DVAO: DYNAMIC VARIANCE-ADAPTIVE ADVANTAGE OPTIMIZATION FOR MULTI-REWARD REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Multi-reward reinforcement learning for large language models combines task accuracy with auxiliary objectives such as length and format compliance. In Group Relative Policy Optimization, Reward Combination normalizes the scalarized reward, while Advantage Combination normalizes each reward separately before applying fixed coefficients. The latter removes differences in reward dispersion before combination, even as those statistics change during training. We propose **D**ynamic **V**ariance-adaptive **A**dvantage **O**ptimization (**DVAO**), which adapts the coefficients of normalized advantages using each reward's empirical standard deviation within a rollout group. We prove that DVAO's pointwise advantage magnitude is bounded by that of Reward Combination and characterize the cross-objective coupling in its sensitivity to reward perturbations. An equivalent groupwise rescaling representation relates this magnitude to reward correlations. Experiments on mathematical reasoning and tool use cover four *Qwen3* and *Qwen2.5* model settings. DVAO achieves the highest average accuracy among the compared multi-reward methods in each setting, with length compliance above 99.9% on both math models and the highest average format compliance on both tool-use models.

1 INTRODUCTION

Large Language Models (LLMs), including *Qwen3* (Yang et al., 2025), *Kimi K3* (Team et al., 2026), and *DeepSeek-R1* (Guo et al., 2025), support a wide range of natural language processing tasks (Plaat et al., 2025). Reinforcement Learning (RL) is widely used to align these models with human intent and task requirements (Zhang et al., 2025b; Chu et al., 2025). Group Relative Policy Optimization (GRPO) (Shao et al., 2024) and its variants (Yu et al., 2025; Zheng et al., 2025; Jiang et al., 2025) provide alternatives to Proximal Policy Optimization (PPO) (Schulman et al., 2017) for LLMs. By estimating relative advantages within a sampled rollout group, GRPO removes the need for a separate value model, reducing memory overhead and simplifying the training pipeline (Liu et al., 2025c).

However, deploying LLMs in real-world scenarios rarely involves optimizing a single, isolated metric. Practical applications require accurate answers together with length compliance (Sui et al., 2025; Feng et al., 2025), low bug rates in code generation (Tambon et al., 2025; Gao et al., 2025b), low hallucination rates (Huang et al., 2025a; Sahoo et al., 2024), and correct tool-call formatting (Qian et al., 2025; Liu et al., 2026). Adapting GRPO to this multi-reward setting requires a rule for combining these signals. Standard approaches either combine raw rewards before normalization (Reward Combination, RC) or normalize each reward before combining the resulting advantages (Advantage Combination, AC).

Normalization determines which reward statistics enter the combined advantage. RC normalizes the scalarized reward as a whole. AC instead brings each nonconstant reward component to a common scale before combining them with fixed coefficients. This removes the original dispersion differences between components: with equal base weights, the normalized advantages of low- and high-dispersion rewards receive the same coefficient. As reward distributions change during training, this motivates adapting the coefficients to the dispersion observed in each group.

We propose **D**ynamic **V**ariance-adaptive **A**dvantage **O**ptimization (**DVAO**), which reintroduces these dispersion differences into the combination of normalized advantages. DVAO multiplies

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.

each base weight by the corresponding empirical reward standard deviation and normalizes the resulting coefficients across objectives. Its analysis connects the mean squared advantages of RC and AC, establishes a pointwise magnitude bound relative to RC, and characterizes the cross-objective reward sensitivity introduced by the shared denominator. The adaptive weighting rule introduces no additional tuning parameter beyond the base combination weights.

We evaluate DVAO on mathematical reasoning and tool use with four *Qwen* model settings. Across these settings, DVAO obtains the highest average accuracy among the compared multi-reward methods, with length compliance above 99.9% on both math models and the highest average format compliance on both tool-use models. Training curves and weight sweeps examine how these trade-offs develop and vary with the base preferences. Reward-statistics analysis connects observed correlations and channel saturation to the adaptive normalization. Evaluations with *Llama-3.2-3B-Instruct*, different rollout group sizes, and three rewards examine the same combination rule across model families and reward configurations.

2 Related Work

GRPO and Advantage Estimation GRPO (Shao et al., 2024) replaces the learned value model used in PPO (Schulman et al., 2017) with group-relative advantage estimation and forms part of the training framework for reasoning models such as *DeepSeek-R1* (Guo et al., 2025). Subsequent methods modify different parts of this framework. GSPO (Zheng et al., 2025) computes importance ratios at the sequence level, while DAPO (Yu et al., 2025) combines dynamic sampling with token-level policy loss aggregation. SAPO (Gao et al., 2025a) replaces hard clipping with a temperature-controlled soft gate on token-level importance ratios. Other methods adapt advantage estimation: MAPO (Huang et al., 2025b) mixes standard-deviation-normalized and mean-relative advantages according to the estimated success rate within a rollout group, and KRPO (Wang et al., 2025) uses a Kalman filter to estimate a reward baseline and its uncertainty. These mechanisms change policy updates or the estimation of a scalar advantage. DVAO instead specifies how the statistics of multiple reward components determine the combined advantage.

Multi-Reward Reinforcement Learning in LLMs Multi-reward training combines task performance with requirements such as human preferences (Lai et al., 2024; Jang et al., 2023), response length (Shrivastava et al., 2025; Liu et al., 2025a; Luo et al., 2026), and tool-call formatting (Qian et al., 2025; Zhang et al., 2025a). Reward Combination (RC) aggregates raw rewards before group normalization, whereas Advantage Combination (AC) normalizes each reward separately before combining the resulting advantages. MO-GRPO (Ichihara et al., 2026) follows the latter ordering and analyzes how reward variance and correlation affect each objective's influence. GDPO (Liu et al., 2026) also normalizes rewards independently, then sums their advantages and applies batch-wise advantage normalization to control the resulting scale. DVAO uses the weighted sum of the component standard deviations to normalize the centered aggregate reward. This yields a group-dependent rescaling of RC whose scale reflects the component standard deviations, weights, and correlations. Our analysis characterizes this rescaling and its reward sensitivity, and our experiments compare the resulting trade-offs with RC, fixed-weight AC, and GDPO on mathematical reasoning and tool use.

3 Preliminaries

GRPO (Shao et al., 2024) estimates advantages relative to a group of sampled responses. Multi-reward applications combine correctness with requirements on response length or tool-call format (Liu et al., 2026; Qian et al., 2025). We examine how the order of reward normalization and combination determines the advantage used for policy optimization.

Formally, given a dataset $\mathcal{D}$, x is a query and y is a response. The policy π_θ assigns likelihood $\pi_\theta(y|x) = \prod_{t=1}^{|y|} \pi_\theta(y_t|x, y_{<t})$. In a multi-reward setting, n reward functions $r_1, r_2, \cdots, r_n$ evaluate different objectives. For an input-output pair (x_i, y_j), let $r_k^{(i,j)} = r_k(x_i, y_j) \in [0, 1]$. Reward combination (RC) uses a convex combination of these components:

$$r_{\text{sum}}^{(i,j)} = \sum_k w_k r_k^{(i,j)}, \qquad \sum_k w_k = 1, \quad w_k \geq 0, \tag{1}$$

where w_k specifies the base preference for objective k.

For each query x_i, GRPO samples G responses from $\pi_{\theta_{\text{old}}}$. Our analysis uses the population-form group standard deviation, $\sigma_k^i = [G^{-1}\sum_j (r_k^{(i,j)} - \mu_k^i)^2]^{1/2}$, with $\mu_k^i = G^{-1}\sum_j r_k^{(i,j)}$, and omits numerical denominator offsets. A constant reward channel is assigned zero advantage. Appendix E discusses degenerate groups and alternative normalization conventions. RC computes

$$A_{\text{sum}}^{(i,j)} = \frac{r_{\text{sum}}^{(i,j)} - \mu_{\text{sum}}^i}{\sigma_{\text{sum}}^i}, \tag{2}$$

where μ_{sum}^i and σ_{sum}^i are the corresponding group mean and standard deviation.

