How to Allocate, How to Learn? Dynamic Rollout Allocation and Advantage Modulation for Policy Optimization

Yangyi Fang[1,2,*], **Jiaye Lin**[1,*], **Xiaoliang Fu**[1,3,*], **Cong Qin**[1,4], **Haolin Shi**[2], **Chaowen Hu**[1], **Lu Pan**[1], **Ke Zeng**[1], **Xunliang Cai**[1,†]
[1]Meituan [2]Tsinghua University [3]Fudan University [4]Peking University
{fangyangyi, linjiaye}@meituan.com

Abstract

Reinforcement Learning with Verifiable Rewards (RLVR) has proven effective for Large Language Model (LLM) reasoning, yet current methods face key challenges in resource allocation and policy optimization dynamics: (i) uniform rollout allocation ignores gradient variance heterogeneity across problems, and (ii) the softmax policy structure causes gradient attenuation for high-confidence correct actions, while excessive gradient updates may destabilize training. Therefore, we propose **DynaMO**, a theoretically-grounded dual-pronged optimization framework. *At the sequence level*, we prove that uniform allocation is suboptimal and derive variance-minimizing allocation from the first principle, establishing Bernoulli variance as a computable proxy for gradient informativeness. *At the token level*, we develop gradient-aware advantage modulation grounded in theoretical analysis of gradient magnitude bounds. Our framework compensates for gradient attenuation of high-confidence correct actions while utilizing entropy changes as computable indicators to stabilize excessive update magnitudes. Extensive experiments conducted on a diverse range of mathematical reasoning benchmarks demonstrate consistent improvements over strong RLVR baselines. Our implementation is available at: https://github.com/GithubX-F/DynaMO-RL.

1 Introduction

Reinforcement Learning with Verifiable Rewards (RLVR) has emerged as a powerful paradigm for advancing Large Language Model (LLM) reasoning (Ouyang et al., 2022; Bai et al., 2022). Recent breakthrough models, such as OpenAI o1 (Jaech et al., 2024) and DeepSeek-R1 (Guo et al., 2025), demonstrate emergent capabilities like long-form chain-of-thought and self-reflection. Building on these successes, numerous works have explored RLVR methods (Shao et al., 2024; Fang et al., 2026), most commonly combining the GRPO algorithm (Shao et al., 2024) or its variants (Liu et al., 2025; Yu et al., 2025a) with outcome-based rewards for reinforcement learning. However, despite these advances, fundamental challenges persist in both computational resource allocation and policy optimization dynamics.

Current RLVR methods uniformly distribute rollout budgets across training instances (Shao et al., 2024), ignoring heterogeneous gradient informativeness among various problems. This overlooks a fundamental trade-off: while high-variance problems contribute more informative learning signals, they simultaneously introduce greater estimation noise that may destabilizes model performance. Existing adaptive strategies either prioritize sample selection over resource allocation (Bengio et al., 2009; Schaul et al., 2015; Tong et al., 2024) or target different optimization frameworks (Dong et al., 2023; Yao et al., 2025), leaving the variance-informativeness trade-off in policy gradient methods unaddressed. Moreover, this limitation is particularly pronounced on datasets with diverse difficulties, where effective allocation requires dynamically balancing informativeness and noise.

Compounding this resource allocation challenge, the policy gradient dynamics in RLVR training expose fundamental token-level optimization issues. Theoretical analysis shows that the mathematical structure of the softmax policy induces an inherent tension (Li, 2025): high-confidence correct actions naturally yield smaller gradient magnitudes, leading to insufficient learning signals, whereas excessive gradient updates may destabilize training (Luo et al., 2025). Our analysis further reveals that gradient magnitude is upper-bounded by policy entropy, allowing entropy changes to serve as computable indicators of such instability. Existing approaches attempt mitigation via ratio clipping (Yu et al., 2025a; Yang et al., 2025), sample reweighting (Zhu et al.,

[*] Equal contribution. [†] Corresponding author.

Findings of the Association for Computational Linguistics: ACL 2026, pages 14727–14744

2025), or entropy-induced advantages (Cheng et al., 2025; Tan and Pan, 2025; Wang et al., 2025), but these coarse-grained interventions lack a unified theoretical grounding and even amplify rather than stabilize training fluctuations.

To address these key challenges, we propose **DynaMO**, a dual-pronged optimization framework grounded in variance minimization theory and policy gradient analysis. *At the sequence level*, we derive dynamic rollout allocation that explicitly balances the informativeness-noise trade-off by gradient variance minimization specifically for policy gradient methods, establishing Bernoulli variance as a lightweight computable proxy. *At the token level*, we introduce gradient-aware advantage modulation through integrated compensation and stabilization mechanisms based on our gradient-entropy analysis: compensation for gradient attenuation in high-confidence correct actions and stabilization against excessive update magnitudes by monitoring entropy changes as an indicator. Our core contributions in this paper are summarized as follows:

- We prove that uniform allocation is suboptimal and subsequently derive variance-minimizing rollout allocation with a lightweight proxy.
- We establish the gradient-entropy relationship through theoretical analysis, enabling gradient-aware advantage modulation with compensation and stabilization mechanisms.
- Extensive experiments across six benchmarks and three LLM scales demonstrate consistent improvements, with comprehensive ablations validating each component and visualizations revealing stable optimization dynamics.

2 Related Works

2.1 Reinforcement Learning for LLMs

Reinforcement learning has emerged as a dominant paradigm for LLM post-training, with RLHF and RLVR demonstrating significant success (Ouyang et al., 2022; Bai et al., 2022; Schulman et al., 2017). Recent breakthrough models, including DeepSeek-R1 (Guo et al., 2025), DeepSeekMath (Shao et al., 2024), OpenAI o1 (Jaech et al., 2024), and Kimi k1.5 (Team et al., 2025), further demonstrate the effectiveness of RLVR on reasoning tasks with verifiable rewards. While subsequent works have introduced algorithmic refinements (Liu et al., 2025; Yu et al., 2025a; Chu et al., 2025; Hu et al., 2025; Fang et al., 2025), fundamental challenges in computational efficiency and optimization stability still persist.

