BACK TO BASICS: REVISITING EXPLORATION IN REINFORCEMENT LEARNING FOR LLM REASONING VIA GENERATIVE PROBABILITIES

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement learning with verifiable rewards (RLVR) can improve reasoning accuracy while producing increasingly concentrated outputs. Under binary rewards, Group Relative Policy Optimization (GRPO) assigns identical positive advantages to correct responses within mixed-reward groups, without accounting for their generation confidence. We propose Probabilistic GRPO (ProGRPO), which introduces a confidence-based Advantage Re-weighting Mechanism (ARM). ARM uses the difference between prompt and response confidence, computed as geometric means of token probabilities over low-probability positions under the old policy. Within each reward class, lower-confidence responses receive larger adjusted advantage coefficients. The correction is disabled for uniform-reward groups and preserves advantage signs under a sufficient bound on its magnitude. Experiments on mathematical reasoning and code generation using Qwen2.5 and DeepSeek-R1-Distill-Qwen models show improvements in accuracy and multi-sample solution coverage. On Qwen2.5-7B, adding ARM to a matched GRPO with Clip-Higher baseline improves average Pass@1 and Pass@32 by 3.3 and 8.3 percentage points, respectively. The complete ProGRPO method improves these metrics over vanilla GRPO by 5.7 and 13.8 percentage points. Training-entropy and output-similarity measurements provide complementary evidence of reduced output concentration in the evaluated settings.

1 INTRODUCTION

Reinforcement learning (RL) has become an important component of post-training large language models (LLMs) for complex reasoning (Jaech et al., 2024; Guo et al., 2025). In particular, reinforcement learning with verifiable rewards (RLVR) uses automatically checkable outcomes to guide policy optimization, improving performance on tasks such as mathematical reasoning and code generation (Lambert et al., 2024; Guo et al., 2025). These advances motivate a closer examination of how reward-guided optimization shapes not only task accuracy, but also the range of successful responses that a policy can generate.

Despite improvements in accuracy, RLVR training can exhibit declining policy entropy and increasingly concentrated outputs (Cui et al., 2025; Yue et al., 2025). A policy may become more likely to produce a correct response on its first attempt while gaining less from additional samples, limiting the benefits of inference-time exploration. This tension motivates methods that improve reasoning performance while preserving solution coverage under multiple sampling attempts.

Existing approaches address this challenge through entropy regularization (Ziebart et al., 2008; Haarnoja et al., 2018; Cui et al., 2025; Wang et al., 2025b), increased upper clipping thresholds (Yu et al., 2025), dynamic clipping (Yang et al., 2025), and updates focused on high-entropy tokens (Wang et al., 2025a). These methods modify the optimization objective or control which token-level updates are retained. A complementary intervention is to reconsider how advantages weight sampled responses. Recent findings on negative reinforcement further suggest that the balance between reinforcing correct responses and suppressing incorrect ones can affect exploration and reasoning performance (Zhu et al., 2025).

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.

We focus on how Group Relative Policy Optimization (GRPO) allocates learning signals among equally rewarded responses (Shao et al., 2024). Under binary rewards, all correct responses within a group containing both correct and incorrect samples receive the same positive advantage, regardless of the policy's confidence in generating them. Yet a correct response containing low-confidence decisions may reveal successful behavior that the policy has not yet learned to generate reliably. This motivates a concrete question: *can generation confidence complement verifiable rewards in determining how strongly each sampled response should be reinforced?* The model's own probability signals provide a readily available basis for investigating this question (Li et al., 2025).

We propose **Probabilistic GRPO (ProGRPO)**, which introduces a confidence-based **Advantage Re-weighting Mechanism (ARM)**. ARM adds a correction to group-relative advantages using the difference between prompt and response confidence, estimated from token probabilities at low-probability positions. Within a mixed-reward group, the shared prompt term cancels when comparing two correct responses, assigning a larger adjusted advantage to the less confident one. Across prompts, the prompt term provides an input-dependent reference for the correction. ARM leaves verifiable rewards unchanged, is disabled for groups with uniform rewards, and preserves advantage signs when the correction is appropriately bounded.

We evaluate ProGRPO on mathematical reasoning, code generation, and out-of-distribution benchmarks using Qwen2.5 and DeepSeek-R1-Distill-Qwen models. On Qwen2.5-7B, adding ARM to GRPO with Clip-Higher improves average Pass@1 and Pass@32 by 3.3 and 8.3 percentage points, respectively, with all other training settings held fixed. Compared with vanilla GRPO, the complete ProGRPO method improves these metrics by 5.7 and 13.8 percentage points. These results indicate gains in both single-sample accuracy and multi-sample solution coverage, complemented by training-entropy and output-similarity measurements showing reduced output concentration.

Our main contributions are:

- **Confidence-based advantage correction.** We introduce ARM, which uses prompt and response confidence to differentiate advantage coefficients among equally rewarded responses without changing the verifiable reward signal.
- **Analysis of the correction mechanism.** We characterize sufficient conditions for advantage sign preservation and the relative weighting induced by ARM within a group. Ablations examine the effects of the confidence formulation, token selection, correction strength, and clipping.
- **Empirical evaluation.** Across multiple model scales and reasoning benchmarks, ProGRPO improves accuracy and multi-sample solution coverage, with complementary analyses of training dynamics and output diversity.

2 Preliminaries

2.1 Problem Setup

Let $q \sim D$ denote a prompt and $o = (o_1, \ldots, o_{|o|})$ a response generated by an autoregressive policy π_θ:

$$\pi_\theta(o \mid q) = \prod_{t=1}^{|o|} \pi_\theta(o_t \mid q, o_{<t}). \tag{1}$$

We consider outcome-supervised RLVR with a binary verifiable reward $R(q, o) \in \{-1, 1\}$, where $+1$ indicates a correct response and -1 an incorrect response. The learning objective is to maximize the expected reward:

$$\mathcal{J}_{\mathrm{RL}}(\theta) = \mathbb{E}_{q \sim D,\, o \sim \pi_\theta(\cdot|q)}\big[R(q, o)\big]. \tag{2}$$

2.2 Group Relative Policy Optimization

GRPO (Shao et al., 2024) builds on the clipped surrogate objective of PPO (Schulman et al., 2017), replacing the learned value baseline with reward statistics computed from a group of sampled responses. For each prompt q, we independently sample G responses $\{o_i\}_{i=1}^{G}$ from the old policy $\pi_{\theta_{\mathrm{old}}}(\cdot \mid q)$ and obtain rewards $R_i = R(q, o_i)$.

The group-relative advantage is

$$A_i = \frac{R_i - \bar{R}}{\sigma_R + \delta}, \qquad \bar{R} = \frac{1}{G}\sum_{j=1}^{G} R_j, \tag{3}$$

Here, σ_R is the population standard deviation of the group rewards, and $\delta > 0$ is a small numerical stability constant. Under outcome supervision, all tokens of response o_i share the same advantage A_i, which is treated as fixed during policy optimization.

Define the token-level probability ratio as

$$r_{i,t}(\theta) = \frac{\pi_\theta(o_{i,t} \mid q, o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t} \mid q, o_{i,<t})}. \tag{4}$$

For compactness, write the clipped surrogate term as

$$\ell_\epsilon(r, A) = \min\big(rA,\ \text{clip}(r, 1-\epsilon, 1+\epsilon)A\big), \tag{5}$$

where $\epsilon > 0$ controls clipping. This surrogate limits the incentive for certain large probability-ratio changes without imposing a hard constraint on the ratios.

Using per-response length normalization and omitting KL regularization, the GRPO policy objective is

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{\text{old}}\left[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{t=1}^{|o_i|}\ell_\epsilon\big(r_{i,t}(\theta), A_i\big)\right], \tag{6}$$

where $\mathbb{E}_{\text{old}}$ denotes expectation over $q \sim D$ and the group of responses sampled from $\pi_{\theta_{\text{old}}}(\cdot \mid q)$. KL regularization is disabled in our experiments. Equation (6) uses per-response length normalization (sample-mean aggregation). Our main GRPO baseline instead uses token-mean aggregation, as detailed in Section 4.1. Sample-mean variants are evaluated separately in Table 6. KL regularization is disabled in our experiments.

Advantages under binary rewards. Let $K = \sum_{i=1}^{G}(R_i + 1)/2$ denote the number of correct responses in a group, and let $p = K/G$ be its empirical success rate. For a mixed-reward group with $0 < K < G$, we have $\bar{R} = 2p - 1$ and $\sigma_R = 2\sqrt{p(1-p)}$. The advantages therefore take two values:

$$A_i = \begin{cases} \dfrac{2(1-p)}{2\sqrt{p(1-p)} + \delta}, & R_i = 1, \\ \dfrac{-2p}{2\sqrt{p(1-p)} + \delta}, & R_i = -1. \end{cases} \tag{7}$$

For groups with $K = 0$ or $K = G$, equivalently $\sum_{i=1}^{G} R_i \in \{-G, G\}$, all advantages are zero.

For $K \in \{0, G\}$, all advantages are zero. Otherwise, equally rewarded responses share the same advantage coefficient, regardless of their generation confidence. This motivates the confidence-based correction introduced next.

3 METHODOLOGY

ProGRPO uses confidence estimated from low-probability tokens to correct group-relative advantages before policy optimization.

3.1 CONFIDENCE ESTIMATION FROM LOW-PROBABILITY TOKENS

Prior work highlights the importance of token-level uncertainty in RLVR (Wang et al., 2025a). Motivated by these findings, we estimate confidence from the lowest-probability tokens in each sequence. This design is intended to reduce the influence of highly predictable tokens on the confidence score. Our selection criterion uses the probability assigned to the observed token, which is distinct from the entropy of the full next-token distribution.

For a nonempty sequence $s = (s_1, \ldots, s_{|s|})$ conditioned on context C, define

$$p_t(s \mid C) = \pi_{\theta_{\text{old}}}(s_t \mid C, s_{<t}). \tag{8}$$

Here, $|s|$ counts valid, non-padding tokens. We select the indices of the m_s lowest-probability tokens:

$$\mathcal{T}^{\text{low}}_{s|C} = \text{BottomK}\left(\{p_t(s \mid C)\}_{t=1}^{|s|}, m_s\right), \qquad m_s = \max\left(1, \lceil \rho|s| \rceil\right), \tag{9}$$

where BottomK returns token indices, with ties resolved by position. We use $\rho = 0.2$.