The GRPO surrogate objective is

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

where $s_{j,t}(\theta) = \pi_\theta(y_{j,t}|x_i, y_{j,<t})/\pi_{\theta_{\text{old}}}(y_{j,t}|x_i, y_{j,<t})$ and ϵ is the clipping range. We omit the KL term to isolate the reward-dependent surrogate. For the following gradient comparison, let $\mathcal{J}_{\text{unc}}(\theta, A)$ denote the unclipped surrogate, obtained by replacing the minimum in Equation 3 with $s_{j,t}(\theta)A^{(i,j)}$. The sampled responses and their advantages are held fixed during differentiation, giving

$$\nabla_\theta \mathcal{J}_{\text{unc}}(\theta, A) = \mathbb{E}_{x_i\sim\mathcal{D},\{y_j\}\sim\pi_{\theta_{\text{old}}}(\cdot|x_i)}\left[\frac{1}{G}\sum_{j=1}^{G}\frac{1}{|y_j|}\sum_{t=1}^{|y_j|} s_{j,t}(\theta)A^{(i,j)}\nabla_\theta \log \pi_\theta(y_{j,t}|x_i, y_{j,<t})\right]. \tag{4}$$

Advantage combination (AC) instead normalizes each reward within the group before combining the resulting advantages:

$$A_k^{(i,j)} = \frac{r_k^{(i,j)} - \mu_k^i}{\sigma_k^i}, \qquad k = 1, \ldots, n, \tag{5}$$

$$A^{(i,j)} = \sum_k w_k A_k^{(i,j)}, \qquad \sum_k w_k = 1, \quad w_k \geq 0. \tag{6}$$

The combined advantage replaces A_{sum} in Equation 3. Group reward-Decoupled Normalization Policy Optimization (GDPO) (Liu et al., 2026) additionally applies batch-wise advantage normalization.

4 METHOD

We first compare the advantage scales and reward dependence of RC and AC, then introduce DVAO and analyze its magnitude and reward sensitivity.

4.1 REWARD COMBINATION AND ADVANTAGE COMBINATION

RC and AC normalize the same reward components in different orders. The following proposition characterizes the resulting difference in mean squared advantage under the group normalization defined above.

Proposition 1. *For a fixed query x_i, suppose $\sigma_{sum}^i > 0$ and $\sigma_k^i > 0$ for every k with $w_k > 0$. Let $\hat{\rho}_{kl}^i$ be the within-group correlation of rewards k and l. Then*

$$\frac{1}{G}\sum_{j=1}^{G}\left(A_{sum}^{(i,j)}\right)^2 = 1 \geq \frac{1}{G}\sum_{j=1}^{G}\left(A^{(i,j)}\right)^2. \tag{7}$$

Equality holds if and only if $\hat{\rho}_{kl}^i = 1$ for every pair with $w_k w_l > 0$.

RC has unit mean squared advantage, while the mean squared advantage of AC depends on the correlations among its normalized components. This is a comparison of advantage scales. The norm of a policy gradient also depends on the score vectors and their cancellation. The proof is in Appendix A.

Under the fixed-rollout, unclipped analysis of Equation 4, AC satisfies

$$\nabla_\theta \mathcal{J}_{\text{unc}}(\theta, A) = \sum_k w_k \nabla_\theta \mathcal{J}_{\text{unc}}(\theta, A_k). \tag{8}$$

Thus AC combines the per-objective surrogate gradients with fixed coefficients. Each objective is normalized separately, although correlations among the rewards still affect the combined advantage and gradient. The base coefficients do not change as the within-group reward dispersions evolve.

4.2 Dynamic Variance-adaptive Advantage Optimization

DVAO reintroduces reward dispersion into AC's combination of individually normalized advantages. It scales each base weight by the corresponding group standard deviation, using the same statistic that appears in per-objective normalization.

For $S^i = \sum_l w_l \sigma_l^i > 0$, DVAO sets $\tilde{w}_k^i = w_k \sigma_k^i / S^i$ and computes

$$A_{\text{DVAO}}^{(i,j)} = \sum_k \tilde{w}_k^i A_k^{(i,j)} = \frac{\sum_k w_k \sigma_k^i A_k^{(i,j)}}{S^i}. \tag{9}$$

A channel with zero within-group variance contributes zero. If $S^i = 0$, all rewards with positive base weight are constant within the group, and we define $A_{\text{DVAO}}^{(i,j)} = 0$. The coefficients reflect observed reward dispersion, while base weights continue to specify the objective preferences.

The adaptation is specific to each query's sampled responses. For example, when all responses satisfy a length constraint, the length reward is constant and supplies no relative preference among them. DVAO then assigns the length channel zero coefficient and uses the remaining active rewards. If compliant and noncompliant responses coexist, the length channel becomes active again. The coefficients are recomputed from each new rollout group, allowing the same objective to receive different coefficients across queries within a batch. This connects the combination rule to the distinctions actually present among the sampled responses.

The next proposition compares DVAO and RC on the same rollout group.

Proposition 2. *For a fixed query x_i with $\sigma_{sum}^i > 0$, define the active objective set $\mathcal{K}_i = \{k : w_k \sigma_k^i > 0\}$. Then*

$$\left| A_{DVAO}^{(i,j)} \right| \le \left| A_{sum}^{(i,j)} \right|, \qquad j = 1, \dots, G. \tag{10}$$

Equality holds for all responses simultaneously if and only if $\hat{\rho}_{kl}^i = 1$ for all distinct $k, l \in \mathcal{K}_i$.

The proof is in Appendix B. Under these normalization assumptions, DVAO equals RC multiplied by a group scalar $\lambda^i = \sigma_{\text{sum}}^i / S^i \in (0, 1]$, preserving the within-group advantage direction while changing the relative scales across groups. Appendix E derives this identity, its correlation dependence, and finite-G bounds.

We next examine how the combined advantage changes when a single observed reward is perturbed, holding the other reward entries fixed. This sensitivity characterizes the coupling introduced by the normalization denominator.

Proposition 3. *For a fixed rollout group with $\sigma_k^i > 0$ and $S^i > 0$, the sensitivities of AC and DVAO to $r_k^{(i,j)}$ are*

$$\frac{\partial A^{(i,j)}}{\partial r_k^{(i,j)}} = \frac{w_k}{\sigma_k^i} \left(1 - \frac{1}{G} - \frac{\left(A_k^{(i,j)} \right)^2}{G} \right), \tag{11}$$

$$\frac{\partial A_{DVAO}^{(i,j)}}{\partial r_k^{(i,j)}} = \frac{w_k}{S^i} \left(1 - \frac{1}{G} - \frac{A_{DVAO}^{(i,j)} A_k^{(i,j)}}{G} \right). \tag{12}$$

For AC, the derivative depends on the normalized reward of objective k. For DVAO, the shared denominator produces a cross-term involving the combined advantage, so perturbing one reward changes its sensitivity through the other objectives' observed values. These derivatives describe reward sensitivity, whereas policy optimization treats the computed advantages as fixed coefficients. The proof is in Appendix C.

The experiments examine the accuracy–compliance trade-offs and training dynamics obtained with this adaptive normalization.

5 EXPERIMENTS

5.1 EXPERIMENTAL SETUP

Benchmarks. We evaluate DVAO on mathematical reasoning and tool use. For mathematical reasoning, we measure accuracy and length compliance on AIME-2024[1], AIME-2025[2], MATH500 (Lightman et al., 2024), OlympiadBench (He et al., 2024), and AMC23[3]. For tool use, we evaluate correctness and format compliance on the Live, Non-Live, and Multi-Turn subsets of the Berkeley Function Calling Leaderboard (BFCL-v4) (Patil et al., 2025).

Baselines and Models. We compare with GRPO (Shao et al., 2024) using the single accuracy reward r_{acc}, and with three multi-reward methods: Reward Combination (RC) and Advantage Combination (AC), implemented within GRPO, and GDPO (Liu et al., 2026). We use *Qwen3-4B-Base* and *Qwen3-8B-Base* (Yang et al., 2025) for mathematical reasoning, and *Qwen2.5-3B-Instruct* and *Qwen2.5-7B-Instruct* (Yang et al., 2024) for tool use.

Training Data and Rewards. Mathematical reasoning models are trained on DAPO-Math-17k with accuracy and binary length rewards. A response satisfies the length reward if it contains at most 4,000 tokens. Tool-use training combines 2k examples from ToolACE (Liu et al., 2025b), 1k from Hammer (Lin et al., 2024), and 1k from xLAM (Zhang et al., 2025a). Its correctness reward is paired with a binary format reward checking the required structure, fields, and field order. All reward components lie in $[0, 1]$, and the multi-reward methods use equal base weights unless otherwise specified. We report the accuracy and auxiliary-compliance scores separately to expose the balance between the objectives.

Optimization and Evaluation. We implement the methods in verl (Sheng et al., 2025), using its official implementation for GDPO. The main configuration uses AdamW (Loshchilov & Hutter, 2019) with a constant learning rate of 10^{-6}, a prompt batch size of 128, $G = 16$ responses per prompt, and 500 training steps. The maximum generation length is 8,192 tokens. Mathematical reasoning evaluation uses mean@16 with temperature 0.6 and top-p 0.95. Mean@16 averages each metric over 16 sampled responses per prompt and then over prompts. The Average column weights the five math benchmarks equally. For tool use, it weights the three BFCL subsets equally. The main tables report single training runs. Appendix D gives further implementation details.