2.2 Entropy Dynamics in Policy Optimization

Entropy regularization balances exploration and exploitation (Haarnoja et al., 2018; Mnih et al., 2016), yet its role in LLM training remains contentious (Ouyang et al., 2022; Shao et al., 2024; Yu et al., 2025a; Chu et al., 2025). Entropy collapse (Luo et al., 2025) motivates mitigation strategies such as ratio clipping (Yu et al., 2025a; Yang et al., 2025), sample reweighting (Zhu et al., 2025), or entropy-induced advantages (Cheng et al., 2025; Tan and Pan, 2025; Wang et al., 2025; Li et al., 2026a). Li (2025) further shows that high-confidence actions yield attenuated gradient magnitudes, inducing asymmetric learning dynamics. Yet existing methods lack unified theoretical grounding, and the interplay between gradient attenuation and entropy dynamics remains underexplored. Our work addresses this through gradient-aware policy update control grounded in the gradient-entropy relationship, where entropy serves as a computable indicator of update magnitude rather than a direct optimization target.

2.3 Sample Efficiency

Sample efficiency is critical for RLVR training, where generating multiple rollouts per problem incurs substantial cost. Standard methods employ uniform rollout budgets (Shao et al., 2024), overlooking heterogeneous gradient informativeness. Curriculum learning (Bengio et al., 2009) and prioritized experience replay (Schaul et al., 2015) choose which problems to train on, but emphasize sample ordering rather than resource allocation. Offline methods (Tong et al., 2024) repeatedly sample until obtaining a fixed number of correct responses, lacking dynamic scheduling for online training. EM-based methods (Dong et al., 2023; Gulcehre et al., 2023; Yao et al., 2025; Liu et al., 2021a,b, 2022; Yu et al., 2025b) enhance efficiency via iterative rejection sampling, yet require gradient norm computations and target the EM framework rather than policy gradient methods. In contrast, we derive optimal rollout allocation by minimizing gradient variance for policy gradient methods, with a lightweight, gradient-free proxy based on historical success statistics.

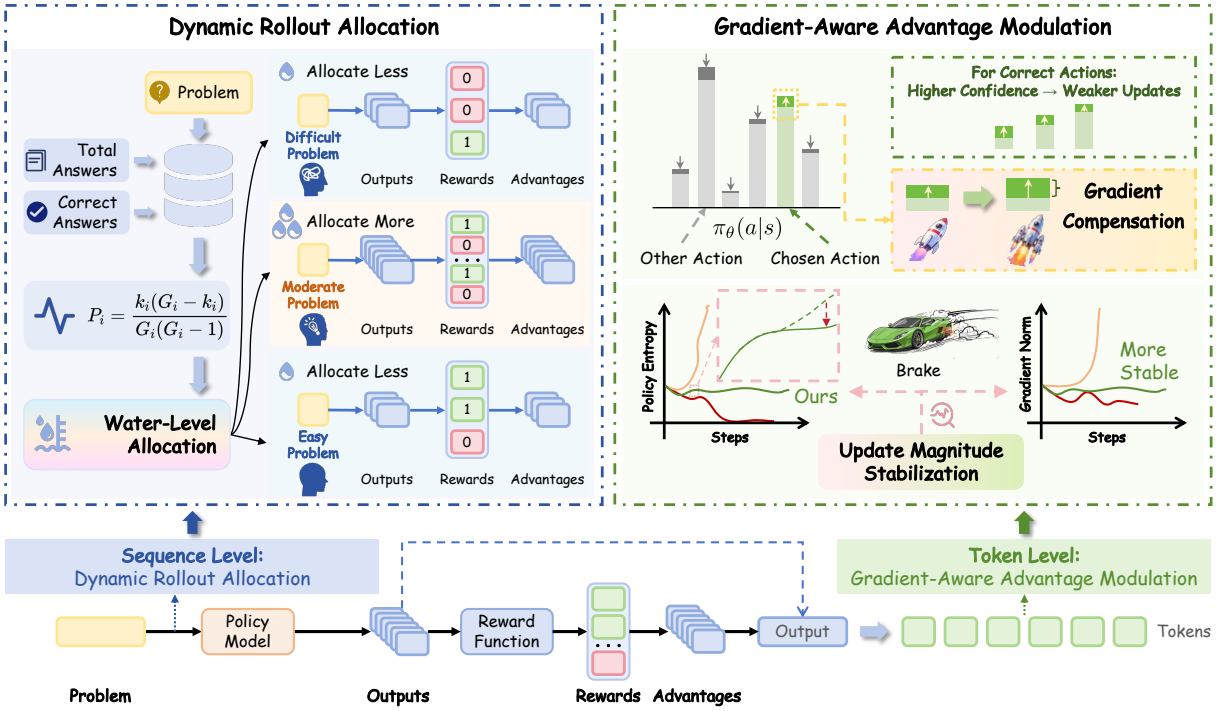


Figure 1: Overview of DynaMO, which operates at both the sequence and token levels, enabling fine-grained control over optimization. (i) Left: Dynamic allocation concentrates the rollout budget on high-variance problems. (ii) Right: Gradient-aware advantage modulation compensates for attenuated gradients and stabilizes excessive updates.

3 Preliminaries

Policy Optimization in RLVR. Formally, given a prompt q sampled from the training data $\mathcal{D}$, let π_θ denotes the policy model parameterized by θ. In the context of RLVR, the model autoregressively generates a response $o = \{o_1, o_2, \ldots, o_T\}$, where each token o_t represents an action taken at step t and T represents the sequence length. The final objective is to maximize the expected reward:

$$J(\theta) = \mathbb{E}_{q\sim\mathcal{D}, o\sim\pi_\theta(\cdot|q)}[R(o)], \tag{1}$$

where $R(\cdot)$ is the reward function that evaluates the quality of the generated response o.