The confidence score is the geometric mean of the selected token probabilities:

$$c_{\text{old}}(s \mid C) = \exp\left(\frac{1}{|\mathcal{T}^{\text{low}}_{s|C}|} \sum_{t \in \mathcal{T}^{\text{low}}_{s|C}} \log p_t(s \mid C)\right). \tag{10}$$

Prompt confidence $c_{\text{old}}(q)$ scores the observed prompt tokens conditioned on any preceding system or chat prefix. Response confidence $c_{\text{old}}(o_i \mid q)$ scores the generated response conditioned on that prefix and the prompt. The low-probability token subsets are selected independently for the prompt and each response.

Both scores are computed once under the fixed old policy and held constant during subsequent policy updates. They measure token predictability; prompt confidence is not assumed to directly measure task difficulty. The selected subsets are used only for confidence estimation, while the policy objective includes all valid response tokens.

3.2 Confidence-Based Advantage Correction

Let A_i be the group-relative advantage defined in the preliminaries, and let $K = \sum_{i=1}^{G}(R_i + 1)/2$ be the number of correct responses in the group. Our Advantage Re-weighting Mechanism (ARM) adds a confidence-based correction:

$$\widetilde{A}_i = \begin{cases} A_i, & K \in \{0, G\}, \\ A_i + \alpha\big(c_{\text{old}}(q) - c_{\text{old}}(o_i \mid q)\big), & 0 < K < G, \end{cases} \tag{11}$$

where $\alpha \geq 0$ controls the correction strength. The resulting advantages are treated as fixed coefficients during policy optimization.

Relative weighting within a group. For two equally rewarded responses o_i and o_j in the same mixed-reward group, $A_i = A_j$ and the shared prompt term cancels:

$$\widetilde{A}_i - \widetilde{A}_j = \alpha\big(c_{\text{old}}(o_j \mid q) - c_{\text{old}}(o_i \mid q)\big). \tag{12}$$

For $\alpha > 0$, the less confidently generated response therefore receives a larger advantage coefficient. Among correct responses, this gives greater relative weight to lower-confidence samples. Among incorrect responses, it produces a less negative coefficient for lower-confidence samples, provided that advantage signs are preserved.

The prompt term supplies a shared, input-dependent offset within each group. Its contribution is distinct from the within-group ordering induced by response confidence, and is examined through the confidence-formulation ablations.

Groups with uniform rewards. When $K \in \{0, G\}$, all original advantages are zero. A nonzero confidence correction would then introduce an update driven solely by confidence. ARM therefore leaves these groups unchanged, preserving their zero policy-surrogate contribution.

Sign preservation. For mixed-reward groups, a sufficient condition for preserving the sign of each advantage is

$$\alpha\,|c_{\text{old}}(q) - c_{\text{old}}(o_i \mid q)| < |A_i|. \tag{13}$$

Under the binary rewards $R_i \in \{-1, 1\}$ and the population-standard-deviation normalization $A_i = (R_i - \bar{R})/(\sigma_R + \delta)$, we have

$$|A_i| \geq \frac{1}{\sqrt{G-1} + G\delta/2}, \qquad 0 < K < G. \tag{14}$$

Since both confidence scores lie in $[0, 1]$, a sufficient group-independent choice is

$$0 \leq \alpha < \frac{1}{\sqrt{G-1} + G\delta/2}. \tag{15}$$

Under the bound in Equation equation 15, correct responses retain positive advantages and incorrect responses retain negative advantages. ARM thus changes the coefficients of the policy-gradient updates while leaving the verifiable rewards unchanged.

3.3 ProGRPO Objective

We combine the ARM correction in Equation equation 11 with asymmetric clipping and token-mean aggregation within each prompt group. Define

$$\ell_{\varepsilon_{\text{low}},\varepsilon_{\text{high}}}(r, A) = \min\big(rA,\ \text{clip}(r, 1-\varepsilon_{\text{low}}, 1+\varepsilon_{\text{high}})A\big), \tag{16}$$

where $0 < \varepsilon_{\text{low}} < \varepsilon_{\text{high}}$. The larger upper threshold follows the Clip-Higher design (Yu et al., 2025).

The ProGRPO objective is

$$\mathcal{J}_{\text{ProGRPO}}(\theta) = \mathbb{E}_{\text{old}}\left[\frac{1}{\sum_{i=1}^{G}|o_i|}\sum_{i=1}^{G}\sum_{t=1}^{|o_i|}\ell_{\varepsilon_{\text{low}},\varepsilon_{\text{high}}}\big(r_{i,t}(\theta), \widetilde{A}_i\big)\right]. \tag{17}$$

Here, $\mathbb{E}_{\text{old}}$ averages over prompts $q \sim D$ and response groups sampled from $\pi_{\theta_{\text{old}}}(\cdot \mid q)$. The probability ratio $r_{i,t}(\theta)$ follows the definition in the preliminaries.

Each $|o_i|$ counts valid response tokens, excluding prompt and padding tokens. Within each group, this aggregation weights each response's mean token-level surrogate in proportion to its valid-token length. Group objectives are then averaged over sampled prompts.

The corrected advantage $\widetilde{A}_i$ is shared across the tokens of response o_i and held fixed; gradients flow only through $r_{i,t}(\theta)$. We minimize $\mathcal{L}_{\text{ProGRPO}}(\theta) = -\mathcal{J}_{\text{ProGRPO}}(\theta)$, without KL regularization or an entropy bonus.

Setting $\alpha = 0$ removes ARM while retaining the same asymmetric clipping and token-mean aggregation. This configuration corresponds to our GRPO with Clip-Higher baseline.

4 Experiments

4.1 Experimental Settings

Training data and model backbones. For mathematical reasoning, we train Qwen2.5-7B, Qwen2.5-32B, and DeepSeek-R1-Distill-Qwen-1.5B (Guo et al., 2025) on the DAPO dataset (Yu et al., 2025). For code generation, we train DeepSeek-R1-Distill-Qwen-7B (Guo et al., 2025) on the training split of DeepCoder (Luo et al., 2025).

Training setup. Unless otherwise specified, we set $\alpha = 0.3$ and $\rho = 0.2$. We sample $G = 8$ responses per prompt at a temperature of 1.0, with maximum prompt and response lengths of 2048 and 8192 tokens, respectively.

We optimize Equation 17 using token-mean aggregation, without reference-policy KL regularization or an entropy bonus. Optimization settings are listed in Table 1. Experiments run on two nodes, each equipped with eight NVIDIA A100 GPUs.

Controlled comparison for ARM. To evaluate the contribution of ARM, we compare ProGRPO with a matched GRPO with Clip-Higher baseline. The baseline uses the same token-mean aggregation, clipping thresholds, training budget, and remaining training settings. The only difference is the confidence correction: the baseline sets $\alpha = 0$, while ProGRPO enables ARM.

Table 1: Optimization settings.

Hyperparameter	Value
Training batch size	512
PPO mini-batch size	32
Clipping interval	$[0.8, 1.28]$
Optimizer	AdamW
Learning rate	1×10^{-6}
Epoch	1

4.2 Evaluation

Benchmarks. We evaluate mathematical reasoning on AIME 2024 (Li et al., 2024), AIME 2025 (Balunovic et al., 2025), AMC 2023 (Li et al., 2024), MATH-500 (Lightman et al., 2024), Minerva Math (Lewkowycz et al., 2022), and OlympiadBench (He et al., 2024). For code generation, we use LiveCodeBench (Jain et al., 2025), Codeforces (Quan et al., 2025), and HumanEval+ (Liu et al., 2023). We assess out-of-distribution generalization on GPQA (Rein et al., 2024) and MMLU-Pro (Wang et al., 2024).

Decoding settings. We use a sampling temperature of 0.6 and top-p of 0.95. The maximum output length is 8192 tokens for Qwen2.5 models. For DeepSeek-R1-Distill-Qwen models, the limits are 32768 tokens for reasoning and 8192 for code generation. All methods evaluated with the same backbone on the same benchmark use identical decoding settings and output-length limits.

Metrics. For mathematical reasoning, we report Pass@1, Pass@32, and Pass@k curves across sampling budgets. For code generation, we report Avg@16 and Pass@16 on LiveCodeBench, Avg@16 on HumanEval+, and ratings and percentiles on Codeforces. For GPQA, we report Avg@3.

4.3 Results

Table 2: Mathematical reasoning results. Each cell reports Pass@1 / Pass@32 (%). Average denotes the unweighted mean across six benchmarks. Base model denotes the checkpoint before RL training. The best result for each metric within each backbone block is bolded, including ties.