5.2 MAIN RESULTS

Tables 1 and 2 summarize accuracy and auxiliary compliance across methods and model scales. DVAO achieves the highest average accuracy among the compared multi-reward methods in all four model settings. On *Qwen3-4B-Base*, it obtains 42.19% accuracy and 99.91% length compliance, compared with 38.99% and 96.39% for RC. On *Qwen3-8B-Base*, the corresponding DVAO scores are 47.49% and 99.92%, compared with 46.26% and 98.71% for RC. Accuracy-only GRPO attains higher average accuracy on 8B (52.57%) at lower length compliance (63.47%). GDPO reaches 99.99% length compliance on 8B but has substantially lower accuracy under this training configuration.

On tool use, DVAO has the highest average accuracy and format compliance among the compared methods at both scales. Its average accuracy/format-compliance scores are 56.66%/76.65% on *Qwen2.5-3B-Instruct* and 63.00%/79.21% on *Qwen2.5-7B-Instruct*. RC achieves higher format

[1] https://huggingface.co/datasets/Maxwell-Jia/AIME_2024

[2] https://huggingface.co/datasets/yentinglin/aime_2025

[3] https://huggingface.co/datasets/AI-MO/aimo-validation-amc

Table 1: Mathematical reasoning results under mean@16 evaluation. **Acc.**: accuracy (%). **Len.**: length compliance (%), measured as the proportion of responses with at most 4,000 tokens. Average is the unweighted mean over the five benchmarks. Bold entries mark the best result among the multi-reward methods (RC, AC, GDPO, and DVAO).

Method	AIME-2024		AIME-2025		MATH500		Olympiad		AMC23		Average	
	Acc.	Len.	Acc.	Len.	Acc.	Len.	Acc.	Len.	Acc.	Len.	Acc.	Len.
Qwen3-4B-Base												
Model	7.92	88.12	4.79	88.54	55.05	91.42	26.98	92.02	34.18	91.94	25.78	90.41
+ GRPO	17.91	62.08	10.20	68.33	78.92	93.04	42.88	82.82	49.62	82.91	39.91	77.84
+ RC	14.58	92.50	9.38	95.42	78.31	98.91	41.18	97.43	51.50	97.67	38.99	96.39
+ AC	16.25	91.04	9.38	95.21	77.65	98.69	41.02	98.08	49.47	98.12	38.75	96.23
+ GDPO	2.08	95.83	3.75	96.46	30.06	99.52	16.56	98.29	14.60	98.95	13.41	97.81
+ DVAO	**16.87**	**100.0**	**13.54**	**99.79**	**81.36**	**99.94**	**45.63**	**99.96**	**53.53**	**99.85**	**42.19**	**99.91**
Qwen3-8B-Base												
Model	11.87	89.79	8.96	93.33	69.20	97.44	34.95	94.66	41.56	96.16	33.31	94.28
+ GRPO	29.58	40.00	24.58	52.08	87.93	89.61	55.57	70.81	65.21	64.83	52.57	63.47
+ RC	21.04	97.08	16.25	99.38	84.97	99.59	49.73	98.48	59.33	99.02	46.26	98.71
+ AC	20.41	97.71	15.62	98.96	84.42	99.58	48.52	98.93	58.13	99.02	45.42	98.84
+ GDPO	1.67	**100.0**	0.00	**100.0**	35.07	**100.0**	9.15	**99.96**	27.56	**100.0**	14.69	**99.99**
+ DVAO	**21.87**	**100.0**	**18.33**	**100.0**	**86.10**	99.99	**50.62**	99.76	**60.54**	99.85	**47.49**	99.92

Table 2: Tool-use results on the Live, Non-Live, and Multi-Turn subsets of BFCL-v4. **Acc.**: accuracy (%). **Format.**: format compliance (%). Average is the unweighted mean over the three subsets. Bold entries mark the best result among the multi-reward methods (RC, AC, GDPO, and DVAO).

Method	Live		Non-Live		Multi-Turn		Average	
	Acc.	Format.	Acc.	Format.	Acc.	Format.	Acc.	Format.
Qwen2.5-3B-Instruct								
Model	59.44	11.02	47.10	4.89	1.75	1.08	36.10	5.66
+ GRPO	59.73	10.35	47.94	5.18	2.00	1.29	36.56	5.61
+ RC	67.51	61.71	79.94	83.45	5.62	36.29	51.02	60.48
+ AC	69.43	67.04	82.35	84.82	8.62	42.20	53.47	64.69
+ GDPO	67.73	65.08	82.08	84.53	8.38	48.04	52.73	65.88
+ DVAO	**72.73**	**77.44**	**84.75**	**95.11**	**12.50**	**57.40**	**56.66**	**76.65**
Qwen2.5-7B-Instruct								
Model	62.92	0.0	69.56	0.0	11.00	0.0	47.83	0.00
+ GRPO	68.76	0.0	81.65	0.0	6.38	0.0	52.26	0.00
+ RC	75.06	**87.58**	85.33	**96.11**	14.75	45.56	58.38	76.42
+ AC	63.80	67.39	56.21	85.40	12.75	51.33	44.25	68.04
+ GDPO	76.17	68.90	86.73	85.83	17.50	49.63	60.13	68.12
+ DVAO	**79.68**	77.93	**87.06**	95.11	**22.25**	**64.58**	**63.00**	**79.21**

compliance on the 7B Live and Non-Live subsets. The multi-reward methods use equal base weights here, with other weight settings examined in Section 5.4.

5.3 Training Dynamics

We track accuracy reward, length reward, and response length throughout training on *Qwen3-4B-Base* and *Qwen3-8B-Base* (Figure 1 and 2). All curves are smoothed with a centered moving average (window = 15).

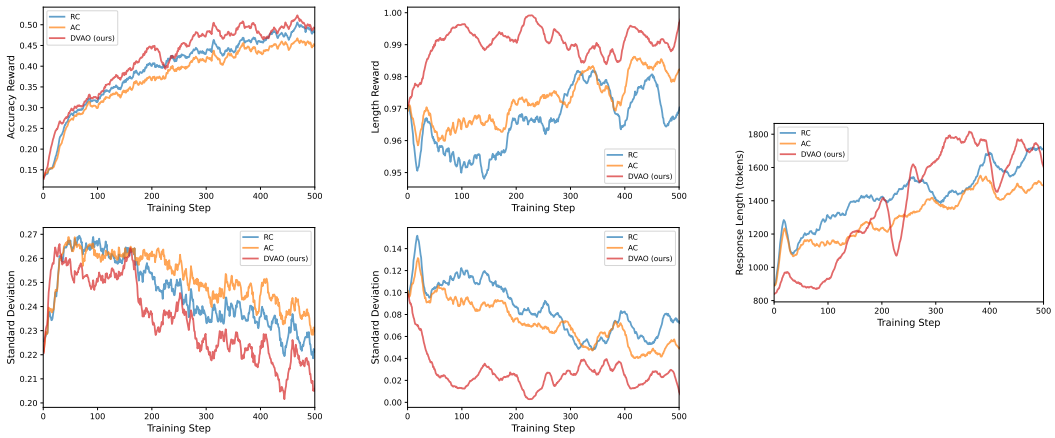


Figure 1: Training dynamics on *Qwen3-4B-Base*. Left: accuracy reward, with the mean above and standard deviation below. Middle: the same statistics for length reward. Right: average response length.