To reduce variance, GRPO (Shao et al., 2024) samples G responses $\{o_i\}_{i=1}^G$ per prompt, estimating advantages via group-wise normalization:

$$A_{i,t} = \frac{R(o_i) - \text{mean}(\{R(o_j)\}_{j=1}^G)}{\text{std}(\{R(o_j)\}_{j=1}^G)}, \tag{2}$$

where $R(o_i)$ denotes the reward for response o_i, evaluated by the reward function $R(\cdot)$. mean$(\cdot)$ and std$(\cdot)$ denote the mean and standard deviation of the rewards within the group.

With the advantage $A_{i,t}$ shared across all tokens, GRPO maximizes the clipped surrogate objective:

$$\mathcal{L}(\theta) = \mathbb{E}_{q\sim\mathcal{D}, \{o_i\}_{i=1}^G\sim\pi_{\text{old}}(\cdot|q)}\left[\frac{1}{\sum_{i=1}^G |o_i|}\sum_{i=1}^G\sum_{t=1}^{|o_i|} \min\left(r_{i,t}A_{i,t}, \text{clip}(r_{i,t}, 1-\epsilon, 1+\epsilon)A_{i,t}\right)\right], \tag{3}$$

where $r_{i,t} = \frac{\pi_\theta(o_{i,t}|q,o_{i,<t})}{\pi_{\text{old}}(o_{i,t}|q,o_{i,<t})}$ is the importance sampling ratio and ϵ is the clipping parameter. Consistent with prior works (Yu et al., 2025a; Liu et al., 2025), we omit the KL divergence penalty term and do not elaborate further on this component.

Policy Entropy. Policy entropy quantifies the uncertainty in the model's action selection process, serving as a key indicator of exploration in reinforcement learning. Given a policy model π_θ and training data $\mathcal{D}$, the policy entropy is defined as:

$$\mathcal{H}(\pi_\theta, \mathcal{D}) = \mathbb{E}_{q\sim\mathcal{D}, \{o_i\}_{i=1}^G\sim\pi_\theta(\cdot|q)}\left[\frac{1}{\sum_{i=1}^G |o_i|}\sum_{i=1}^G\sum_{t=1}^{|o_i|} \left(-\sum_{v\in\mathcal{V}} \pi_\theta(v|q, o_{i,<t}) \log \pi_\theta(v|q, o_{i,<t})\right)\right], \tag{4}$$

where $\mathcal{V}$ represents the vocabulary. This entropy metric reflects the model's confidence distribution: higher values indicate greater uncertainty and exploration potential, whereas lower values suggest more deterministic behavior selection.

4 Methodology

4.1 Overview

To effectively address the challenges of inefficient resource allocation and policy optimization dynamics highlighted in Section 1, we propose **DynaMO** (**Dyna**mic Rollout Allocation and Advantage **M**odulation for Policy **O**ptimization). As illustrated in Figure 1, this dual-pronged framework operates at both the sequence and token levels, enabling fine-grained control over optimization.

Notably, DynaMO comprises two key components: (i) dynamic rollout allocation that adaptively distributes computational budget based on gradient variance minimization, concentrating resources on problems with balanced success-failure distributions where learning signals are most informative, and (ii) gradient-aware advantage modulation based on the gradient-entropy upper bound relationship, which compensates for gradient attenuation in high-confidence actions while using entropy changes to stabilize excessive updates.

4.2 Dynamic Rollout Allocation

We establish a theoretical framework for optimal rollout allocation by formulating it as a gradient variance minimization problem, which considers a training dataset $\mathcal{D}$ with N prompts $\{q_i\}_{i=1}^N$.

4.2.1 Optimization Theory

For a prompt q_i with n_i rollout budget, the corresponding gradient estimator $\hat{g}_i = \frac{1}{n_i}\sum_{k=1}^{n_i} g_{i,k}$ has variance $\mathrm{Var}[\hat{g}_i] = \sigma_i^2/n_i$. Minimizing the total estimation variance $\sum_i \mathrm{Var}[\hat{g}_i]$ under budget constraint $\sum_i n_i = B$ yields (proof in Appendix C):

$$n_i^* = B \cdot \frac{\sigma_i}{\sum_{k=1}^N \sigma_k}. \tag{5}$$

This principle allocates more rollouts to problems with higher gradient variance. To implement this, we use Bernoulli variance P_i as a practical proxy for σ_i (derivation in Appendix C).

4.2.2 Bernoulli Variance as Practical Proxy

For binary rewards, the variance follows $p(1-p)$, where p represents the success probability. Then, we estimate this via historical statistics with k_i correct responses out of G_i total rollouts generated:

$$P_i = \frac{k_i(G_i - k_i)}{G_i(G_i-1)}, \quad \mathbb{E}[P_i] = p_i(1-p_i). \tag{6}$$

This unbiased estimator can be efficiently computed from historical success counts, with problems that maximize P_i (characterized by balanced correct/incorrect responses) exhibiting large σ_i.