Method	AIME 2024	AIME 2025	AMC 2023	MATH-500	Minerva Math	OlympiadBench	Average
Qwen2.5-7B (maximum response length: 8192 tokens)							
Base model	5.4 / 30.0	2.5 / 20.0	32.4 / 82.5	54.7 / 92.6	22.0 / 54.4	24.6 / 63.1	23.6 / 57.1
GRPO	9.2 / 26.7	6.1 / 30.0	65.5 / 80.0	75.3 / 87.0	33.8 / 51.1	35.6 / 53.6	37.6 / 54.7
GRPO w/ KL-Cov	6.1 / 26.7	8.4 / 36.7	46.3 / 80.0	56.9 / 87.6	23.2 / **54.8**	25.7 / 60.2	27.8 / 57.7
GRPO w/ Clip-Higher	19.4 / 36.7	9.1 / 33.3	**67.3** / 87.5	73.9 / 89.4	32.9 / 53.3	37.5 / 61.0	40.0 / 60.2
20/80 Rule w/ Clip-Higher	10.7 / 40.0	13.6 / 26.7	65.8 / **92.5**	76.9 / 92.0	**34.9 / 54.8**	40.7 / 64.8	40.4 / 61.8
FlowRL	14.6 / 33.3	10.4 / 40.0	54.0 / 85.0	66.9 / 87.0	30.9 / 54.4	35.0 / 60.7	35.3 / 60.1
ProGRPO (ours)	**21.3 / 53.3**	**15.9 / 50.0**	67.2 / **92.5**	**80.5 / 94.2**	32.0 / 53.3	**42.7 / 67.5**	**43.3 / 68.5**
Qwen2.5-32B (maximum response length: 8192 tokens)							
Base model	5.7 / 43.3	2.2 / 26.7	29.7 / 87.5	52.0 / 91.8	27.3 / **59.6**	22.4 / 68.4	23.2 / 62.9
GRPO	16.8 / 43.3	13.9 / 26.7	80.1 / 95.0	**84.7** / 92.4	**42.3** / 55.9	49.7 / 65.1	47.9 / 63.1
ProGRPO (ours)	**34.6 / 60.0**	**24.8 / 46.7**	**80.6 / 97.5**	82.8 / **96.2**	39.3 / 58.5	**54.1 / 74.5**	**52.7 / 72.2**
DeepSeek-R1-Distill-Qwen-1.5B (maximum response length: 32768 tokens)							
Base model	31.6 / **76.7**	24.9 / **60.0**	71.6 / 95.0	74.3 / 95.6	26.1 / 56.3	49.1 / 80.1	46.3 / 77.3
GRPO	26.6 / 70.0	28.0 / 56.7	77.9 / **97.5**	80.6 / 96.6	29.9 / 56.3	53.1 / 79.2	49.4 / 76.1
ProGRPO (ours)	**46.0 / 76.7**	**33.6 / 60.0**	**86.3** / 95.0	**87.0 / 98.2**	**34.8 / 58.1**	**62.0 / 81.8**	**58.3 / 78.3**

Mathematical reasoning. ProGRPO achieves the highest average Pass@1 and Pass@32 among the evaluated methods on all three backbones (Table 2). On Qwen2.5-7B, it achieves 43.3% Pass@1 and 68.5% Pass@32, exceeding GRPO by 5.7 and 13.8 percentage points, respectively. Under otherwise identical training settings, adding ARM to GRPO with Clip-Higher improves Pass@1 and Pass@32 by 3.3 and 8.3 percentage points, respectively.

The improvements over GRPO extend to Qwen2.5-32B and DeepSeek-R1-Distill-Qwen-1.5B, with average Pass@1 / Pass@32 gains of 4.8 / 9.1 and 8.9 / 2.2 percentage points, respectively. The magnitude of improvement varies across backbones and benchmarks.

Code generation. ProGRPO achieves the best results on all reported metrics (Table 3). Compared with GRPO, it improves LiveCodeBench Avg@16 and Pass@16 by 1.53 and 0.36 percentage points, respectively, and HumanEval+ Avg@16 by 0.69 percentage points. On Codeforces, it gains 179.3 rating points, raising the percentile from 60.7 to 75.4.

Table 3: Code generation results for DeepSeek-R1-Distill-Qwen-7B. Avg@16 and Pass@16 are percentages; Pct. denotes percentile. Best results are bolded. *FlowRL uses weights obtained from Hugging Face.

Method	LiveCodeBench		Codeforces		HumanEval+
	Avg@16	Pass@16	Rating	Pct.	Avg@16
Base model	30.96	48.03	764.9	8.2	80.33
GRPO	34.94	53.76	1243.2	60.7	83.32
FlowRL*	33.98	51.97	1129.6	48.5	82.67
ProGRPO	**36.47**	**54.12**	**1422.5**	**75.4**	**84.01**

Tables 2 and 3 summarize results on mathematical reasoning and code generation, respectively.

Out-of-distribution evaluation. We evaluate general-domain transfer on MMLU-Pro and GPQA using Qwen2.5-7B (Table 4). ProGRPO achieves 54.3% on MMLU-Pro and 42.3% Avg@3 on GPQA, exceeding GRPO by 2.2 and 3.4 percentage points, respectively. It also outperforms the base model on both benchmarks.

ProGRPO achieves the highest or tied-highest Pass@k among the compared methods on all three benchmarks at every evaluated sampling budget up to $k = 128$ (Figure 1).

Table 4: OOD results for Qwen2.5-7B. Scores are percentages.

Method	MMLU-Pro	GPQA (Avg@3)
Base model	48.7	32.4
GRPO	52.1	38.9
ProGRPO	**54.3**	**42.3**

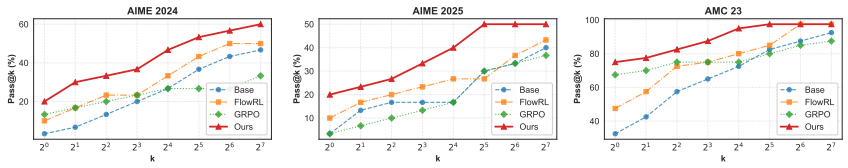


Figure 1: Pass@k comparison of the base model, GRPO, FlowRL, and ProGRPO on AIME 2024, AIME 2025, and AMC 2023 using Qwen2.5-7B. We conduct a separate evaluation with 128 responses per problem to assess whether ProGRPO retains its advantage at larger sampling budgets.

4.4 Analysis of Training

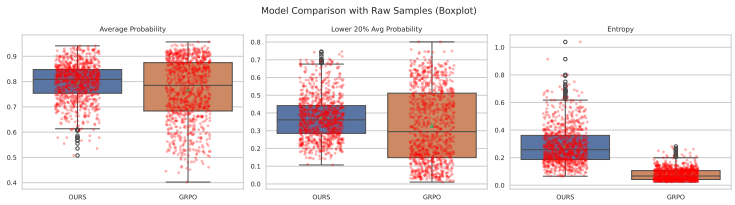
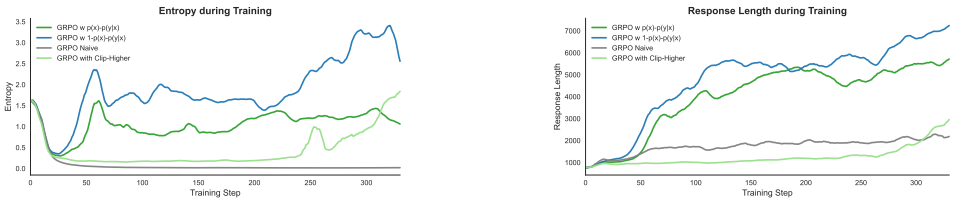


Figure 2: Generation statistics on AIME 2024 using 32 rollouts per problem. From left to right: average token probability, the bottom-20% token-probability statistic, and entropy. ProGRPO exhibits higher entropy and narrower interquartile ranges for the token-probability statistics than GRPO.

Generation statistics and output similarity. Figure 2 shows similar median average-token probabilities for ProGRPO and GRPO. ProGRPO has higher entropy and a narrower interquartile range for the bottom-20% token-probability statistic. Together with the similarity metrics and AIME 2024 heatmaps (Appendix D), these results suggest reduced output concentration.

Training dynamics. Figure 3 compares entropy and response length during training on Qwen2.5-7B. ProGRPO's entropy recovers after an initial decline, whereas vanilla GRPO remains at low entropy. During later training, ProGRPO maintains higher entropy and generates longer responses than vanilla GRPO.



(a) Training entropy.

(b) Response length.

Figure 3: Training dynamics on Qwen2.5-7B. ProGRPO maintains higher entropy and generates longer responses than vanilla GRPO during the later stages of training.

Diversity of correct responses. Table 5 reports similarity among correct responses on AIME 2024. Compared with GRPO, ProGRPO reduces Self-BLEU by 27.5% and semantic cosine similarity by 5.1%, indicating lower response similarity under both metrics. Semantic similarity is computed using all-MiniLM-L6-v2 embeddings (Reimers & Gurevych, 2019).

Table 5: Similarity of correct AIME 2024 responses. Lower is better; best values are bolded.

Method	Self-BLEU ↓	Sem. cosine ↓
GRPO	0.9299	0.9725
ProGRPO	**0.6746**	**0.9233**

4.5 Ablation Studies

We analyze ProGRPO on Qwen2.5-7B across six mathematical reasoning benchmarks, examining ARM, clipping, confidence formulations, and correction strength. We additionally report results from two saved checkpoints. Unless otherwise specified, scores are reported as Pass@1 / Pass@32 (%), with an unweighted average across benchmarks.

Table 6: ARM under different loss aggregations on Qwen2.5-7B. Each cell reports Pass@1 / Pass@32 (%). Average is the unweighted mean across six benchmarks. Standard clipping uses $[0.8, 1.2]$; Clip-Higher uses $[0.8, 1.28]$. Best results within each aggregation block are bolded, including ties.

Method	AIME 2024	AIME 2025	AMC 2023	MATH-500	Minerva Math	OlympiadBench	Average
Base model	5.4 / 30.0	2.5 / 20.0	32.4 / 82.5	54.7 / 92.6	22.0 / 54.4	24.6 / 63.1	23.6 / 57.1
Token-mean aggregation							
GRPO (standard clipping)	9.2 / 26.7	6.1 / 30.0	65.5 / 80.0	75.3 / 87.0	33.8 / 51.1	35.6 / 53.6	37.6 / 54.7
GRPO + ARM (standard clipping)	15.2 / 33.3	7.7 / 26.7	65.1 / **95.0**	76.2 / 93.0	**34.8 / 57.0**	40.1 / **69.4**	39.8 / 62.4
GRPO + Clip-Higher	19.4 / 36.7	9.1 / 33.3	**67.3** / 87.5	73.9 / 89.4	32.9 / 53.3	37.5 / 61.0	40.0 / 60.2
ProGRPO (ARM + Clip-Higher)	**21.3 / 53.3**	**15.9 / 50.0**	67.2 / 92.5	**80.5 / 94.2**	32.0 / 53.3	**42.7** / 67.5	**43.3 / 68.5**
Sample-mean aggregation							
GRPO + Clip-Higher	12.8 / 30.0	6.5 / **30.0**	67.8 / 85.0	74.4 / 87.6	32.1 / 51.8	36.5 / 55.8	38.3 / 56.7
ProGRPO (ARM + Clip-Higher)	**16.9 / 43.3**	**13.5 / 30.0**	**68.3 / 87.5**	**77.6 / 93.2**	**34.9 / 57.7**	**41.0 / 65.6**	**42.0 / 62.9**

(1)ARM, clipping, and loss aggregation. Table 6 compares ARM across clipping and loss-aggregation settings. With token-mean aggregation, adding ARM improves average Pass@1 /

Pass@32 by 2.2 / 7.7 percentage points under standard clipping and 3.3 / 8.3 under Clip-Higher. With sample-mean aggregation and Clip-Higher, the evaluated ProGRPO checkpoint exceeds its GRPO counterpart by 3.7 / 6.2 percentage points. These results show higher average performance with ARM under both aggregation schemes.