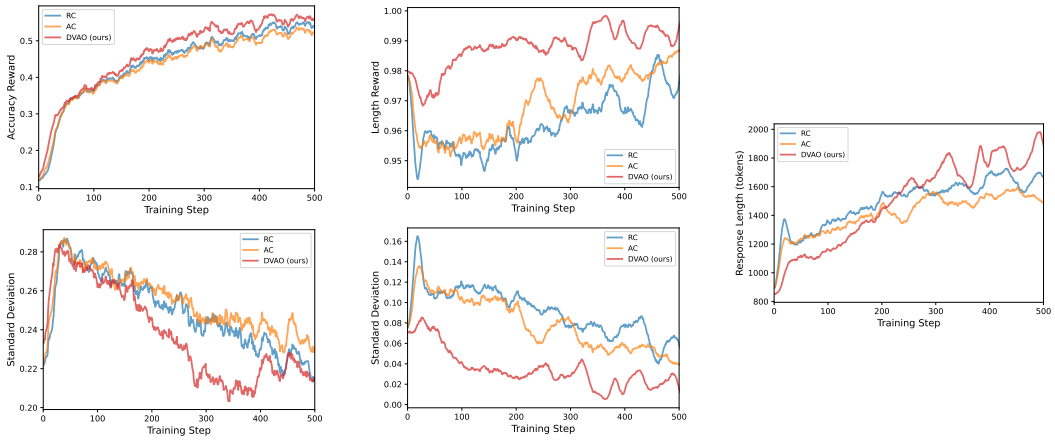
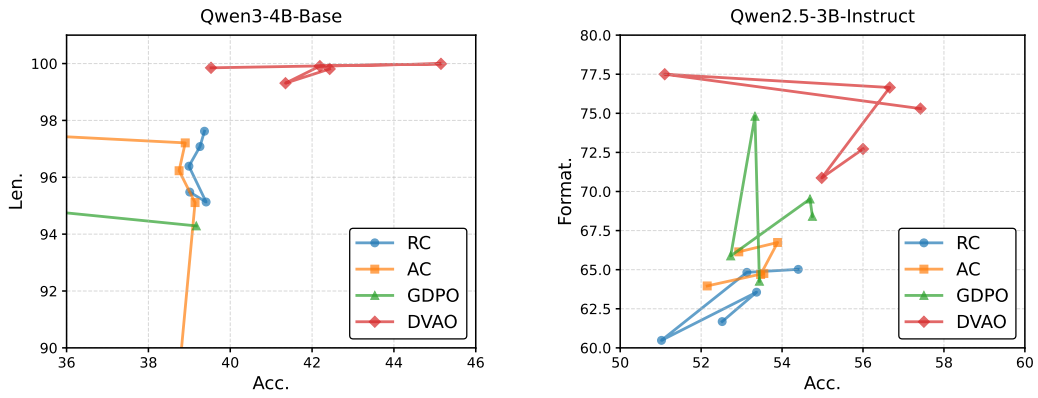


Figure 2: Training dynamics on *Qwen3-8B-Base*. Left: accuracy reward, with the mean above and standard deviation below. Middle: the same statistics for length reward. Right: average response length.

Accuracy reward. On both model scales, DVAO's mean accuracy reward increases during training and reaches a higher final level than RC and AC. Its within-group reward standard deviation decreases overall from its early peak, with fluctuations later in training.

Length reward. DVAO's length reward approaches 1.0 while its standard deviation decreases. RC and AC retain more variation in this reward at the end of the displayed runs. As the length signal saturates, its coefficient decreases in groups where the length-reward standard deviation falls relative to that of accuracy. Section 5.5 connects the aggregate training-log statistics with group-level correlation and rescaling measurements on validation rollouts.

Response length. Response lengths increase from their initial levels for all three methods. In the second half of training, DVAO generally produces longer responses than AC, with visible oscillations. Its response lengths overlap with RC on 4B and are generally higher on 8B. Because the length reward checks an upper threshold, responses can grow longer while remaining compliant.



(a) Mathematical Reasoning Task (*Qwen3-4B-Base*) (b) Tool-Use Task (*Qwen2.5-3B-Instruct*)

Figure 3: Accuracy vs. length/format compliance across methods. Points correspond to the five tested base-weight settings, and connecting lines trace the weight sweep.

5.4 Accuracy–Compliance Trade-offs

Section 5.2 reports one checkpoint per method at equal base weights. To examine how each method balances accuracy and auxiliary compliance, we sweep the accuracy weight w_1 (with length or format weight $w_2 = 1 - w_1$) over $\{0.1, 0.3, 0.5, 0.7, 0.9\}$. Figure 3a shows the resulting trade-offs for mathematical reasoning with *Qwen3-4B-Base*, and Figure 3b shows tool use with *Qwen2.5-3B-Instruct*.

On mathematical reasoning, DVAO maintains length compliance above 99% throughout the weight sweep, while its accuracy varies with the base weight. On tool use, two operating points near 75% format compliance reach approximately 57% accuracy for DVAO and 53% for GDPO. The methods trace different accuracy–compliance trade-offs under the same grid of base weights.

5.5 Reward Statistics and Adaptive Weighting

We examine how reward variation translates into group rescaling using validation rollouts from a *Qwen3-8B-Base* DVAO run with equal base weights. The data comprise 1,318 prompts, 16 responses per prompt, and 26 checkpoints spanning steps 0–500. Among the 34,268 prompt–checkpoint groups, 4,485 have nonzero variance in both rewards. These jointly active groups have mean reward correlation 0.267, with negative correlation in 6.82% of them. Their mean rescaling factor λ is 0.801 and its fifth percentile is 0.683 (Table 4). Thus, positive average reward correlation coexists with attenuation of RC advantages. This observation connects the pointwise comparison in Proposition 2 to the reward distributions encountered by the trained policies.

The share of jointly active groups falls from 40.52% at step 0 to 9.56% at step 500 as length compliance increases. A group with one active reward has $\lambda = 1$, whereas constant-reward groups carry no centered reward signal. The changing prevalence of these groups distinguishes saturation of an auxiliary reward from changes in rescaling among groups where both objectives still vary. As more groups contain uniformly compliant responses, fewer groups require combining distinct accuracy and length signals.

The training logs provide a complementary aggregate view. With bars denoting batch averages of group standard deviations, the accuracy-weight proxy $\widehat{w}_{\mathrm{acc}} = \bar{\sigma}_{\mathrm{acc}}/(\bar{\sigma}_{\mathrm{acc}} + \bar{\sigma}_{\mathrm{len}})$ rises from 0.692 over the first 25 steps to 0.842 over the last 25, while $\bar{\sigma}_{\mathrm{len}}$ falls from 0.1117 to 0.0432. This ratio of averages differs from the average individual group weight. The aggregate shift is consistent with reduced emphasis on the length channel as its reward variation declines. Appendix F gives the diagnostic protocol and degenerate-group counts.

Table 3: DVAO sensitivity to training group size on *Qwen3-4B-Base*. All columns use mean@16 evaluation over the five math benchmarks.

Training group size G	**4**	**8**	**16**	**32**
Accuracy (%)	39.79	40.99	42.19	**43.49**
Length compliance (%)	97.80	97.80	**99.91**	99.10

5.6 Sensitivity to Group Size, Model Family, and Reward Design

Rollout group size. Group size controls the number of responses available for estimating reward dispersion. Table 3 varies training G from 4 to 32 on *Qwen3-4B-Base*, keeping 500 training steps and 128 prompts per batch. Evaluation uses 16 responses per prompt for every configuration. Accuracy increases across the four settings, while length compliance remains at least 97.80%. Even at $G = 4$, accuracy exceeds the base model's 25.78% by 14.01 percentage points. The results show that DVAO remains effective with smaller rollout groups in this setting. Larger groups increase the rollout budget alongside the samples available for estimating reward variation. Appendix G provides the full configuration.

Model family. We evaluate *Llama-3.2-3B-Instruct* using the training configuration of Table 1. Evaluation uses mean@16 on MATH500, OlympiadBench, and AMC23, excluding the AIME benchmarks. DVAO reaches 38.06% average accuracy and 94.19% length compliance, compared with 27.87% and 87.96% for RC (Table 5). These averages weight the three benchmarks equally. DVAO also achieves the highest accuracy and length compliance on each benchmark among the compared methods, extending the evaluation to an additional model family.

Reward design. We add correctness-gated brevity as a third reward for *Qwen3-4B-Base*, using the training and evaluation settings of Table 1. DVAO achieves the highest scores among the compared three-reward methods on all three metrics, reaching 41.70% accuracy, 97.27% length compliance, and a brevity score of 34.15 (Table 6). Its accuracy and length compliance exceed RC by 4.97 and 2.51 percentage points, respectively. A separate short-run comparison removes the correctness gate, allowing empty outputs to earn maximal auxiliary rewards. Both RC and DVAO collapse within 20 steps to one-token outputs and 0.0% accuracy. These comparisons show gains under three objectives and continued sensitivity to auxiliary-reward design.

6 Conclusion

We introduce Dynamic Variance-adaptive Advantage Optimization for multi-reward GRPO. DVAO uses within-group reward standard deviations to adapt the coefficients of normalized advantages. The resulting advantage preserves RC's within-group direction while changing its scale across groups, with pointwise magnitudes no greater than RC and a shared denominator that couples reward sensitivity across objectives. On mathematical reasoning and tool use, DVAO achieves the highest average accuracy among the compared multi-reward methods in all four *Qwen* model settings while maintaining high auxiliary compliance. These results support using within-group dispersion to combine task accuracy with structural auxiliary rewards.

AI use statement

We use generative AI tools solely for language polishing to improve the grammar, clarity, and readability of this paper. These tools were not used for experimental design, implementation, execution, or analysis. We take full responsibility for the final content of this work.