Algorithm 1 Variance-Driven Dynamic Allocation

Require: Historical statistics $\{(G_i, k_i)\}_{i=1}^N$, total rollout budget B, allocation bounds $[G_{\min}, G_{\max}]$
Compute priorities $P_i = k_i(G_i - k_i)/[G_i(G_i-1)]$
Initialize $G_i^{\text{new}} = G_{\min}$, $B_{\text{rem}} = B - N \cdot G_{\min}$
while $B_{\text{rem}} > 0$ and $\exists i : G_i^{\text{new}} < G_{\max}$ **do**
  $\mathcal{E} = \{i : G_i^{\text{new}} < G_{\max}\}$
  for $i \in \mathcal{E}$ **do**
    $\Delta G_i = \min\left(\left\lfloor \frac{B_{\text{rem}} P_i}{\sum_{j\in\mathcal{E}} P_j} \right\rfloor, G_{\max} - G_i^{\text{new}}\right)$
    $G_i^{\text{new}} \leftarrow G_i^{\text{new}} + \Delta G_i$
    $B_{\text{rem}} \leftarrow B_{\text{rem}} - \Delta G_i$
  end for
end while
return $\{G_1^{\text{new}}, \ldots, G_N^{\text{new}}\}$

Water-Level Implementation. Algorithm 1 implements the proportional budget allocation $n_i \propto P_i$ with the boundary constraints $G_{\min}$ (ensuring minimum coverage) and $G_{\max}$ (preventing over-concentration). After each training iteration, we incrementally update the statistics via $G_i \leftarrow G_i + G_i^{\text{new}}$ and $k_i \leftarrow k_i + k_i^{\text{new}}$, thereby enabling an adaptive allocation as model proficiency evolves. Notably, the dynamic allocation strategy provably reduces variance against the conventional uniform allocation, with theoretical justification and operational details in Appendix C and D.

4.3 Gradient-Aware Advantage Modulation

Beyond resource allocation, the softmax policy structure causes high-confidence correct actions to produce attenuated gradient magnitudes, while excessive gradient updates may undermine training stability. Grounded in theoretical analysis of gradient dynamics, we design an integrated compensation and stabilization mechanism.

4.3.1 Gradient Compensation

Our gradient analysis, detailed in Appendix B, reveals the fundamental relationship between the update magnitude and token-level entropy. Specifically, considering a softmax policy parameterized by logits $z(s)$ at state $s = (q, o_{<t})$, the advantage-weighted updates with learning rate η satisfy:

$$\begin{aligned}\mathbb{E}_{a_k\sim\pi_\theta(\cdot|s)}\left[\|\Delta z(s)\|_2^2\right] &= \eta^2\,\mathbb{E}[A^2]\Big(1 - \sum_{k=1}^{|\mathcal{V}|}\pi_k^2\Big) \\ &\le \eta^2\,\mathbb{E}[A^2]\Big(1 - \exp\big(-\mathcal{H}(\pi_\theta|s)\big)\Big),\end{aligned} \tag{7}$$

where $\mathcal{H}(\pi_\theta|s) = -\sum_{v\in\mathcal{V}}\pi_\theta(v|s)\log\pi_\theta(v|s)$ is the token-level entropy and $\sum_{k=1}^{|\mathcal{V}|}\pi_k^2$ measures policy concentration. Accordingly, high-confidence

tokens ($\pi_k \approx 1$) produce minimal expected update magnitudes ($\sum_{j=1}^{|\mathcal{V}|} \pi_j^2 \approx 1$), resulting in gradient attenuation for confident correct actions—an observation that constitutes the theoretical basis for gradient compensation in our method.

To effectively mitigate this gradient attenuation phenomenon for the confident correct actions, we introduce a gradient compensation factor:

$$\beta_{i,t}^{\text{comp}} = \mathbb{I}[A_{i,t} > 0] \cdot g(\mathcal{H}_{i,t}) + \mathbb{I}[A_{i,t} \leq 0], \tag{8}$$

where the compensation function $g(\cdot)$ is designed to scale inversely with the entropy:

$$g(\mathcal{H}) = 1 + \alpha \cdot \frac{\mathcal{H}_{\max} - \mathcal{H}}{\mathcal{H}_{\max} - \mathcal{H}_{\min}}. \tag{9}$$

Here, α determines the maximum compensation factor, while $\mathcal{H}_{\min}$ and $\mathcal{H}_{\max}$ represent the minimum and maximum token-level entropy within the current training batch. This asymmetric design targets the confidence–update-magnitude asymmetry: for positive-advantage tokens, the compensation function scales inversely with entropy to counteract gradient attenuation, moderately amplifying learning signals for high-confidence correct actions; for negative-advantage tokens, compensation is bypassed to preserve the natural penalty signals. Furthermore, for uncertain actions with high entropy, the function returns to unity, maintaining natural update dynamics.

4.3.2 Update Magnitude Stabilization

While gradient compensation addresses the attenuation for high-confidence actions, excessive gradient updates still may destabilize the training dynamics. By leveraging the gradient-entropy relationship established above, we utilize entropy changes as an indicator to detect such optimization instability. Specifically, our policy update decomposition reveals that changes in logits induce corresponding entropy changes through a factorized form:

$$\Delta\mathcal{H}(\pi_\theta^k|s) \approx -\eta \sum_a \pi_\theta^k(a|s)^2 \cdot \Lambda_\theta^k(a|s) \cdot \xi_{i,t}(a), \tag{10}$$

where $\Lambda_\theta^k(a|s)$ denotes the centered log-probability (Definition 2) that captures the deviation of action a's log-probability from the policy's average entropy. $\xi_{i,t}(a) = \mathbb{I}_{\text{clip}} \cdot r_{i,t} \cdot A_{i,t}$ represents the composite update coefficient combining clipping, importance sampling, and advantage estimation. The complete derivation is provided in Appendix A.

This decomposition shows that when both the policy concentration $\pi_\theta^k(a|s)^2$ and the composite coefficient magnitude $|\xi_{i,t}(a)|$ are large, entropy changes become substantial. Building upon the established gradient-entropy relationship, such large entropy changes indicate excessive gradient magnitudes that may destabilize training. Accordingly, we define the token-level instability indicator:

$$\Xi_{i,t} = \left|\Delta\mathcal{H}(\pi_\theta^k|s_{i,t})\right|, \tag{11}$$

which quantitatively measures the estimated contribution of each token to overall update instability.