(2) Confidence formulation. Table 10 compares confidence-based corrections. The response-only formulation, $1 - c_{\text{old}}(o_i \mid q)$, achieves average Pass@1 / Pass@32 of 38.5% / 60.6%. The default correction, $c_{\text{old}}(q) - c_{\text{old}}(o_i \mid q)$, and the alternative, $1 - c_{\text{old}}(q) - c_{\text{old}}(o_i \mid q)$, induce the same ordering among equally rewarded responses within a mixed-reward group, but introduce different shared offsets. We adopt the default formulation for its higher average Pass@32: 68.5% versus 67.1%.

(3) Correction strength. Table 9 evaluates $\alpha \in \{0.3, 0.5, 0.7, 1.0\}$ under Clip-Higher, with GRPO reported separately for reference. We use $\alpha = 0.3$ as the shared default, as it yields the highest average Pass@32 on both backbones. On Qwen2.5-7B, increasing α to 0.5 gains 0.6 percentage points in average Pass@1 but loses 6.2 points in average Pass@32.

5 Related Work

RLVR and solution coverage. Reinforcement learning with verifiable rewards (RLVR) improves mathematical and code reasoning through automatically checked outcomes (Shao et al., 2024; Guo et al., 2025; Yu et al., 2025). However, improvements in single-sample accuracy do not necessarily translate into broader solution coverage at larger sampling budgets (Yue et al., 2025). Declining policy entropy and increasingly concentrated outputs motivate evaluating both Pass@1 and multi-sample performance (Cui et al., 2025).

Uncertainty-aware optimization. Existing methods encourage exploration through different components of policy optimization. Entropy regularization incorporates an entropy term into the objective (Haarnoja et al., 2018; Ziebart et al., 2008). Clip-Higher raises the upper clipping threshold (Yu et al., 2025), while high-entropy token selection restricts policy updates to selected positions (Wang et al., 2025a). More directly related to our approach, Cheng et al. (2026) introduce entropy-based advantage shaping, adding a detached, bounded entropy term to token-level advantages while preserving their signs. FlowRL instead learns to match a reward-induced response distribution (Zhu et al., 2026).

Sequence-level confidence correction. ProGRPO studies a sequence-level confidence signal for advantage correction. Unlike token-level entropy-based shaping, ARM uses the difference between prompt and response confidence, computed from old-policy probabilities at low-probability token positions. This produces a correction that can be positive or negative. Within mixed-reward groups, response confidence differentiates equally rewarded responses, while prompt confidence supplies a shared, input-dependent offset. The selected token positions are used only for confidence estimation; all valid response tokens remain in the policy objective. This design leaves verifiable rewards unchanged and allows the confidence correction to be evaluated separately from clipping and loss aggregation.

6 Conclusion

We introduced ProGRPO, which incorporates generation confidence into group-relative advantage estimation through an additive correction. ARM uses the difference between prompt and response confidence, estimated from old-policy probabilities at low-probability token positions. It leaves verifiable rewards unchanged, is disabled for uniform-reward groups, and preserves advantage signs under a sufficient bound on the correction magnitude.

Experiments on mathematical reasoning and code generation show improvements in average accuracy and multi-sample performance. Matched ablations on Qwen2.5-7B demonstrate benefits from ARM under both standard clipping and Clip-Higher. Training-entropy and response-similarity measurements provide complementary evidence consistent with reduced output concentration in the evaluated settings.

AI USE STATEMENT

We used generative AI tools solely to assist with translation, language polishing, and proofreading of the manuscript, including improvements to grammar, wording, and readability. All AI-assisted edits were reviewed by the authors to ensure accuracy and preserve the intended technical meaning. The authors take full responsibility for the final content of this work, including text produced with the assistance of generative AI.

ETHICS STATEMENT

This work investigates reinforcement learning methods for improving language model reasoning, with experiments focused on mathematical reasoning and code generation using existing datasets and pretrained models. The study does not involve human participants or the collection of new personal data. Improvements on these benchmarks do not establish the safety or reliability of model outputs in real-world applications: generated reasoning may remain incorrect, and generated code may contain security vulnerabilities even when it passes the available tests. Downstream use therefore requires application-specific evaluation and appropriate human oversight. Reuse of the datasets, models, and associated artifacts should respect their applicable licenses and access conditions.

REPRODUCIBILITY STATEMENT

To facilitate reproducibility, Sections 2 and 3 specify the optimization objectives, confidence estimator, advantage correction, and loss aggregation used in ProGRPO. Section 4.1 and Table 1 describe the training data sources, model backbones, and optimization settings. Section 4.2 and Appendix C document the evaluation benchmarks, decoding settings, sampling procedures, and correctness verification, including the computation of Pass@k. Appendix A states the assumptions underlying our analytical results and provides their proofs. Appendix B reports additional ablations, repeated evaluations of a fixed checkpoint across sampling seeds, and sign-preservation diagnostics.

REFERENCES

Mislav Balunovic, Jasper Dekoninck, Ivo Petrov, Nikola Jovanović, and Martin Vechev. MathArena: Evaluating LLMs on uncontaminated math competitions. In *Advances in Neural Information Processing Systems*, volume 38, pp. 22851–22888. Curran Associates, Inc., 2025. doi: 10.52202/085713-0679. Datasets and Benchmarks Track.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Daixuan Cheng, Shaohan Huang, Xuekai Zhu, Bo Dai, Xin Zhao, Zhenliang Zhang, and Furu Wei. Reasoning with exploration: An entropy perspective. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 30377–30385, 2026.

Ganqu Cui, Yuchen Zhang, Jiacheng Chen, Lifan Yuan, Zhi Wang, Yuxin Zuo, Haozhan Li, Yuchen Fan, Huayu Chen, Weize Chen, et al. The entropy mechanism of reinforcement learning for reasoning language models. *arXiv preprint arXiv:2505.22617*, 2025.

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

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pp. 1861–1870. Pmlr, 2018.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. OlympiadBench: A challenging benchmark for promoting AGI with olympiad-level bilingual multimodal scientific problems. In *Proceedings of the 62nd Annual Meeting of the Association*

for Computational Linguistics (Volume 1: Long Papers), pp. 3828–3850, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.211.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. LiveCodeBench: Holistic and contamination free evaluation of large language models for code. In *International Conference on Learning Representations*, volume 2025, pp. 58791–58831, 2025.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, et al. Tulu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*, 2024.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. Solving quantitative reasoning problems with language models. In *Advances in Neural Information Processing Systems*, volume 35, pp. 3843–3857, 2022. doi: 10.52202/068431-0278.

Jia Li, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Huang, Kashif Rasul, Longhui Yu, Albert Q Jiang, Ziju Shen, et al. Numinamath: The largest public dataset in ai4maths with 860k pairs of competition math problems and solutions. *Hugging Face repository*, 13(9):9, 2024.

Pengyi Li, Matvey Skripkin, Alexander Zubrey, Andrey Kuznetsov, and Ivan Oseledets. Confidence is all you need: Few-shot rl fine-tuning of language models. *arXiv preprint arXiv:2506.06395*, 2025.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations*, volume 2024, pp. 39578–39601, 2024.

Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. Is your code generated by chatgpt really correct? rigorous evaluation of large language models for code generation. *Advances in Neural Information Processing Systems*, 36:21558–21572, 2023.

Michael Luo, Sijun Tan, Roy Huang, Ameen Patel, Alpay Ariyak, Qingyang Wu, Xiaoxiang Shi, Rachel Xin, Colin Cai, Maurice Weber, Ce Zhang, Li Erran Li, Raluca Ada Popa, and Ion Stoica. DeepCoder: A fully open-source 14B coder at O3-mini level. Notion Blog, 2025.

Shanghaoran Quan, Jiaxi Yang, Bowen Yu, Bo Zheng, Dayiheng Liu, An Yang, Xuancheng Ren, Bofei Gao, Yibo Miao, Yunlong Feng, et al. Codeelo: Benchmarking competition-level code generation of llms with human-comparable elo ratings. *arXiv preprint arXiv:2501.01257*, 2025.

Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 3982–3992, Hong Kong, China, 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1410.

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. In *First Conference on Language Modeling*, 2024.

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, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xionghui Chen, Jianxin Yang, Zhenru Zhang, et al. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for llm reasoning. *arXiv preprint arXiv:2506.01939*, 2025a.

Yiping Wang, Qing Yang, Zhiyuan Zeng, Liliang Ren, Liyuan Liu, Baolin Peng, Hao Cheng, Xuehai He, Kuan Wang, Jianfeng Gao, Weizhu Chen, Shuohang Wang, Simon Shaolei Du, and Yelong Shen. Reinforcement learning for reasoning in large language models with one training example. In *Advances in Neural Information Processing Systems*, volume 38, pp. 122721–122764. Curran Associates, Inc., 2025b. doi: 10.52202/085713-4093.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, et al. MMLU-Pro: A more robust and challenging multi-task language understanding benchmark. In *Advances in Neural Information Processing Systems*, volume 37, pp. 95266–95290, 2024. doi: 10.52202/079017-3018. Datasets and Benchmarks Track.

Shihui Yang, Chengfeng Dou, Peidong Guo, Kai Lu, Qiang Ju, Fei Deng, and Rihui Xin. Dcpo: Dynamic clipping policy optimization. *arXiv preprint arXiv:2509.02333*, 2025.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Juncai Liu, et al. DAPO: An open-source LLM reinforcement learning system at scale. In *Advances in Neural Information Processing Systems*, volume 38, pp. 113222–113244. Curran Associates, Inc., 2025. doi: 10.52202/085713-3775.

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in LLMs beyond the base model? In *Advances in Neural Information Processing Systems*, volume 38, pp. 57654–57689. Curran Associates, Inc., 2025. doi: 10.52202/085713-1933.

