to improve the reasoning
 1011 abilities of LLMs without requiring fully supervised data.

1012 A.7.2 RL-BASED POSTED-TRAINING

1015 Early RL-based post-training methods focused primarily on aligning model outputs with human
 1016 preferences in multiple dimensions such as non-toxicity, fairness, or politeness, rather than explicitly
 1017 enhancing reasoning capability. The release of OpenAI's O1 Jaech et al. (2024) model shifted
 1018 attention toward improving reasoning via Monte Carlo Tree Search (MCTS) and process-level re-
 1019wards, encouraging models to explore higher-quality reasoning trajectories. However, this approach
 1020 still requires extensive computational resources to supervise the exploration process and provide re-
 1021ward or value signals. DeepseekMATH Shao et al. (2024b) introduced the GRPO training method
 1022 and demonstrated that sparse, outcome-level rewards could also guide models toward discovering
 1023 correct reasoning paths. R1 further proposed a rule-based reward system, removing the need for a
 1024 learned reward model and reducing computational overhead. Nevertheless, the inherent limitations
 1025 of GRPO led to the frequent disappearance of optimization signals within groups. DAPO Yu et al.
 1026 (2025) attempted to address instability and inefficiency during training, but it did not fundamentally
 1027 resolve the sample inefficiency caused by the design of GRPO.

1026
1027

A.8 LLM USAGE

1028 During manuscript preparation, Large Language Models (LLMs) were employed **solely** for linguis-
1029 tic refinement. Their use was restricted to improving grammar, readability, and stylistic clarity,
1030 without any involvement in research conception, methodology, experimental design, data analysis,
1031 or interpretation of results. All scientific ideas, analyses, and conclusions are entirely the work of
1032 the authors. The LLM-assisted edits were carefully reviewed to ensure accuracy, originality, and ad-
1033 herence to ethical standards. The authors retain full responsibility for the content of this manuscript.

1034

1035

1036

1037

1038

1039

1040

1041

1042

1043

1044

1045

1046

1047

1048

1049

1050

1051

1052

1053

1054

1055

1056

1057

1058

1059

1060

1061

1062

1063

1064

1065

1066

1067

1068

1069

1070

1071

1072

1073

1074

1075

1076

1077

1078

1079