To stabilize tokens exhibiting excessive entropy changes while simultaneously maintaining learning efficiency, we formulate a stabilization factor:

$$\beta_{i,t}^{\text{stab}} = f\left(\frac{\Xi_{i,t}}{\max_j \Xi_{j,t}}\right), \tag{12}$$

where $f(\cdot)$ is a sigmoid-based decay function designed to reduce the modulation factor for tokens with large normalized entropy changes:

$$f(x) = \lambda_{\min} + (1 - \lambda_{\min}) \cdot \sigma(-\gamma(x - \tau)). \tag{13}$$

In this formulation, $\sigma(z) = \frac{1}{1+\exp(-z)}$ is the sigmoid function, $\gamma > 0$ controls the transition sharpness between stable and unstable regions, $\tau \in [0, 1]$ determines the entropy change threshold triggering the decay activation, and $\lambda_{\min} = 1 - \alpha$ establishes a lower bound for the stabilization factor.

4.3.3 Integrated Advantage Modulation

Our complete approach integrates both compensation and stabilization mechanisms through a unified advantage modulation formulation as:

$$A_{i,t}^{\text{final}} = A_{i,t} \cdot \beta_{i,t}^{\text{comp}} \cdot \beta_{i,t}^{\text{stab}}. \tag{14}$$

Subsequently, the training objective incorporates these modulated advantages:

$$\mathcal{L}_{\text{DynaMO}}(\theta) = \mathbb{E}_{q\sim\mathcal{D},\{o_i\}_{i=1}^n\sim\pi_{\text{old}}(\cdot|q)}\left[\frac{1}{\sum_{i=1}^n |o_i|}\sum_{i=1}^n\sum_{t=1}^{|o_i|} \min\left(r_{i,t}A_{i,t}^{\text{final}}, \text{clip}(r_{i,t}, 1-\epsilon, 1+\epsilon)A_{i,t}^{\text{final}}\right)\right], \tag{15}$$

where n denotes the rollout budget allocated to prompt q via dynamic allocation, with the prompt subscript omitted for notational brevity.

This integrated approach addresses both challenges, leveraging the gradient-entropy relationship: the compensation mechanism maintains sufficient learning signals for high-confidence positive actions, while the stabilization mechanism prevents training instability by dampening tokens with excessive entropy changes that indicate large gradient magnitudes, with both mechanisms unified through a single hyperparameter α for simplified tuning.

Table 1: Comparison of benchmark results across Qwen2.5-Math-1.5B and Qwen2.5-Math-7B. Pass@K (%) is abbreviated as P@K. The best results are bold, and the second-best results are underlined, respectively.

Method	AIME24		AIME25		AMC23		MATH500		Minerva		Olympiad		Avg.	
	P@1	P@32	P@1	P@32	P@1	P@32	P@1	P@32	P@1	P@32	P@1	P@32	P@1	P@32
Qwen2.5-Math-1.5B														
GRPO	13.2	32.3	7.6	31.5	56.0	90.0	54.4	79.2	17.2	42.8	25.6	47.0	29.0	53.8
Clip-Higher	12.4	34.7	6.4	30.6	50.6	89.9	56.8	80.2	16.8	41.3	26.4	46.8	28.2	53.9
Entropy Loss	12.6	33.7	5.8	28.4	55.6	86.9	56.3	78.5	17.6	43.6	25.4	46.4	28.9	52.9
Fork Tokens	9.4	32.0	5.9	31.4	52.5	85.6	54.3	74.2	16.6	36.8	25.5	45.2	27.4	50.9
Entropy Advantages	15.7	35.8	8.9	33.4	62.0	86.4	**59.7**	76.2	18.2	43.0	25.9	44.9	31.7	53.3
Clip-COV	13.5	36.4	6.6	**34.4**	59.5	89.7	57.6	75.6	15.8	**44.3**	25.8	47.6	29.8	54.7
KL-COV	12.6	33.9	9.0	33.4	55.8	91.3	54.2	78.1	14.8	40.3	25.4	**48.1**	28.6	54.2
W-REINFORCE	15.3	35.3	8.5	31.7	63.0	85.7	56.7	77.7	18.2	40.3	24.4	46.2	31.0	52.8
DynaMO (Ours)	**17.2**	**37.2**	**9.8**	32.5	**63.6**	**91.9**	58.8	**81.0**	**19.4**	44.0	**27.2**	47.1	**32.7**	**55.6**
Qwen2.5-Math-7B														
GRPO	28.8	52.5	11.7	34.8	68.3	90.8	63.3	75.0	22.6	45.4	28.6	44.7	37.2	57.2
Clip-Higher	27.0	51.9	12.1	39.5	67.8	89.9	64.2	83.6	24.0	46.1	28.1	46.3	37.2	59.6
Entropy Loss	30.6	54.6	13.2	40.6	66.0	87.0	60.6	79.6	23.3	45.9	30.2	41.1	37.3	58.1
Fork Tokens	27.1	52.5	13.4	43.5	71.0	87.3	65.8	79.3	26.1	42.4	30.9	47.3	39.1	58.7
Entropy Advantages	27.5	49.7	9.4	39.2	67.9	85.2	65.3	83.3	23.7	43.7	30.4	47.3	37.4	58.1
Clip-COV	32.2	52.7	13.2	40.4	72.7	89.3	64.3	76.8	25.4	45.9	29.5	44.6	39.5	58.3
KL-COV	32.8	53.3	11.7	36.1	70.6	88.5	64.6	75.3	24.5	39.9	30.2	44.2	39.1	56.2
W-REINFORCE	31.8	55.4	14.3	41.0	72.5	89.8	64.9	**84.0**	26.4	**49.5**	30.9	46.7	40.1	61.1
DynaMO (Ours)	**34.4**	**59.0**	**15.4**	**46.8**	**74.4**	**92.9**	**66.4**	**84.0**	**27.3**	47.2	**31.6**	**50.1**	**41.6**	**63.3**

5 Experiments

5.1 Experimental Setup

Training Configuration. We conduct experiments on three different LLM scales: Qwen2.5-Math-1.5B, Qwen2.5-Math-7B, and Qwen3-14B. Our implementation builds upon the VeRL training framework (Sheng et al., 2025), adapting it to incorporate our dual-pronged approach. Our training data is DAPO-Math-17k (Yu et al., 2025a), which consists of mathematical reasoning problems with verifiable integer answers, ensuring reliable reward signals for RLVR optimization. Training is performed with top-p sampling at p=1.0 and temperature set to 1.0 to maintain exploration diversity. More details are provided in Appendix E.