Xinyu Zhu, Mengzhou Xia, Zhepei Wei, Wei-Lin Chen, Danqi Chen, and Yu Meng. The surprising effectiveness of negative reinforcement in llm reasoning. *arXiv preprint arXiv:2506.01347*, 2025.

Xuekai Zhu, Daixuan Cheng, Dinghuai Zhang, Hengli Li, Kaiyan Zhang, Che Jiang, Youbang Sun, Ermo Hua, Yuxin Zuo, Xingtai Lv, Qizheng Zhang, Lin Chen, Fanghao Shao, Bo Xue, Yunchong Song, Zhenjie Yang, Ganqu Cui, Ning Ding, Jianfeng Gao, Xiaodong Liu, Bowen Zhou, Hongyuan Mei, and Zhouhan Lin. FlowRL: Matching reward distributions for LLM reasoning. In *The Fourteenth International Conference on Learning Representations*, 2026.

Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, Anind K Dey, et al. Maximum entropy inverse reinforcement learning. In *Aaai*, volume 8, pp. 1433–1438. Chicago, IL, USA, 2008.

A Analytical Properties of ARM

We analyze ARM as an additive correction to group-relative advantages. The results characterize sign preservation, within-group weighting, and the resulting change in the clipped policy gradient.

A.1 Setup

For a prompt q, sample $G \geq 2$ responses $\{o_i\}_{i=1}^{G}$ from $\pi_{\theta_{\text{old}}}$. Each response receives a binary reward $R_i \in \{-1, +1\}$. Define

$$\bar{R} = \frac{1}{G}\sum_{i=1}^{G} R_i, \qquad \sigma_R = \sqrt{\frac{1}{G}\sum_{i=1}^{G}(R_i - \bar{R})^2}, \qquad A_i = \frac{R_i - \bar{R}}{\sigma_R + \delta}, \tag{18}$$

where $\delta > 0$. Let

$$K = \sum_{i=1}^{G} \frac{R_i + 1}{2}, \qquad h_q = \mathbf{1}\{0 < K < G\}, \tag{19}$$

so that $h_q = 1$ only for groups containing both correct and incorrect responses.

Write $c_q = c_{\text{old}}(q)$ and $c_i = c_{\text{old}}(o_i \mid q)$. These confidence scores lie in $[0, 1]$ and are computed under the old policy using the low-probability token selection defined in the main text. They measure observed-token predictability, rather than answer correctness or task difficulty.

ARM adds the correction

$$\Delta_i = \alpha h_q (c_q - c_i), \qquad \widetilde{A}_i = A_i + \Delta_i, \qquad \alpha \geq 0. \tag{20}$$

The correction can be positive or negative. Both A_i and Δ_i remain fixed during policy optimization. For uniform-reward groups, $A_i = \Delta_i = \widetilde{A}_i = 0$.

A.2 Sign Preservation

Proposition 1 (Sign preservation)**.** *For a mixed-reward group, the condition*

$$|\Delta_i| < |A_i| \tag{21}$$

ensures that $\widetilde{A}_i$ and A_i have the same sign. A sufficient bound valid for every mixed-reward group of size G is

$$0 \leq \alpha < \frac{1}{\sqrt{G-1} + G\delta/2}. \tag{22}$$

Proof. Adding a perturbation smaller than $|A_i|$ cannot change its sign. For the uniform bound, let $m = \min(K, G-K) \geq 1$. The binary-reward normalization gives

$$\begin{aligned} \min_i |A_i| &= \frac{2m}{2\sqrt{m(G-m)} + G\delta} \\ &= \frac{1}{\sqrt{(G-m)/m} + G\delta/(2m)} \\ &\geq \frac{1}{\sqrt{G-1} + G\delta/2}. \end{aligned} \tag{23}$$

Since $|c_q - c_i| \leq 1$, we have $|\Delta_i| \leq \alpha$. The stated bound therefore implies $|\Delta_i| < |A_i|$ for every response. □

Under this condition, correct responses retain positive advantages and incorrect responses retain negative advantages. The bound is sufficient, not necessary: exceeding it does not by itself imply sign reversal.

A.3 WITHIN-GROUP CORRECTIONS

Proposition 2 (Relative correction)**.** *For two responses in the same group,*

$$\Delta_i - \Delta_j = \alpha h_q(c_j - c_i). \tag{24}$$

Consequently,

$$\widetilde{A}_i - \widetilde{A}_j = A_i - A_j + \alpha h_q(c_j - c_i). \tag{25}$$

For equally rewarded responses in a mixed-reward group,

$$\widetilde{A}_i - \widetilde{A}_j = \alpha(c_j - c_i). \tag{26}$$

Proof. Subtracting the two corrections cancels the shared prompt term. Equal rewards imply $A_i = A_j$. □

For $\alpha > 0$, a lower-confidence response receives a larger correction than an equally rewarded, higher-confidence response. Under sign preservation, this gives a larger positive advantage among correct responses and a less negative advantage among incorrect responses.

Proposition 3 (Mean correction)**.** *Let $\bar{c} = G^{-1}\sum_i c_i$. Then*

$$\frac{1}{G}\sum_{i=1}^{G}\widetilde{A}_i = \frac{1}{G}\sum_{i=1}^{G}\Delta_i = \alpha h_q(c_q - \bar{c}). \tag{27}$$

Proof. The original advantages satisfy $\sum_i A_i = 0$. Averaging the additive correction gives the result. □

Prompt confidence supplies a shared offset without changing the ordering among equally rewarded responses. This identity describes an unweighted mean of coefficients; it does not determine the magnitude of the group's policy update.

A.4 GRADIENT CORRECTION AND CLIPPING

We compare ARM with an uncorrected baseline at the same policy parameters, using the same sampled group, old policy, clipping thresholds, and token-mean normalization. Assume throughout this subsection that advantage signs are preserved.

Let $L = \sum_i |o_i|$ be the number of valid response tokens. The two group objectives are

$$\widehat{\mathcal{J}}_q^{\text{base}}(\theta) = \frac{1}{L}\sum_{i=1}^{G}\sum_{t=1}^{|o_i|}\ell(r_{i,t}(\theta), A_i), \tag{28}$$

$$\widehat{\mathcal{J}}_q^{\text{ARM}}(\theta) = \frac{1}{L}\sum_{i=1}^{G}\sum_{t=1}^{|o_i|}\ell(r_{i,t}(\theta), A_i + \Delta_i), \tag{29}$$

where

$$r_{i,t}(\theta) = \frac{\pi_\theta(o_{i,t} \mid q, o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t} \mid q, o_{i,<t})}, \tag{30}$$

$$\ell(r, A) = \min\big(rA, \text{clip}(r, l, u)A\big). \tag{31}$$

Here, $l = 1 - \varepsilon_{\text{low}}$ and $u = 1 + \varepsilon_{\text{high}}$, with $0 < \varepsilon_{\text{low}} < 1$ and $\varepsilon_{\text{high}} > 0$.

For brevity, write $r_{i,t} = r_{i,t}(\theta)$ and

$$g_{i,t} = \nabla_\theta \log \pi_\theta(o_{i,t} \mid q, o_{i,<t}). \tag{32}$$

Then $\nabla_\theta r_{i,t} = r_{i,t} g_{i,t}$.

Unchanged clipping decisions. In a mixed-reward group, sign preservation ensures that both methods use positive advantages for correct responses and negative advantages for incorrect responses. For ratios not exactly at the clipping thresholds, their common gradient mask is

$$M_{i,t} = \begin{cases} 0, & R_i = +1, \quad r_{i,t} > u, \\ 0, & R_i = -1, \quad r_{i,t} < l, \\ 1, & \text{otherwise.} \end{cases} \tag{33}$$

Within each reward class, this mask depends on the ratio and clipping thresholds, not on the advantage magnitude. Thus, at a fixed ratio, adding Δ_i does not change the clipping decision when the advantage sign is preserved.

Proposition 4 (Additive gradient correction)**.** *Assume policy probabilities are positive and differentiable. For a mixed-reward group with preserved advantage signs, and for ratios not exactly at the clipping thresholds,*

$$\nabla_\theta \ell(r_{i,t}, A_i + \Delta_i) = \nabla_\theta \ell(r_{i,t}, A_i) + M_{i,t} \Delta_i r_{i,t} g_{i,t}. \tag{34}$$

Consequently,

$$\nabla_\theta \widehat{\mathcal{J}}_q^{\mathrm{ARM}} = \nabla_\theta \widehat{\mathcal{J}}_q^{\mathrm{base}} + \frac{\alpha h_q}{L} \sum_{i=1}^{G} \sum_{t=1}^{|o_i|} M_{i,t} r_{i,t} (c_q - c_i) g_{i,t}. \tag{35}$$

Proof. Sign preservation ensures that both surrogates select the same branch. On the active branch, their difference is $\Delta_i r_{i,t}$, whose gradient is $\Delta_i r_{i,t} g_{i,t}$ because Δ_i is fixed. On the selected clipped branch, their difference is constant with respect to θ, so its gradient is zero. Summing over tokens and substituting $\Delta_i = \alpha h_q (c_q - c_i)$ gives the group-gradient identity. □

The correction therefore strengthens or weakens existing gradient contributions without activating terms that are clipped at the same policy parameters. Its clipping mask is inherited from the original advantage; the correction is not clipped independently as a separate advantage.

At the old policy. When $\theta = \theta_{\mathrm{old}}$, all ratios equal one and lie inside (l, u). The gradient becomes

$$\nabla_\theta \widehat{\mathcal{J}}_q^{\mathrm{ARM}}(\theta)\Big|_{\theta=\theta_{\mathrm{old}}} = \frac{1}{L} \sum_{i=1}^{G} \sum_{t=1}^{|o_i|} \big[A_i + \alpha h_q (c_q - c_i)\big] g_{i,t}(\theta_{\mathrm{old}}). \tag{36}$$

ARM and Clip-Higher. ARM changes advantage coefficients through the additive correction Δ_i. Clip-Higher changes the upper threshold u, allowing positive-advantage terms to remain active over a larger range of probability ratios. They therefore modify different parts of the surrogate gradient.