Ethics statement

Use of Human Annotations Human annotations are only used in methodological research at the beginning of the work, to assist in analyzing the feasibility of the proposed solution. Annotators consented to the use of data for research purposes. We ensure that the privacy of all annotators is

protected throughout the annotation process, and all of them are adequately paid according to local standards. Human annotations are not applied during the evaluation of our method.

Risks In this paper, all datasets are obtained from official sources. The datasets adopted have been anonymized and do not contain offensive information. However, we cannot guarantee that the datasets do not contain socially harmful or toxic language.

Reproducibility statement

The reward definitions, datasets, model configurations, and training and evaluation settings are given in Appendix D. Proofs of the three propositions are provided in Appendices A–C, with the extended normalization analysis in Appendix E. The reward-statistics analysis in Appendix F specifies the validation protocol and the computation of group correlations and rescaling factors. Our code is available at **Supplementary Material**.

References

Tianzhe Chu, Yuexiang Zhai, Jihan Yang, Shengbang Tong, Saining Xie, Dale Schuurmans, Quoc V. Le, Sergey Levine, and Yi Ma. SFT memorizes, RL generalizes: A comparative study of foundation model post-training. In *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=dYur3yabMj.

Sicheng Feng, Gongfan Fang, Xinyin Ma, and Xinchao Wang. Efficient reasoning models: A survey. *Trans. Mach. Learn. Res.*, 2025, 2025. URL https://openreview.net/forum?id=sySqlxj8EB.

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

Ruofan Gao, Amjed Tahir, Peng Liang, Teo Susnjak, and Foutse Khomh. A survey of bugs in ai-generated code. *arXiv preprint arXiv:2512.05239*, 2025b.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, Hao Zhang, Hanwei Xu, Honghui Ding, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jingchang Chen, Jingyang Yuan, Jinhao Tu, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaichao You, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingxu Zhou, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Tao Yun, Tian Pei, Tianyu Sun, Tao Wang, Wangding Zeng, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang,

Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nat.*, 645(8081):633–638, 2025. doi: 10.1038/S41586-025-09422-Z. URL https://doi.org/10.1038/s41586-025-09422-z.

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 Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pp. 3828–3850. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.ACL-LONG.211. URL https://doi.org/10.18653/v1/2024.acl-long.211.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Trans. Inf. Syst.*, 43(2):42:1–42:55, 2025a. doi: 10.1145/3703155. URL https://doi.org/10.1145/3703155.

Wenke Huang, Quan Zhang, Yiyang Fang, Jian Liang, Xuankun Rong, Huanjin Yao, Guancheng Wan, Ke Liang, Wenwen He, Mingjun Li, et al. Mapo: Mixed advantage policy optimization. *arXiv preprint arXiv:2509.18849*, 2025b.

Yuki Ichihara, Yuu Jinnai, Tetsuro Morimura, Mitsuki Sakamoto, Ryota Mitsuhashi, and Eiji Uchibe. MO-GRPO: Mitigating reward hacking of group relative policy optimization on multi-objective problems. *Transactions of the Association for Computational Linguistics*, 14:1867–1882, 2026. doi: 10.1162/tacl.a.782. URL https://aclanthology.org/2026.tacl-1.84/.

Joel Jang, Seungone Kim, Bill Yuchen Lin, Yizhong Wang, Jack Hessel, Luke Zettlemoyer, Hannaneh Hajishirzi, Yejin Choi, and Prithviraj Ammanabrolu. Personalized soups: Personalized large language model alignment via post-hoc parameter merging. *arXiv preprint arXiv:2310.11564*, 2023.

Guochao Jiang, Wenfeng Feng, Guofeng Quan, Chuzhan Hao, Yuewei Zhang, Guohua Liu, and Hao Wang. Vcrl: Variance-based curriculum reinforcement learning for large language models. *arXiv preprint arXiv:2509.19803*, 2025.

Yuhang Lai, Siyuan Wang, Shujun Liu, Xuanjing Huang, and Zhongyu Wei. Alarm: Align language models via hierarchical rewards modeling. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, Findings of ACL, pp. 7817–7831. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.FINDINGS-ACL.465. URL https://doi.org/10.18653/v1/2024.findings-acl.465.

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 *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=v8L0pN6EOi.

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

Shih-Yang Liu, Xin Dong, Ximing Lu, Shizhe Diao, Mingjie Liu, Min-Hung Chen, Hongxu Yin, Yu-Chiang Frank Wang, Kwang-Ting Cheng, Yejin Choi, et al. Dler: Doing length penalty right-incentivizing more intelligence per token via reinforcement learning. *arXiv preprint arXiv:2510.15110*, 2025a.

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

Weiwen Liu, Xu Huang, Xingshan Zeng, Xinlong Hao, Shuai Yu, Dexun Li, Shuai Wang, Weinan Gan, Zhengying Liu, Yuanqing Yu, Zezhong Wang, Yuxian Wang, Wu Ning, Yutai Hou, Bin Wang, Chuhan Wu, Xinzhi Wang, Yong Liu, Yasheng Wang, Duyu Tang, Dandan Tu, Lifeng Shang, Xin Jiang, Ruiming Tang, Defu Lian, Qun Liu, and Enhong Chen. Toolace: Winning the points of LLM function calling. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025b. URL https://openreview.net/forum?id=8EB8k6DdCU.

Zihe Liu, Jiashun Liu, Yancheng He, Weixun Wang, Jiaheng Liu, Ling Pan, Xinyu Hu, Shaopan Xiong, Ju Huang, Jian Hu, et al. Part i: Tricks or traps? a deep dive into rl for llm reasoning. *arXiv preprint arXiv:2508.08221*, 2025c.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Haotian Luo, Haiying He, Yibo Wang, Shiwei Liu, Wei Li, Xiaochun Cao, Dacheng Tao, Naiqiang Tan, and Li Shen. O1-pruner: Length-harmonizing fine-tuning for o1-like reasoning pruning. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Findings of the Association for Computational Linguistics, ACL 2026, San Diego, California, United States, July 2-7, 2026*, pp. 14242–14257. Association for Computational Linguistics, 2026. doi: 10.18653/V1/2026.FINDINGS-ACL.697. URL https://doi.org/10.18653/v1/2026.findings-acl.697.

Shishir G. Patil, Huanzhi Mao, Fanjia Yan, Charlie Cheng-Jie Ji, Vishnu Suresh, Ion Stoica, and Joseph E. Gonzalez. The berkeley function calling leaderboard (BFCL): from tool use to agentic evaluation of large language models. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, volume 267 of *Proceedings of Machine Learning Research*. PMLR / OpenReview.net, 2025. URL https://proceedings.mlr.press/v267/patil25a.html.

Aske Plaat, Max J. van Duijn, Niki van Stein, Mike Preuss, Peter van der Putten, and Kees Joost Batenburg. Agentic large language models, a survey. *J. Artif. Intell. Res.*, 84, 2025. doi: 10.1613/JAIR.1.18675. URL https://doi.org/10.1613/jair.1.18675.

Cheng Qian, Emre Can Acikgoz, Qi He, Hongru Wang, Xiusi Chen, Dilek Hakkani-Tur, Gokhan Tur, and Heng Ji. Toolrl: Reward is all tool learning needs. In Danielle Belgrave, Cheng Zhang, Laura N. Montoya, Hsuan-Tien Lin, Razvan Pascanu, Piotr Koniusz, Marzyeh Ghassemi, Nancy Chen, Iván Vladimir Meza Ruíz, and Arturo Loaiza-Bonilla (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025. URL http://papers.nips.cc/paper_files/paper/2025/hash/97c5b2707228e7e3fb67e4ecc2e0e607-Abstract-Conference.html.

Pranab Sahoo, Prabhash Meharia, Akash Ghosh, Sriparna Saha, Vinija Jain, and Aman Chadha. A comprehensive survey of hallucination in large language, image, video and audio foundation models. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, Florida, USA, November 12-16, 2024*, volume EMNLP 2024 of *Findings of ACL*, pp. 11709–11724. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.FINDINGS-EMNLP.685. URL https://doi.org/10.18653/v1/2024.findings-emnlp.685.

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

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

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient RLHF framework. In *Proceedings*

of the Twentieth European Conference on Computer Systems, EuroSys 2025, Rotterdam, The Netherlands, 30 March 2025 - 3 April 2025, pp. 1279–1297. ACM, 2025. doi: 10.1145/3689031.3696075. URL https://doi.org/10.1145/3689031.3696075.

Vaishnavi Shrivastava, Ahmed Awadallah, Vidhisha Balachandran, Shivam Garg, Harkirat Behl, and Dimitris Papailiopoulos. Sample more to think less: Group filtered policy optimization for concise reasoning. *arXiv preprint arXiv:2508.09726*, 2025.