Benchmarks and Metrics. We evaluate DynaMO on six mathematical reasoning benchmarks: AIME24, AIME25 (MAA, 2025), AMC23 (MAA, 2023), MATH500 (Hendrycks et al., 2021), Minerva (Lewkowycz et al., 2022), and Olympiad (He et al., 2024), covering various problem difficulties and types. Detailed descriptions of these benchmarks are presented in Appendix F.

We report average Pass@1 and Pass@32 across three independent training runs with different random seeds, using Math-Verify (HuggingFace, 2025) for answer verification and top-p=1.0, temperature=1.0 for diverse generation. Following Yue et al. (2025), Pass@K equals 1 if at least one of K sampled outputs passes verification, with unbiased estimation from Chen et al. (2021) to mitigate variance.

Baseline Methods. We compare DynaMO with strong baselines, including standard GRPO (Shao et al., 2024) and various entropy intervention techniques: Clip-Higher (Yu et al., 2025a), Entropy Loss (Cheng et al., 2025), Fork Tokens (Wang et al., 2025), Entropy Advantages (Cheng et al., 2025), coverage-based methods (Clip-COV, KL-COV) (Zhu et al., 2025), and W-REINFORCE (Tan and Pan, 2025). All experiments use consistent hyperparameters for fair comparison.

5.2 Main Results

Table 1 presents the comprehensive comparison results across six mathematical reasoning benchmarks on Qwen2.5-Math-1.5B and Qwen2.5-Math-7B. DynaMO consistently outperforms all baseline methods by effectively addressing two complementary challenges: the variance-driven rollout alloca-

Table 2: Ablation study on Qwen2.5-Math-7B with Pass@1 (%). DRA: Dynamic Rollout Allocation, UMS: Update Magnitude Stabilization, GC: Gradient Compensation, w/o ALL: standard GRPO

Method	AIME24	AIME25	AMC23	MATH500	Minerva	Olympiad	Avg.
DynaMO	**34.4**	**15.4**	**74.4**	**66.4**	**27.3**	**31.6**	**41.6**
- w/o GC	33.8	15.0	71.9	65.0	26.7	29.3	40.3
- w/o UMS	33.2	14.7	71.9	64.9	25.9	30.2	40.1
- w/o DRA	31.9	15.2	73.4	65.7	23.0	30.4	39.9
- w/o GC & DRA	31.9	14.5	69.0	64.4	23.7	29.7	38.9
- w/o GC & UMS	30.5	14.3	70.2	63.5	22.2	29.4	38.4
- w/o UMS & DRA	30.0	13.8	70.1	61.6	19.9	30.2	37.6
- w/o ALL	28.8	11.7	68.3	63.3	22.6	28.6	37.2

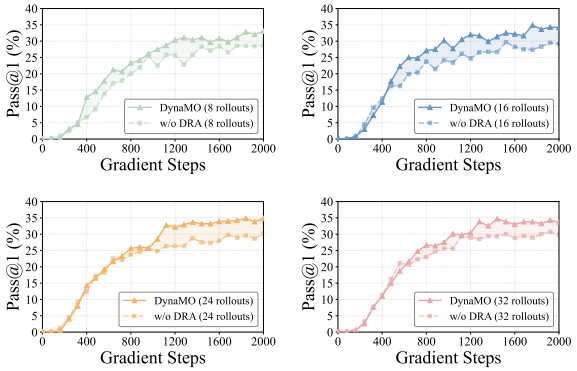


Figure 2: Impact of DRA across different computational budgets on AIME24 with average Pass@1 (%). Solid lines denote the full DynaMO, and dashed lines are the variant that substitutes DRA with uniform allocation.

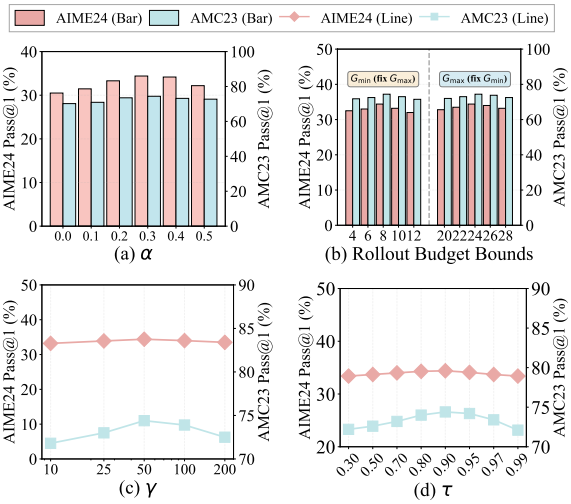


Figure 3: Hyperparameter sensitivity analysis on Qwen2.5-Math-7B across AIME24 and AMC23.

tion concentrates computational budget on problems with balanced success-failure distributions where Bernoulli variance signals high gradient informativeness, while the gradient-aware advantage modulation provides compensation for gradient attenuation in high-confidence correct actions and stabilization against excessive update magnitudes signaled by large entropy changes. Furthermore, a comparison with entropy intervention baselines reveals their critical limitations: coarse-grained clipping and sequence-level reweighting methods like Clip-Higher, Clip-COV, and KL-COV lack fine-grained control over token-level dynamics, while entropy-induced advantage methods such as Entropy Advantages and W-REINFORCE introduce training instability without principled stabilization mechanisms. Results on Qwen3-14B demonstrating effective scaling are presented in Section 5.5.