Scope. These identities compare the two objectives at the same policy parameters. Different corrections across responses can change the summed gradient and subsequent policy updates, so clipping patterns may differ later in training. The analysis does not establish monotonic entropy growth or convergence to a uniform distribution over correct responses. Uniform-reward groups contribute zero to both policy surrogates and their gradients.

B Additional Results

This appendix provides additional ablations, training diagnostics, and PPO experiments for the Advantage Re-weighting Mechanism (ARM). Unless otherwise stated, experiments use Qwen2.5-7B and evaluate six mathematical reasoning benchmarks: AIME 2024, AIME 2025, AMC 2023, MATH-500, Minerva Math, and OlympiadBench. Average scores are unweighted means across these benchmarks.

Notation. For a prompt q, let $\{o_i\}_{i=1}^{G}$ denote a group of sampled responses, with $G \geq 2$. In the GRPO experiments, rewards satisfy $R_i \in \{-1, +1\}$, and

$$\bar{R} = \frac{1}{G}\sum_{i=1}^{G} R_i, \qquad \sigma_R = \sqrt{\frac{1}{G}\sum_{i=1}^{G}(R_i - \bar{R})^2}, \qquad A_i = \frac{R_i - \bar{R}}{\sigma_R + \delta}, \tag{37}$$

where $\delta > 0$. Let

$$K = \sum_{i=1}^{G} \frac{R_i + 1}{2}, \qquad h_q = \mathbf{1}\{0 < K < G\}. \tag{38}$$

Writing prompt and response confidence as c_q and c_i, respectively, ARM applies

$$d_i = c_q - c_i, \qquad \widetilde{A}_i = A_i + \alpha h_q d_i, \qquad \alpha \geq 0. \tag{39}$$

Confidence scores are computed under the old policy and held fixed during optimization. For uniform-reward groups, $h_q = 0$ and $A_i = \widetilde{A}_i = 0$.

B.1 Confidence Estimation

We compare confidence estimates based on all valid tokens with estimates based on the lowest-probability 20% of tokens. For a nonempty sequence s conditioned on context C, define

$$p_t = \pi_{\theta_{\text{old}}}(s_t \mid C, s_{<t}), \qquad c_{\mathcal{T}}(s \mid C) = \exp\left(\frac{1}{|\mathcal{T}|}\sum_{t \in \mathcal{T}} \log p_t\right). \tag{40}$$

Full-sequence estimation uses $\mathcal{T} = \{1, \ldots, |s|\}$. Bottom-20% estimation uses the indices of the $m_s = \max(1, \lceil 0.2|s| \rceil)$ lowest-probability tokens, with ties resolved by position. Token sets are selected separately for the prompt and each response.

Both estimators average log probabilities over their selected token positions. This choice concerns confidence estimation, rather than policy-loss aggregation. Low observed-token probability is distinct from high predictive entropy.

Table 7: Confidence estimation on Qwen2.5-7B using all valid tokens or the lowest-probability 20%. Each cell reports Pass@1 / Pass@32 (%). Average is the unweighted benchmark mean. Δ reports bottom-20% minus full-sequence results in percentage points.

Estimator	AIME 24	AIME 25	AMC 23	MATH 500	Minerva	Olympiad.	Avg
Full sequence	16.6/36.7	10.3/33.3	65.4/90.0	78.5/92.8	**35.0/56.6**	41.5/67.2	41.2/62.8
Bottom 20% (ours)	**21.3/53.3**	**15.9/50.0**	**67.2/92.5**	**80.5/94.2**	32.0/53.3	**42.7/67.5**	**43.3/68.5**
Δ	+4.7/+16.6	+5.6/+16.7	+1.8/+2.5	+2.0/+1.4	−3.0/−3.3	+1.2/+0.3	+2.1/+5.7

Bottom-20% confidence estimation improves average Pass@1 and Pass@32 over full-sequence estimation by 2.1 and 5.7 percentage points, respectively. It achieves higher scores on five of the six benchmarks. Minerva Math is the exception: full-sequence estimation yields higher Pass@1 and Pass@32 by 3.0 and 3.3 percentage points, respectively.

Early-training confidence statistics. Table 8 reports confidence statistics at selected steps during the first 33 training steps on DAPO, using $G = 8$ responses per prompt. Under full-sequence estimation, mean response confidence increases from 0.934 to 0.975, while mean prompt confidence remains near 0.89. The standard deviation of the recorded confidence gap decreases from 0.129 to 0.038. Bottom-20% estimation yields a larger gap standard deviation at every reported step, including 0.107 at step 33. These observations indicate greater dispersion in the recorded confidence gaps under bottom-20% estimation.

Table 8: Confidence statistics during the first 33 training steps on Qwen2.5-7B. Prompt and response columns report mean confidence; gap columns report the mean and standard deviation of the recorded signed confidence difference.

	Full sequence				Bottom 20%			
Step	**Prompt**	**Response**	**Gap mean**	**Gap SD**	**Prompt**	**Response**	**Gap mean**	**Gap SD**
1	0.896	0.934	−0.037	0.129	0.573	0.779	−0.206	0.199
5	0.893	0.939	−0.046	0.108	0.570	0.775	−0.205	0.201
10	0.891	0.956	−0.065	0.055	0.560	0.824	−0.264	0.147
15	0.892	0.963	−0.071	0.050	0.562	0.847	−0.285	0.123
20	0.889	0.967	−0.079	0.038	0.559	0.860	−0.301	0.110
25	0.885	0.974	−0.089	0.038	0.542	0.861	−0.319	0.102
30	0.887	0.973	−0.086	0.032	0.547	0.860	−0.312	0.104
33	0.889	0.975	−0.086	0.038	0.553	0.863	−0.309	0.107

B.2 Sensitivity to the Correction Strength

We evaluate $\alpha \in \{0.3, 0.5, 0.7, 1.0\}$ under Clip-Higher on Qwen2.5-7B and DeepSeek-R1-Distill-Qwen-1.5B. GRPO results are included separately as reference baselines.

Across both backbones, $\alpha = 0.3$ achieves the highest average Pass@32 among the tested values. On Qwen2.5-7B, increasing α to 0.5 improves average Pass@1 by 0.6 percentage points but reduces Pass@32 by 6.2 percentage points. These results support using $\alpha = 0.3$ as a shared default across the two backbones.

Table 9: Sensitivity to the ARM coefficient α under Clip-Higher across two backbones. GRPO baselines are included for reference. Each cell reports Pass@1 / Pass@32 (%). Average is the unweighted mean across six benchmarks. Best results for each metric within each backbone are bolded, including ties.

Setting	**AIME 2024**	**AIME 2025**	**AMC 2023**	**MATH-500**	**Minerva Math**	**OlympiadBench**	**Average**
Qwen2.5-7B							
GRPO (ref.)	9.2 / 26.7	6.1 / 30.0	65.5 / 80.0	75.3 / 87.0	33.8 / 51.1	35.6 / 53.6	37.6 / 54.7
$\alpha = 0.3$	**21.3** / **53.3**	**15.9** / **50.0**	67.2 / **92.5**	80.5 / **94.2**	32.0 / 53.3	42.7 / **67.5**	43.3 / **68.5**
$\alpha = 0.5$	19.4 / 36.7	12.2 / 40.0	**72.3** / 85.0	**81.4** / 92.4	34.5 / **54.8**	**43.9** / 64.7	**43.9** / 62.3
$\alpha = 0.7$	15.1 / 33.3	9.2 / 36.7	66.6 / 82.5	77.3 / 92.4	**36.0** / 50.7	41.3 / 62.9	40.9 / 59.8
$\alpha = 1.0$	13.9 / 26.7	3.8 / 23.3	63.8 / 85.0	75.9 / 88.4	35.7 / 52.9	40.8 / 60.5	39.0 / 56.2
DeepSeek-R1-Distill-Qwen-1.5B							
GRPO (ref.)	26.6 / 70.0	28.0 / 56.7	77.9 / **97.5**	80.6 / 96.6	29.9 / 56.3	53.1 / 79.2	49.4 / 76.1
$\alpha = 0.3$	**46.0** / **76.7**	**33.6** / 60.0	**86.3** / 95.0	**87.0** / **98.2**	**34.8** / **58.1**	**62.0** / **81.8**	**58.3** / **78.3**
$\alpha = 0.5$	37.9 / 63.3	29.0 / 60.0	82.7 / **97.5**	82.5 / 95.6	30.8 / 53.3	59.5 / 80.1	53.7 / 75.0
$\alpha = 0.7$	38.1 / 70.0	24.2 / 53.3	78.0 / 95.0	83.3 / 97.4	29.1 / 56.3	58.6 / 79.2	51.9 / 75.2
$\alpha = 1.0$	29.8 / 66.7	24.7 / **63.3**	73.3 / **97.5**	69.7 / 96.4	21.4 / 50.7	44.6 / 79.2	43.9 / 75.6

Correction strength across backbones. Table 9 compares ARM coefficients on Qwen2.5-7B and DeepSeek-R1-Distill-Qwen-1.5B. Across both backbones, $\alpha = 0.3$ achieves the highest average Pass@32 among the evaluated settings, reaching 68.5% and 78.3%, respectively. It also yields the highest average Pass@1 on the distilled model. On Qwen2.5-7B, increasing α to 0.5 improves average Pass@1 by 0.6 percentage points but reduces Pass@32 by 6.2 percentage points. These results support using $\alpha = 0.3$ as a shared default across the two evaluated backbones.

B.3 Confidence Correction Formulations

We compare three additive correction signals:

$$d_i^{\text{resp}} = 1 - c_i, \qquad d_i^{\text{alt}} = 1 - c_q - c_i, \qquad d_i^{\text{default}} = c_q - c_i. \tag{41}$$

Each signal is used in a correction of the form $\alpha h_q d_i$. For $\alpha > 0$ in mixed-reward groups, all three formulations assign a larger correction to a lower-confidence response. They differ in the offset shared by responses within the group.

The response-only formulation achieves 38.5% average Pass@1 and 60.6% Pass@32. The alternative formulation achieves 44.3% / 67.1%, while the default achieves 43.3% / 68.5%. The alternative therefore yields higher Pass@1, whereas the default yields higher Pass@32. We retain the default formulation for its stronger multi-sample performance in this comparison.