Yang Sui, Yu-Neng Chuang, Guanchu Wang, Jiamu Zhang, Tianyi Zhang, Jiayi Yuan, Hongyi Liu, Andrew Wen, Shaochen Zhong, Na Zou, Hanjie Chen, and Xia Hu. Stop overthinking: A survey on efficient reasoning for large language models. *Trans. Mach. Learn. Res.*, 2025, 2025. URL https://openreview.net/forum?id=HvoG8SxggZ.

Florian Tambon, Arghavan Moradi Dakhel, Amin Nikanjam, Foutse Khomh, Michel C. Desmarais, and Giuliano Antoniol. Bugs in large language models generated code: an empirical study. *Empir. Softw. Eng.*, 30(3):65, 2025. doi: 10.1007/S10664-025-10614-4. URL https://doi.org/10.1007/s10664-025-10614-4.

Kimi Team, Tongtong Bai, Yifan Bai, Yiping Bao, Jianfeng Cai, Xinyuan Cai, Peizhou Cao, Yuxuan Cao, Ziwei Chai, Y Charles, et al. Kimi k3: Open frontier intelligence. *arXiv preprint arXiv:2607.24653*, 2026.

Hu Wang, Congbo Ma, Ian Reid, and Mohammad Yaqub. Kalman filter enhanced grpo for reinforcement learning-based language model reasoning. *arXiv preprint arXiv:2505.07527*, 2025.

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

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

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Juncai Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Ru Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Yonghui Wu, and Mingxuan Wang. DAPO: an open-source LLM reinforcement learning system at scale. In Danielle Belgrave, Cheng Zhang, Laura N. Montoya, Hsuan-Tien Lin, Razvan Pascanu, Piotr Koniusz, Marzyeh Ghassemi, Nancy Chen, Iván Vladimir Meza Ruíz, and Arturo Loaiza-Bonilla (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025. URL http://papers.nips.cc/paper_files/paper/2025/hash/a4277440d50f1f15d2cb4c14f7e0c0d2-Abstract-Conference.html.

Jianguo Zhang, Tian Lan, Ming Zhu, Zuxin Liu, Thai Hoang, Shirley Kokane, Weiran Yao, Juntao Tan, Akshara Prabhakar, Haolin Chen, Zhiwei Liu, Yihao Feng, Tulika Manoj Awalgaonkar, Rithesh R. N., Zeyuan Chen, Ran Xu, Juan Carlos Niebles, Shelby Heinecke, Huan Wang, Silvio Savarese, and Caiming Xiong. xlam: A family of large action models to empower AI agent systems. In Luis Chiruzzo, Alan Ritter, and Lu Wang (eds.), *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2025 - Volume 1: Long Papers, Albuquerque, New Mexico, USA, April 29 - May 4, 2025*, pp. 11583–11597. Association for Computational Linguistics, 2025a. doi: 10.18653/V1/2025.NAACL-LONG.578. URL https://doi.org/10.18653/v1/2025.naacl-long.578.

Kaiyan Zhang, Yuxin Zuo, Bingxiang He, Youbang Sun, Runze Liu, Che Jiang, Yuchen Fan, Kai Tian, Guoli Jia, Pengfei Li, et al. A survey of reinforcement learning for large reasoning models. *arXiv preprint arXiv:2509.08827*, 2025b.

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

A Proof of Proposition 1

Proposition 1. *For a fixed query x_i, suppose $\sigma^i_{sum} > 0$ and $\sigma^i_k > 0$ for every k with $w_k > 0$. Let $\hat{\rho}^i_{kl}$ be the within-group correlation of rewards k and l. Then*

$$\frac{1}{G}\sum_{j=1}^{G}\left(A^{(i,j)}_{sum}\right)^2 = 1 \geq \frac{1}{G}\sum_{j=1}^{G}\left(A^{(i,j)}\right)^2.$$

Equality holds if and only if $\hat{\rho}^i_{kl} = 1$ for every pair with $w_k w_l > 0$.

Proof. Fix the rollout group and use the population-form standard deviation defined in the main text. Every reward with positive base weight has zero-mean, unit-mean-square normalized advantages:

$$\frac{1}{G}\sum_j A^{(i,j)}_k = 0, \qquad \frac{1}{G}\sum_j \left(A^{(i,j)}_k\right)^2 = 1.$$

The normalized scalarized reward also has unit mean square because $\sigma^i_{\text{sum}} > 0$. Writing $\mathcal{W} = \{k : w_k > 0\}$, the AC mean square is

$$\begin{aligned}\frac{1}{G}\sum_j \left(\sum_{k\in\mathcal{W}} w_k A^{(i,j)}_k\right)^2 &= \sum_{k\in\mathcal{W}} w_k^2 + 2\sum_{\substack{k<l\\ k,l\in\mathcal{W}}} w_k w_l \hat{\rho}^i_{kl} \\ &= 1 - 2\sum_{\substack{k<l\\ k,l\in\mathcal{W}}} w_k w_l (1-\hat{\rho}^i_{kl}) \leq 1.\end{aligned}$$

All subtracted terms are nonnegative, so equality holds exactly when each positively weighted pair has correlation one. With a single positively weighted objective, the pairwise condition is vacuous and equality holds. □

B Proof of Proposition 2

Proposition 2. *For a fixed query x_i with $\sigma^i_{sum} > 0$, define $\mathcal{K}_i = \{k : w_k \sigma^i_k > 0\}$. Then*

$$\left|A^{(i,j)}_{DVAO}\right| \leq \left|A^{(i,j)}_{sum}\right|, \qquad j = 1, \ldots, G.$$

Equality holds for all responses simultaneously if and only if $\hat{\rho}^i_{kl} = 1$ for all distinct $k, l \in \mathcal{K}_i$.

Proof. Linearity of the group mean gives

$$\sigma^i_{\text{sum}} A^{(i,j)}_{\text{sum}} = \sum_k w_k (r^{(i,j)}_k - \mu^i_k) = \sum_k w_k \sigma^i_k A^{(i,j)}_k.$$

A zero-variance channel has both $r^{(i,j)}_k - \mu^i_k = 0$ and $A^{(i,j)}_k = 0$, so the identity includes such channels. Let $S^i = \sum_k w_k \sigma^i_k$. Expanding the scalarized variance over the active objectives yields

$$\begin{aligned}(\sigma^i_{\text{sum}})^2 &= \sum_{k\in\mathcal{K}_i} w_k^2 (\sigma^i_k)^2 + 2\sum_{\substack{k<l\\ k,l\in\mathcal{K}_i}} w_k w_l \sigma^i_k \sigma^i_l \hat{\rho}^i_{kl} \\ &\leq \left(\sum_k w_k \sigma^i_k\right)^2 = (S^i)^2.\end{aligned}$$

Hence $0 < \sigma^i_{\text{sum}}/S^i \leq 1$, and the preceding identity gives the pointwise inequality after division by S^i. Since $\sigma^i_{\text{sum}} > 0$, at least one RC advantage is nonzero. Equality for all responses therefore requires $\sigma^i_{\text{sum}} = S^i$, which is equivalent to correlation one for every pair in $\mathcal{K}_i$. Equality at an individual response with zero RC advantage does not require this pairwise condition. □

C Proof of Proposition 3

Proposition 3. *For a fixed rollout group with $\sigma_k^i > 0$ and $S^i > 0$, the sensitivities of AC and DVAO to $r_k^{(i,j)}$ are*

$$\frac{\partial A^{(i,j)}}{\partial r_k^{(i,j)}} = \frac{w_k}{\sigma_k^i}\left(1 - \frac{1}{G} - \frac{(A_k^{(i,j)})^2}{G}\right),$$
$$\frac{\partial A_{DVAO}^{(i,j)}}{\partial r_k^{(i,j)}} = \frac{w_k}{S^i}\left(1 - \frac{1}{G} - \frac{A_{DVAO}^{(i,j)} A_k^{(i,j)}}{G}\right).$$

Proof. Hold all reward entries other than $r_k^{(i,j)}$ fixed. Under population-form group normalization,

$$\frac{\partial \mu_k^i}{\partial r_k^{(i,j)}} = \frac{1}{G}, \qquad \frac{\partial \sigma_k^i}{\partial r_k^{(i,j)}} = \frac{r_k^{(i,j)} - \mu_k^i}{G\sigma_k^i} = \frac{A_k^{(i,j)}}{G}.$$