5.3 Ablation Study

5.3.1 Overall Component Analysis

Table 2 conducts systematic ablation on Qwen2.5-Math-7B. DynaMO integrates three components: Dynamic Rollout Allocation (DRA), Update Magnitude Stabilization (UMS), and Gradient Compensation (GC). Removing individual components degrades average performance, with DRA showing the largest impact on Minerva, GC on Olympiad, and UMS providing stability across benchmarks. Removing multiple components reveals synergistic effects: *w/o* GC & UMS degrades more than the sum of individual removals, indicating UMS stabilizes the amplified gradients from GC.

5.3.2 Impact of Dynamic Rollout Allocation

Figure 2 evaluates DRA across varying computational budgets, specifically ranging from an average of 8 to 32 rollouts per problem. The results demonstrate that DRA provides stable performance gains across all configurations. During early training, limited historical statistics result in similar allocations across problems. However, as variance data accumulates, DRA progressively concentrates the budget on problems with balanced success-failure distributions where Bernoulli variance peaks. Crucially, these problems reside within the capability gap, being neither trivially solved nor currently inaccessible, where both positive and negative advan-

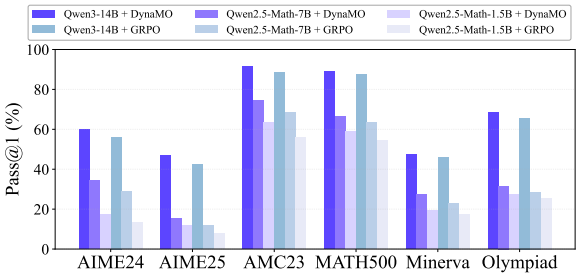


Figure 4: Performance comparison across different LLM scales (1.5B/7B/14B) with DynaMO and GRPO.

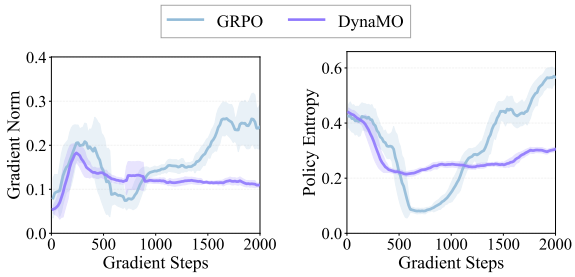


Figure 5: Training dynamics comparison. DynaMO maintains stable gradient norms and smooth entropy evolution, while GRPO exhibits severe spikes and fluctuations. Shaded regions show raw data range.

tage signals are actively generated, thereby maximizing learning signal quality per computational unit. In contrast, uniform allocation continues wasting resources on problems outside this optimal learning zone throughout training.

5.4 Hyperparameter Analysis

Figure 3 illustrates the hyperparameter sensitivity analysis on Qwen2.5-Math-7B across AIME24 and AMC23. The unified modulation parameter α exhibits an inverted-U pattern: performance degrades without modulation at $\alpha = 0$, peaks at moderate values, and subsequently declines at excessive settings, validating that insufficient modulation fails to address gradient attenuation while over-intervention constrains the learning process. Allocation bounds $G_{\min}$ and $G_{\max}$ demonstrate stable performance across wide ranges, confirming variance-driven allocation avoids under-sampling issues with minimal tuning. Similarly, the stabilization sharpness γ exhibits a flat plateau around its optimal values, balancing selective intervention against learning flexibility. Entropy threshold τ displays smooth variation characterized by a broad optimal region, thereby demonstrating the reliability in identifying problematic tokens.

5.5 Scaling to Larger Models

To validate scalability, we extend experiments to Qwen3-14B across all benchmarks in Figure 4. DynaMO consistently outperforms GRPO with widening gaps as scale increases: moderate gains at 1.5B, amplified substantially at 7B, and expanded further at 14B. This trend confirms that our theoretically-grounded mechanisms become increasingly effective at larger scales, as variance-driven allocation and gradient-aware modulation better exploit the enhanced representational capacity while mitigating intensified optimization instabilities.

6 Case Study

To provide a deeper insight into the effectiveness of DynaMO, we compare it with GRPO through training dynamics analysis. As visually depicted in Figure 5, GRPO suffers from severe gradient spikes and erratic entropy fluctuations, whereas DynaMO maintains stable dynamics across both metrics. Two core components of DynaMO contribute to this stability: dynamic rollout allocation concentrates the budget on high-variance problems, while gradient-aware modulation prevents excessive update magnitudes. Notably, despite following similar overall trends, DynaMO achieves substantially smoother transitions in gradient norms and policy entropy, indicating more controlled optimization process throughout training. These phenomena confirm that DynaMO mitigates training instability at both sequence and token levels while simultaneously achieving superior performance.

7 Conclusion

In this paper, we propose **DynaMO**, a theoretically-grounded dual-pronged framework designed to systematically address fundamental RLVR challenges. *At the sequence level*, we prove that uniform allocation is suboptimal and subsequently derive a variance-minimizing allocation strategy to concentrate computational resources on high-informativeness problems. *At the token level*, we establish the gradient-entropy relationship, enabling an integrated advantage modulation mechanism that compensates for gradient attenuation while stabilizing excessive updates. Extensive experiments conducted across multiple reasoning benchmarks and varying LLM scales demonstrate consistent improvements over strong baselines, with comprehensive ablation studies validating the independent contributions from both mechanisms.

Limitations

Our evaluation is conducted on the Qwen model family, spanning three distinct scales (1.5B, 7B, and 14B). Although we have not systematically evaluated other model families, it is important to note that our method operates at the algorithmic level—modifying rollout allocation and advantage computation—without relying on any architecture-specific features, suggesting transferability potential. Future work could extend this evaluation to additional model architectures and scales, as well as multimodal reasoning scenarios where robustness against adversarial attacks (Li et al., 2026b) becomes critical.