Table 10: Confidence correction ablation on Qwen2.5-7B. Entries report Pass@1 / Pass@32 (%); Average is the unweighted benchmark mean. Best results are bolded.

Setting / correction d_i	**AIME 2024**	**AIME 2025**	**AMC 2023**	**MATH-500**	**Minerva Math**	**OlympiadBench**	**Average**
Base model	5.4 / 30.0	2.5 / 20.0	32.4 / 82.5	54.7 / 92.6	22.0 / 54.4	24.6 / 63.1	23.6 / 57.1
GRPO (reference)	9.2 / 26.7	6.1 / 30.0	65.5 / 80.0	75.3 / 87.0	33.8 / 51.1	35.6 / 53.6	37.6 / 54.7
Response only: $1 - c_i$	12.5 / 33.3	9.7 / 30.0	66.3 / **95.0**	73.7 / 91.2	32.7 / 52.9	36.0 / 61.1	38.5 / 60.6
Alternative: $1 - c_q - c_i$	16.6 / 43.3	15.2 / 46.7	**70.0** / 90.0	**81.8 / 95.0**	**36.4 / 57.4**	**45.8 / 70.0**	**44.3** / 67.1
Default: $c_q - c_i$	**21.3 / 53.3**	**15.9 / 50.0**	67.2 / 92.5	80.5 / 94.2	32.0 / 53.3	42.7 / 67.5	43.3 / **68.5**

Constant versus prompt-confidence reference. In Table 10, replacing the constant reference in $1 - c_i$ with prompt confidence, $c_q - c_i$, improves average Pass@1 and Pass@32 from 38.5% / 60.6% to 43.3% / 68.5%. The gains are 4.8 and 7.9 percentage points, respectively, favoring prompt confidence over the tested constant reference of 1.

Small-scale ablation of confidence components. We additionally conduct a separate, small-scale ablation to examine the contributions of prompt and response confidence. We train three variants on DAPO-17K, using prompt confidence only, response confidence only, or both, and report Pass@1 and Pass@4 (Table 11).

Table 11: Small-scale ablation of confidence components of training on DAPO-17K. Each cell reports Pass@1 / Pass@4 (%). Average is the unweighted mean across six benchmarks. Best results for each metric are bolded, including ties.

Benchmark	**Prompt only**	**Response only**	**Both**
AIME 2024	13.33 / **20.00**	**14.17 / 20.00**	12.50 / **20.00**
AIME 2025	5.00 / 13.33	4.17 / 16.67	**10.83 / 23.33**
AMC 2023	**57.50** / 72.50	**57.50** / 72.50	**57.50 / 75.00**
MATH-500	75.00 / 85.80	76.20 / **88.80**	**76.55** / 86.80
Minerva Math	35.66 / 46.32	35.11 / 46.69	**35.75 / 48.16**
OlympiadBench	40.69 / 54.45	39.39 / **54.90**	**40.73** / 54.30
Average	37.86 / 48.73	37.76 / 49.93	**38.98 / 51.27**

Using both confidence terms achieves the highest average Pass@1 and Pass@4, at 38.98% and 51.27%, respectively. Compared with response confidence alone, the gains are 1.22 and 1.34 percentage points. The gains vary across benchmarks: response confidence alone yields higher Pass@1 on AIME 2024 and higher Pass@4 on MATH-500 and OlympiadBench. These results favor the joint formulation under this limited training budget.

B.4 Comparison with Entropy-Based Advantage Shaping

Table 12 presents reported results for GRPO with entropy-based advantage shaping and ProGRPO on Qwen2.5-7B. The comparison is descriptive, as training and evaluation settings are not fully matched.

Table 12: Reported results on Qwen2.5-7B (%). GRPO w/ Entropy Adv. results are taken from Cheng et al. (2026). Higher reported scores are bolded, including ties.

Method	AIME 2025		AIME 2024		AMC 2023	
	Pass@256	Pass@1	Pass@256	Pass@1	Pass@128	Pass@1
GRPO w/ Entropy Adv.	**53.3**	11.8	56.7	12.6	91.6	57.8
ProGRPO	**53.3**	**15.1**	**60.0**	**20.5**	**97.5**	**67.5**

Results are from separate studies with potentially different training and evaluation settings. The reference reports AMC 2023 Pass@128 rather than Pass@256, so the corresponding entry is omitted.

B.5 Variation across Evaluation Seeds

We evaluate the same Qwen2.5-7B ProGRPO checkpoint using sampling seeds 41, 42, 43, and 44. The primary results correspond to seed 42. Table 13 summarizes all four evaluations. For each seed, we compute the unweighted average across the six benchmarks before calculating the mean and standard deviation across seeds. These measurements quantify evaluation variability for a fixed checkpoint.

Table 13: Repeated evaluation of the Qwen2.5-7B ProGRPO checkpoint. Values are mean ± sample standard deviation (%) across evaluation seeds 41, 42, 43, and 44.

Benchmark	Pass@1	Pass@32
AIME 2024	20.20 ± 0.81	44.99 ± 5.76
AIME 2025	15.64 ± 0.28	40.00 ± 6.67
AMC 2023	66.98 ± 0.88	93.75 ± 1.44
MATH-500	80.54 ± 0.15	94.10 ± 0.12
Minerva Math	31.93 ± 0.09	54.41 ± 1.16
OlympiadBench	42.60 ± 0.11	66.77 ± 0.79
Average	42.99 ± 0.30	65.68 ± 1.89

B.6 Sign-Preservation Diagnostics

A sufficient condition. Since $c_q, c_i \in [0, 1]$, the confidence gap satisfies $|d_i| \leq 1$. For a mixed-reward group, a sufficient condition for preserving each advantage's sign is

$$\alpha|d_i| < |A_i|. \tag{42}$$

A bound independent of the observed confidence values follows from the binary-reward normalization. Let $m = \min(K, G - K) \geq 1$. Then

$$\begin{aligned} \min_i |A_i| &= \frac{2m}{2\sqrt{m(G-m)} + G\delta} \\ &= \frac{1}{\sqrt{(G-m)/m} + G\delta/(2m)} \\ &\geq \frac{1}{\sqrt{G-1} + G\delta/2}. \end{aligned} \tag{43}$$

Consequently,

$$0 \le \alpha < \frac{1}{\sqrt{G-1} + G\delta/2} \tag{44}$$

is sufficient for every mixed-reward group of size G.

For $G = 8$, the upper bound is $1/(\sqrt{7} + 4\delta)$. In particular, $\alpha = 0.3$ satisfies the condition whenever

$$0 < \delta < \frac{1/0.3 - \sqrt{7}}{4} \approx 0.1719. \tag{45}$$

This guarantee follows from the bounded confidence scores and does not require an empirical bound estimated from their means and standard deviations.

Recorded sign-flip rates. For N recorded responses, define

$$\mathrm{FlipRate} = \frac{100}{N} \sum_{i=1}^{N} \mathbf{1}\left\{\mathrm{sign}(\widetilde{A}_i) \neq \mathrm{sign}(A_i)\right\}. \tag{46}$$

Table 14 reports response-level sign-flip rates on Qwen2.5-7B, aggregated over the first 33 training steps. All four correction strengths yield a reported rate of 0.00%. This is consistent with the sufficient sign-preservation condition: exceeding its bound does not necessarily cause a sign reversal. These measurements cover only the recorded early-training interval. They do not establish sign preservation later in training or explain the performance differences across correction strengths.

Table 14: Sign-flip rates under binary-reward GRPO during the first 33 training steps.

α	0.3	0.5	0.7	1.0
Flip rate (%)	0.00	0.00	0.00	0.00

B.7 Extension to PPO with GAE

Generalized Advantage Estimation (GAE) produces token-level advantage coefficients whose magnitudes can be arbitrarily small. A fixed additive confidence correction therefore need not preserve their signs, even when it is suitable for group-normalized binary rewards.

Adaptive correction. For a fixed scalar advantage A and confidence gap d, consider

$$\alpha_{\mathrm{eff}}(A, d) = \min\left(\alpha, \frac{\beta|A|}{|d| + \epsilon_c}\right), \qquad \widetilde{A} = A + \alpha_{\mathrm{eff}}(A, d)d, \tag{47}$$

where $\alpha \ge 0$, $0 < \beta < 1$, and $\epsilon_c > 0$. For the adaptive correction in our PPO experiments, we set $\alpha = 0.1$, $\beta = 0.9$, and $\epsilon_c = 10^{-8}$. The confidence gap and corrected advantages are held fixed during policy optimization.

For any $A \neq 0$,

$$|\alpha_{\mathrm{eff}}(A, d)d| \le \beta|A|\frac{|d|}{|d| + \epsilon_c} \le \beta|A| < |A|. \tag{48}$$

Thus, the correction preserves the sign of A. If $A = 0$, the adaptive coefficient is zero, so $\widetilde{A} = 0$.

For token-level GAE, let $\Delta_{i,t} = \widetilde{A}_{i,t} - A_{i,t}$ denote the applied correction. Sign preservation holds if

$$|\Delta_{i,t}| \le \beta|A_{i,t}|, \qquad 0 < \beta < 1, \qquad t \in \mathcal{V}_i, \tag{49}$$

where $\mathcal{V}_i$ contains the valid response-token positions. When $A_{i,t} = 0$, this condition requires $\Delta_{i,t} = 0$. The guarantee applies to individual advantage signs and does not determine the direction of the aggregated parameter update.

Table 15: PPO with GAE and adaptive ARM on Qwen2.5-7B, trained for two epochs on DAPO-17K. Each cell reports Pass@1 / Pass@16 (%). Average is the unweighted mean across the six benchmarks. Best results are bolded, including ties.

Method	AIME 2024	AIME 2025	AMC 2023	MATH-500	Minerva Math	OlympiadBench	Average
PPO + GAE	**13.65** / 26.67	7.50 / **30.00**	53.05 / **90.00**	75.29 / **93.60**	**35.17** / 53.68	40.12 / **67.21**	37.46 / 60.19
+ Adaptive ARM	13.54 / **33.33**	**9.38** / **30.00**	**57.50** / **90.00**	**76.96** / 92.20	34.56 / **56.25**	**42.05** / 66.47	**39.00** / **61.38**

Experimental results. We evaluate PPO with GAE and an adaptive ARM variant on Qwen2.5-7B, trained for two epochs on DAPO-17K. Table 15 reports Pass@1 and Pass@16 on the six mathematical reasoning benchmarks.