Applying the quotient rule to $A_k^{(i,j)} = (r_k^{(i,j)} - \mu_k^i)/\sigma_k^i$ gives

$$\frac{\partial A_k^{(i,j)}}{\partial r_k^{(i,j)}} = \frac{1}{\sigma_k^i}\left(1 - \frac{1}{G} - \frac{(A_k^{(i,j)})^2}{G}\right).$$

The other normalized reward channels are unchanged under this perturbation, so multiplication by w_k gives the AC expression. For DVAO, write

$$A_{\text{DVAO}}^{(i,j)} = \frac{N^{(i,j)}}{S^i}, \qquad N^{(i,j)} = \sum_l w_l (r_l^{(i,j)} - \mu_l^i).$$

Since $\partial N^{(i,j)}/\partial r_k^{(i,j)} = w_k(1 - 1/G)$ and $\partial S^i/\partial r_k^{(i,j)} = w_k A_k^{(i,j)}/G$, another application of the quotient rule gives

$$\frac{\partial A_{\text{DVAO}}^{(i,j)}}{\partial r_k^{(i,j)}} = \frac{w_k(1-1/G)S^i - N^{(i,j)} w_k A_k^{(i,j)}/G}{(S^i)^2}$$
$$= \frac{w_k}{S^i}\left(1 - \frac{1}{G} - \frac{A_{\text{DVAO}}^{(i,j)} A_k^{(i,j)}}{G}\right).$$

This establishes the sensitivity of the normalization map to an observed reward. □

D Implementation Details

For mathematical reasoning, we train on DAPO-Math-17k[4], which contains 17k prompts with integer answers. For tool use, we follow ToolRL and use 2k examples from ToolACE (Liu et al., 2025b), 1k from Hammer (Lin et al., 2024), and 1k from xLAM (Zhang et al., 2025a). Each example contains a question and ground-truth tool calls.

Mathematical reasoning uses accuracy reward r_{acc} and length reward r_{length}. The length reward is the indicator that the generated response has at most $l = 4{,}000$ tokens. Tool use combines correctness reward r_{acc} with binary format reward r_{format}, which checks the required structure, fields, and their order. All reward components lie in $[0, 1]$. Base weights sum to one and are equal unless otherwise specified.

We implement the methods using verl (Sheng et al., 2025), including its official GDPO implementation. DVAO computes group standard deviations with denominator G and assigns zero advantage to constant reward channels. Groups with $S^i = 0$ receive zero combined advantage. We train with AdamW (Loshchilov & Hutter, 2019) at a constant learning rate of 10^{-6}. The prompt batch size is 128, with $G = 16$ sampled responses per prompt in the main configuration. Main experiments run

[4] https://huggingface.co/datasets/BytedTsinghua-SIA/DAPO-Math-17k

for 500 training steps, with a maximum generation length of 8,192 tokens. Mathematical reasoning results use mean@16 with evaluation temperature 0.6 and top-p 0.95. The five-benchmark Average column is the unweighted mean of the benchmark scores. Experiments use a server with 8×NVIDIA H20-3e GPUs and an Intel Xeon Platinum 8575C CPU. Settings for the group-size, model-family, and reward-design studies are given in Appendices G–I.

E Analysis of Advantage Normalization

This section supplements Propositions 1–3 with the relation between DVAO and RC, the role of within-group reward correlations, and the normalization conditions used by the analysis. Fix a query x_i and define

$$N^{(i,j)} = \sum_k w_k (r_k^{(i,j)} - \mu_k^i), \qquad S^i = \sum_k w_k \sigma_k^i, \qquad \mathcal{K}_i = \{k : w_k \sigma_k^i > 0\}. \tag{13}$$

E.1 Equivalence to Groupwise Rescaling

For $\sigma_{\text{sum}}^i > 0$, the normalization definitions give

$$A_{\text{DVAO}}^{(i,j)} = \frac{N^{(i,j)}}{S^i} = \underbrace{\frac{\sigma_{\text{sum}}^i}{S^i}}_{\lambda^i} A_{\text{sum}}^{(i,j)}, \qquad 0 < \lambda^i \le 1. \tag{14}$$

The upper bound follows from $\sigma_{\text{sum}}^i \le S^i$, proved in Appendix B. DVAO preserves the relative advantages and their signs within a group, while changing the scale assigned to that group relative to other queries. Its mean squared advantage is

$$\frac{1}{G} \sum_j \left(A_{\text{DVAO}}^{(i,j)} \right)^2 = (\lambda^i)^2, \tag{15}$$

whereas RC has unit mean squared advantage on a nonconstant scalarized reward.

For a binary reward with positive base weight and empirical success fraction p_k^i, the population-form standard deviation is $\sigma_k^i = \sqrt{p_k^i (1 - p_k^i)}$. It is largest when half the responses succeed and vanishes when all responses receive the same reward. Holding the other reward statistics fixed, the channel's DVAO coefficient decreases as p_k^i rises from $1/2$ toward 1, provided another positively weighted channel remains active. This explains how saturation can reduce the weight assigned to a length-compliance reward. Dispersion also vanishes when every response fails, so the reward mean distinguishes saturation at success from uniform failure.

For the fixed-rollout, unclipped surrogate analyzed in the main text, let $g_{\text{RC}}^i(\theta)$ be the policy-surrogate gradient contributed by group i. With the group advantages held fixed during differentiation, Equation 14 gives $g_{\text{DVAO}}^i(\theta) = \lambda^i g_{\text{RC}}^i(\theta)$. Different group coefficients can therefore change the direction of their aggregate gradient. Optimizer state, gradient clipping, and any additional KL term also affect the eventual parameter update.

E.2 Dependence on Reward Correlations

Expanding the variance of the scalarized reward gives

$$\begin{aligned} (\lambda^i)^2 &= \frac{\sum_{k \in \mathcal{K}_i} w_k^2 (\sigma_k^i)^2 + 2 \sum_{\substack{k<l \\ k,l \in \mathcal{K}_i}} w_k w_l \sigma_k^i \sigma_l^i \hat{\rho}_{kl}^i}{(S^i)^2} \\ &= 1 - \frac{2 \sum_{\substack{k<l \\ k,l \in \mathcal{K}_i}} w_k w_l \sigma_k^i \sigma_l^i (1 - \hat{\rho}_{kl}^i)}{(S^i)^2}. \end{aligned} \tag{16}$$

For fixed base weights and marginal standard deviations, lower pairwise correlations decrease $(\lambda^i)^2$. Perfect positive correlation among the active reward channels gives $\lambda^i = 1$, recovering RC. When the active rewards vary in less aligned ways, DVAO attenuates the group's advantage relative to RC. With two active objectives and $w_1 \sigma_1^i = w_2 \sigma_2^i$, this reduces to $(\lambda^i)^2 = (1 + \hat{\rho}_{12}^i)/2$.

Here correlation measures agreement among reward values across sampled responses. It is distinct from the inner product between objective-specific policy gradients or a Pareto conflict between policies.

E.3 Finite-Group Magnitude Bounds

With the population-form group standard deviation, any nonconstant normalized reward vector $a \in \mathbb{R}^G$ satisfies $\sum_j a_j = 0$ and $\sum_j a_j^2 = G$. For each coordinate, Cauchy–Schwarz gives

$$a_j^2 = \left(\sum_{h \neq j} a_h\right)^2 \leq (G-1)\sum_{h \neq j} a_h^2 = (G-1)(G - a_j^2), \tag{17}$$

so $|a_j| \leq \sqrt{G-1}$. Consequently,

$$|A^{(i,j)}_{\text{DVAO}}| \leq |A^{(i,j)}_{\text{sum}}| \leq \sqrt{G-1}, \qquad |A^{(i,j)}| \leq \sqrt{G-1}, \tag{18}$$

where the AC bound follows from its convex combination of normalized advantages. RC therefore has a finite-G magnitude bound, and Proposition 2 further compares DVAO and RC pointwise on the same group.

E.4 Degenerate Groups and Normalization Conventions

If $\sigma_k^i = 0$, the channel has $r_k^{(i,j)} - \mu_k^i = 0$ for every response. Its normalized advantage is defined as zero, and its DVAO contribution is zero. If $S^i = 0$, every positively weighted channel is constant, so DVAO and RC are both defined as zero. If $S^i > 0$ but $\sigma^i_{\text{sum}} = 0$, the centered reward components cancel exactly: $N^{(i,j)} = 0$ for all j. DVAO is then zero. With the zero-advantage convention for RC, Equation 14 extends by setting $\lambda^i = 0$. Correlations are used only for pairs in $\mathcal{K}_i$. Proposition 3 concerns perturbations at $\sigma_k^i > 0$, since the unsmoothed standard deviation is not differentiable at a constant reward vector.