Average Pass@1 / Pass@16 increases from 37.46 / 60.19 to 39.00 / 61.38. The gains vary across benchmarks: ARM improves Pass@1 on four of the six benchmarks, while Pass@16 increases on AIME 2024 and Minerva Math, remains unchanged on AIME 2025 and AMC 2023, and decreases on MATH-500 and OlympiadBench.

Sign-flip diagnostics. Table 16 reports token-level sign-flip rates for PPO with GAE on Qwen2.5-7B. The fixed-coefficient diagnostic applies $\alpha = 0.3$ and compares the signs of $A + 0.3d$ and A. It yields a peak per-step rate of 7.29% and a mean rate of 2.41%; both reported rates are zero after adaptive correction. These measurements show that a fixed correction can reverse advantage signs on the recorded data, whereas no reversals are reported after adaptation. The performance comparison evaluates PPO with the combined adaptive ARM variant.

Table 16: Token-level sign-flip rates (%) for PPO with GAE.

Statistic	Fixed α	Adaptive
Peak per step	7.29	0.00
Mean	2.41	0.00

C Pass@1 and Pass@k Computation

For each of the N benchmark problems, we independently sample n responses under fixed decoding settings. Let $c_j(k)$ denote the number of correct responses among the first k sampled responses to problem j, where $1 \le k \le n$. Responses are considered in their original sampling order.

For the Pass@k curves, we report empirical solution coverage:

$$\widehat{\text{Pass@}k} = \frac{1}{N} \sum_{j=1}^{N} \mathbf{1}\{c_j(k) > 0\}\,. \tag{50}$$

A problem is counted as solved if at least one of its first k responses is correct. Thus, this metric measures the fraction of benchmark problems solved within k sampling attempts. At $k = n$, it is the fraction of problems with at least one correct response among all n samples and coincides with the $k = n$ case of the estimator in Chen et al. (2021).

For the primary Pass@1 results, we estimate single-sample accuracy by averaging correctness over all n sampled responses per problem:

$$\widehat{\text{Pass@1}} = \frac{1}{N} \sum_{j=1}^{N} \frac{c_j(n)}{n}.$$

This equals the $k = 1$ case of the estimator in Chen et al. (2021). In contrast, the $k = 1$ point on the empirical coverage curves uses only the first sampled response to each problem. Both estimate single-sample accuracy, but their realized values may differ because they use different samples and aggregation procedures.

For intermediate budgets $1 < k < n$, the curves use the direct coverage statistic in Equation (50), rather than the combinatorial estimator that averages over all subsets of k responses from a pool of n samples. The same sample pool is reused across the evaluated budgets. Reusing samples does not by itself establish statistical significance.

Sampling configuration. For the primary mathematical reasoning results, we sample 32 responses per problem. Pass@1 is the average correctness over these responses, and Pass@32 is the fraction of problems with at least one correct response in the same pool.

For the separate Pass@k curves, we conduct an independent evaluation with 128 responses per problem and compute empirical coverage using the first k responses at each evaluated budget. Consequently, results at the same sampling budget may differ between the primary evaluation and the curve evaluation.

For code-generation sampling metrics, we use 16 responses per problem. Avg@16 denotes the average correctness over the 16 responses, while Pass@16 denotes the fraction of problems with at least one correct response among them. Decoding uses a temperature of 0.6 and top-p of 0.95.

Verification. For mathematical reasoning, correctness is determined by matching the final boxed answer against the reference after normalizing whitespace, LaTeX notation, and fraction representations. For code generation, a response is correct if it passes all evaluation tests within the benchmark's time and memory limits.

D Output Similarity Heatmaps

For each of the 30 AIME 2024 problems, we compute a 32×32 pairwise cosine similarity matrix from the embeddings of 32 sampled responses, using all-MiniLM-L6-v2 (Reimers & Gurevych, 2019). All responses are included, regardless of correctness. Red indicates higher embedding similarity, while green indicates lower similarity.

Figures 4 and 5 show the results for GRPO and ProGRPO, respectively. ProGRPO exhibits more varied within-prompt similarity patterns, consistent with reduced output concentration in this evaluation. These heatmaps describe similarities among sampled responses; lower embedding similarity alone does not establish distinct reasoning strategies.

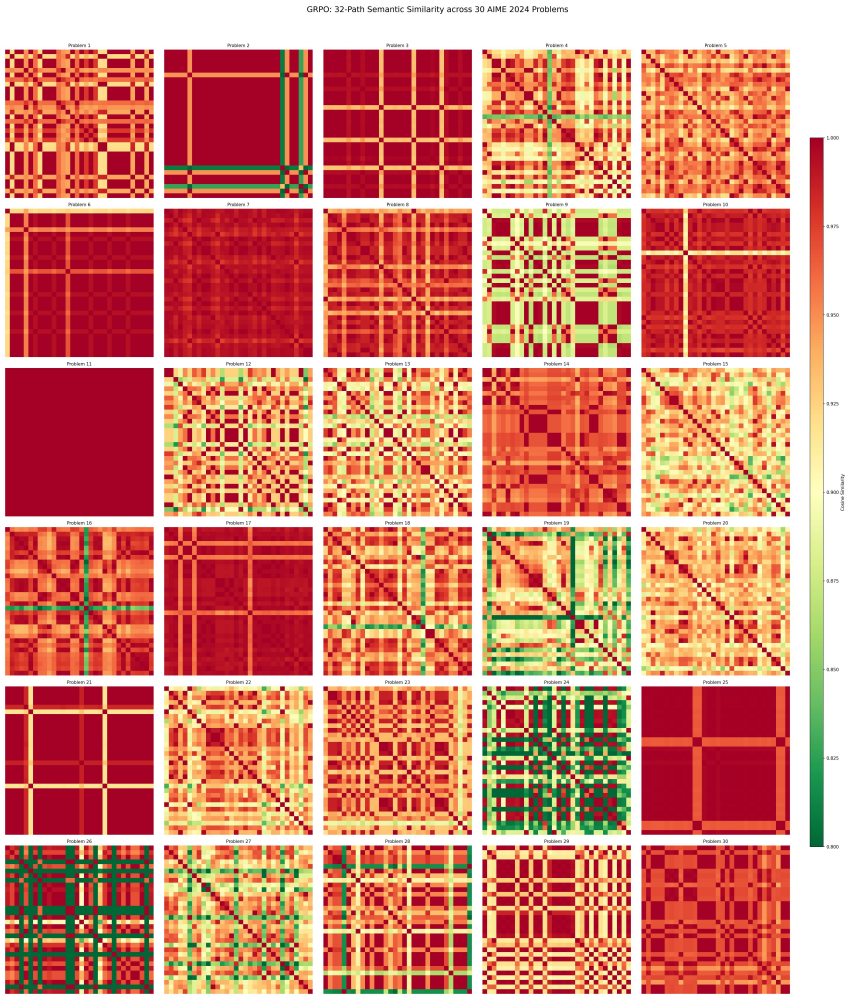


Figure 4: Pairwise response-embedding cosine similarity for GRPO on AIME 2024. Each panel corresponds to one problem and includes 32 sampled responses.

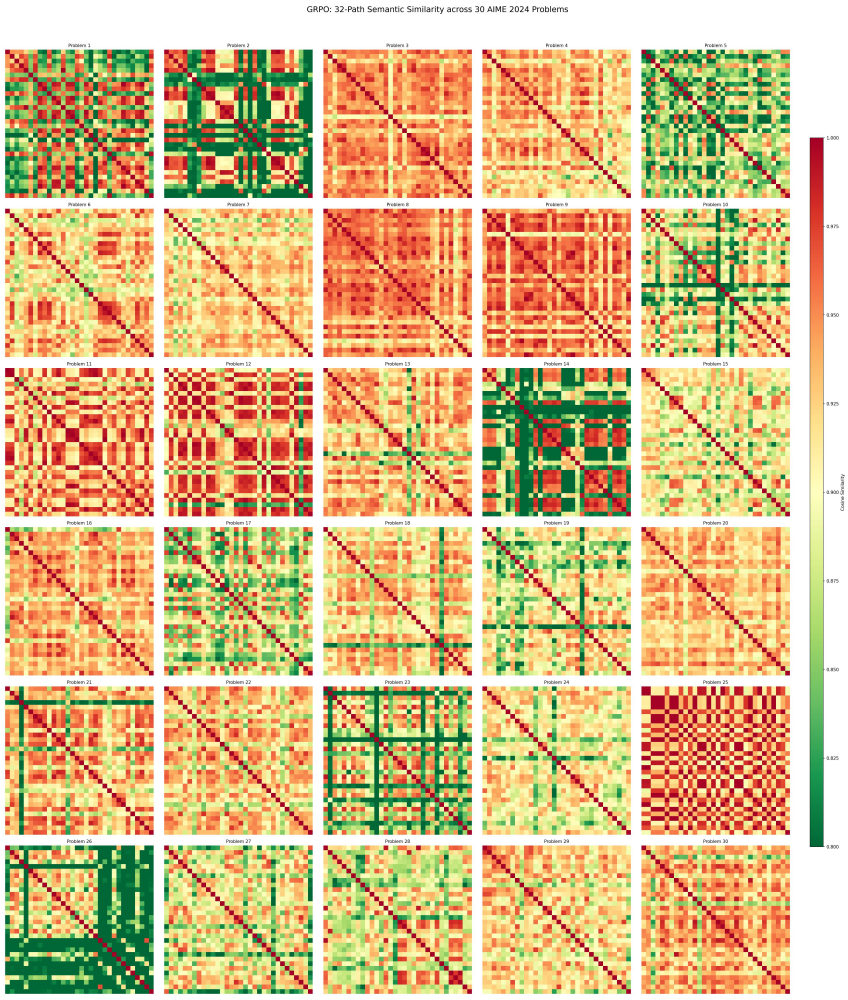


Figure 5: Pairwise response-embedding cosine similarity for ProGRPO on AIME 2024. Each panel corresponds to one problem and includes 32 sampled responses.