The theoretical identities use the population-form denominator G and omit numerical denominator offsets. If a common sample-standard-deviation convention with denominator $G-1$ is used throughout, the ratio $\sigma^i_{\text{sum}}/S^i$ and the rescaling identity are unchanged. Each nonconstant normalized reward component, including RC, then has mean square $(G-1)/G$. DVAO has mean square $(\lambda^i)^2(G-1)/G$, while AC has mean square at most $(G-1)/G$. The single-coordinate magnitude bound becomes $(G-1)/\sqrt{G}$. The derivative of the standard deviation becomes $A_k^{(i,j)}/(G-1)$, so the last terms in the sensitivity formulas use $1/(G-1)$ in place of $1/G$.

F Reward Correlations and Group Rescaling

We examine training-time validation rollouts from a DVAO run with *Qwen3-8B-Base* and equal base weights for accuracy and length compliance. The validation prompt template and online verifier differ from the evaluation pipeline used for Table 1. The data contain 1,318 prompts from the five mathematical reasoning benchmarks, with 16 responses per prompt at each of 26 checkpoints spanning steps 0 to 500. Responses were sampled at temperature 0.6. Accuracy rewards were taken from the validation outputs, and length rewards were recomputed as $\mathbf{1}[\text{len}(y) \leq 4000]$ using a *Qwen3* tokenizer. Group standard deviations use the sample convention, with denominator $G-1$.

The resulting 34,268 prompt–checkpoint groups include 4,485 groups in which both rewards have nonzero variance. Table 4 summarizes these *jointly active* groups, whose mean reward correlation is positive and among which 6.82% have negative correlation. As length compliance becomes more frequent, the share of jointly active groups falls from 40.52% of all groups at step 0 to 9.56% at step 500.

The mean rescaling factor of 0.801 shows attenuation relative to RC on jointly active groups. Groups with a single active reward have $\lambda = 1$. The data also contain 10 jointly active groups with $\lambda = 0$, where the weighted centered rewards cancel, and 14,305 groups with zero variance in both rewards. These constant-reward groups have no centered reward signal, so their undefined standard-deviation ratios are excluded from statistics of λ.

Table 4: Reward correlation and rescaling statistics pooled over the 4,485 jointly active validation groups. Percentiles describe this conditional distribution. Repeated checkpoints contain the same evaluation prompts.

Statistic	**Value**
Correlation $\hat{\rho}$: mean / median	0.267 / 0.218
Correlation $\hat{\rho}$: 5th / 95th percentile	-0.124 / 0.856
Fraction with $\hat{\rho} < 0$	6.82%
Rescaling λ: mean / median	0.801 / 0.793
Rescaling λ: 5th / 10th percentile	0.683 / 0.730

The scalar training logs summarize how reward variability evolves within training batches. At each step, let $\bar{\sigma}_{\mathrm{acc},t}$ and $\bar{\sigma}_{\mathrm{len},t}$ denote the logged batch averages of the group standard deviations. The ratio

$$\widehat{w}_{\mathrm{acc},t} = \frac{\bar{\sigma}_{\mathrm{acc},t}}{\bar{\sigma}_{\mathrm{acc},t} + \bar{\sigma}_{\mathrm{len},t}} \tag{19}$$

is an aggregate proxy for the accuracy weight, rather than the mean of the individual group weights. Its average rises from 0.692 over steps 1–25 to 0.842 over steps 476–500. Over the same intervals, the logged length-reward standard deviation decreases from 0.1117 to 0.0432 as the length signal saturates.

G Sensitivity to Rollout Group Size

Table 3 reports DVAO with *Qwen3-4B-Base*, accuracy and length rewards, and training group sizes $G \in \{4, 8, 16, 32\}$. All runs use the main experimental configuration, changing only G: DAPO-Math-17k training data, 500 training steps, a prompt batch size of 128, and a maximum generation length of 8,192 tokens. Evaluation uses the macro average over the five math benchmarks with 16 samples per prompt, temperature 0.6, and top-p 0.95, independently of training group size. The $G = 16$ entry is the main configuration from Table 1. The base model scores 25.78% accuracy and 90.41% length compliance under this evaluation protocol.

Accuracy increases across the four reported group sizes, while length compliance remains at least 97.80%. At $G = 4$, accuracy remains 14.01 percentage points above the base model. With prompt batch size and training steps fixed, the number of sampled responses increases linearly with G. Thus, group size changes both the statistical estimation and the rollout budget in this comparison.

H Evaluation Across Model Families

To evaluate DVAO on another model family, we repeat the mathematical reasoning comparison with *Llama-3.2-3B-Instruct*. Training follows the configuration of Table 1: DAPO-Math-17k, 500 training steps, a prompt batch size of 128, $G = 16$, and a maximum generation length of 8,192 tokens. The multi-reward methods use equally weighted accuracy and length rewards with a 4,000-token threshold, while GRPO uses the accuracy reward alone. We evaluate on MATH500, OlympiadBench, and AMC23. The AIME benchmarks are omitted because of their difficulty for this model. Evaluation uses mean@16, temperature 0.6, and top-p 0.95. The Average column gives the unweighted mean over these three benchmarks.

Table 5 shows that DVAO achieves the highest accuracy and length compliance on each benchmark among the compared methods. Its macro accuracy and length compliance are 38.06% and 94.19%, exceeding RC by 10.19 and 6.23 percentage points, respectively. Accuracy-only GRPO reaches 32.85% average accuracy with 54.83% length compliance.

I Three-Objective Optimization and Reward Design

We study three objectives using *Qwen3-4B-Base* under the training and evaluation settings of Table 1. The model is trained on DAPO-Math-17k for 500 steps, with a prompt batch size of 128, $G = 16$,

Table 5: Mathematical reasoning results on *Llama-3.2-3B-Instruct* under mean@16 evaluation. **Acc.**: accuracy (%). **Len.**: length compliance (%). Average is the unweighted mean over MATH500, OlympiadBench, and AMC23. The AIME benchmarks are excluded. Bold entries mark the best result among the multi-reward methods.

Method	MATH500		Olympiad		AMC23		Average	
	Acc.	Len.	Acc.	Len.	Acc.	Len.	Acc.	Len.
Llama-3.2-3B-Instruct								
Model	40.20	93.71	8.80	79.45	22.52	68.37	23.84	80.51
+ GRPO	49.50	59.01	21.40	58.35	27.64	47.12	32.85	54.83
+ RC	44.58	96.03	11.03	90.74	28.01	77.11	27.87	87.96
+ AC	46.83	93.89	10.80	87.02	30.05	72.29	29.23	84.40
+ GDPO	42.26	94.95	13.41	69.39	24.62	70.52	26.76	78.29
+ DVAO	**54.81**	**97.34**	**24.13**	**94.54**	**35.24**	**90.68**	**38.06**	**94.19**

Table 6: Three-objective results on *Qwen3-4B-Base* under the same training and evaluation settings as Table 1. Scores are macro-averaged over the five math benchmarks using mean@16. Accuracy and length compliance are percentages. Brevity is the mean reward from Equation 20 multiplied by 100. Bold entries mark the best result among the compared multi-reward methods.

Method	Accuracy	Length compliance	Brevity
Base	25.78	90.41	13.12
+ RC	36.73	94.76	26.74
+ AC	34.55	92.70	25.31
+ GDPO	18.52	87.45	11.64
+ DVAO	**41.70**	**97.27**	**34.15**

and a maximum generation length of 8,192 tokens. Evaluation reports the macro average over the five mathematical reasoning benchmarks using mean@16, temperature 0.6, and top-p 0.95. We use equal base weights for accuracy, length compliance, and brevity. The correctness-gated brevity reward is

$$r_{\text{brevity}}(y) = r_{\text{acc}}(y) \max\left(0, 1 - \frac{\text{len}(y)}{4000}\right). \tag{20}$$

This reward favors shorter correct responses. Table 6 compares RC, AC, GDPO, and DVAO under the three-objective configuration. DVAO reaches the highest score on all three metrics: 41.70% accuracy, 97.27% length compliance, and a brevity score of 34.15. Relative to RC, the gains are 4.97 percentage points in accuracy, 2.51 percentage points in length compliance, and 7.41 points in brevity. RC and AC also improve all three scores over the base model, whereas GDPO scores below the base model on each metric in this configuration.

In a separate short-run reward-design comparison, we remove the correctness gate and compare RC with DVAO on *Qwen3-4B-Base* using DAPO-Math-17k and equal base weights. Short outputs then earn brevity reward regardless of their answer, and an empty output can receive maximal brevity and length rewards. Both methods collapse within 20 training steps, reaching one-token outputs and 0.0% accuracy. This failure illustrates the incentive to exploit a brevity reward that is independent of correctness, which persists under both combination rules.