References

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, and 1 others. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *International Conference on Machine Learning (ICML)*.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, and 1 others. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Daixuan Cheng, Shaohan Huang, Xuekai Zhu, Bo Dai, Wayne Xin Zhao, Zhenliang Zhang, and Furu Wei. 2025. Reasoning with exploration: An entropy perspective. *arXiv preprint arXiv:2506.14758*.

Xiangxiang Chu, Hailang Huang, Xiao Zhang, Fei Wei, and Yong Wang. 2025. Gpg: A simple and strong reinforcement learning baseline for model reasoning. *arXiv preprint arXiv:2504.02546*.

Hanze Dong, Wei Xiong, Deepanshu Goyal, Yihan Zhang, Winnie Chow, Rui Pan, Shizhe Diao, Jipeng Zhang, Kashun Shum, and Tong Zhang. 2023. Raft: Reward ranked finetuning for generative foundation model alignment. *arXiv preprint arXiv:2304.06767*.

Tianqing Fang, Zhisong Zhang, Xiaoyang Wang, Rui Wang, Can Qin, Yuxuan Wan, Jun-Yu Ma, Ce Zhang, Jiaqi Chen, Xiyun Li, Hongming Zhang, Haitao Mi, and Dong Yu. 2025. Cognitive kernel-pro: A framework for deep research agents and agent foundation models training. *CoRR*, abs/2508.00414.

Yangyi Fang, Jiaye Lin, Xiaoliang Fu, Cong Qin, and Haolin Shi. 2026. Placing puzzle pieces where they matter: A question augmentation framework for reinforcement learning. *Preprint*, arXiv:2604.15830.

Caglar Gulcehre, Tom Le Paine, Srivatsan Srinivasan, Ksenia Konyushkova, Lotte Weerts, Abhishek Sharma, Aditya Siddhant, Alex Ahern, Miaosen Wang, Chenjie Gu, and 1 others. 2023. Reinforced self-training (rest) for language modeling. *arXiv preprint arXiv:2308.08998*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. 2018. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning (ICML)*.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, and 1 others. 2024. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. *arXiv preprint arXiv:2402.14008*.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*.

Jingcheng Hu, Yinmin Zhang, Qi Han, Daxin Jiang, Xiangyu Zhang, and Heung-Yeung Shum. 2025. Open-reasoner-zero: An open source approach to scaling up reinforcement learning on the base model. *arXiv preprint arXiv:2503.24290*.

HuggingFace. 2025. Math-verify.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, and 1 others. 2024. Openai o1 system card. *arXiv preprint arXiv:2412.16720*.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, and 1 others. 2022. Solving quantitative reasoning problems with language models. *Conference on Neural Information Processing Systems (NeurIPS)*.

Mukai Li, Qingcheng Zeng, Tianqing Fang, Zhenwen Liang, Linfeng Song, Qi Liu, Haitao Mi, and Dong Yu. 2026a. Verified critical step optimization for LLM agents. *CoRR*, abs/2602.03412.

Yingru Li. 2025. Logit dynamics in softmax policy gradient methods. *arXiv preprint arXiv:2506.12912*.

Zhiheng Li, Zongyang Ma, Yuntong Pan, Ziqi Zhang, Xiaolei Lv, Bo Li, Jun Gao, Jianing Zhang, Chunfeng Yuan, Bing Li, and Weiming Hu. 2026b. Making mllms blind: Adversarial smuggling attacks in mllm content moderation. *Preprint*, arXiv:2604.06950.

Peiyang Liu, Sen Wang, Xi Wang, Wei Ye, and Shikun Zhang. 2021a. Quadrupletbert: An efficient model for embedding-based large-scale retrieval. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3734–3739.

Peiyang Liu, Xi Wang, Sen Wang, Wei Ye, Xiangyu Xi, and Shikun Zhang. 2021b. Improving embedding-based large-scale retrieval via label enhancement. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 133–142.

Peiyang Liu, Xiangyu Xi, Wei Ye, and Shikun Zhang. 2022. Label smoothing for text mining. In *Proceedings of the 29th international conference on computational linguistics*, pages 2210–2219.

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

Michael Luo, Sijun Tan, Justin Wong, Xiaoxiang Shi, William Y Tang, Manan Roongta, Colin Cai, Jeffrey Luo, Tianjun Zhang, Li Erran Li, and 1 others. 2025. Deepscaler: Surpassing o1-preview with a 1.5 b model by scaling rl. *Notion Blog*.

MAA. 2023. American mathematics competitions - amc.

MAA. 2025. American invitational mathematics examination - aime.

Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. 2016. Asynchronous methods for deep reinforcement learning. In *International Conference on Machine Learning (ICML)*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Conference on Neural Information Processing Systems (NeurIPS)*.

Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. 2015. Prioritized experience replay. *arXiv preprint arXiv:1511.05952*.

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

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. 2025. Hybridflow: A flexible and efficient rlhf framework. In *European Conference on Computer Systems (EuroSys)*.

Richard Sutton and Andrew Barto. 2018. Reinforcement learning: an introduction.

Hongze Tan and Jianfei Pan. 2025. Gtpo and grpo-s: Token and sequence-level reward shaping with policy entropy. *arXiv preprint arXiv:2508.04349*.

Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, and 1 others. 2025. Kimi k1. 5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599*.

Yuxuan Tong, Xiwen Zhang, Rui Wang, Ruidong Wu, and Junxian He. 2024. Dart-math: Difficulty-aware rejection tuning for mathematical problem-solving. *Conference on Neural Information Processing Systems (NeurIPS)*.

Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xionghui Chen, Jianxin Yang, Zhenru Zhang, and 1 others. 2025. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for llm reasoning. *arXiv preprint arXiv:2506.01939*.

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

Jiarui Yao, Yifan Hao, Hanning Zhang, Hanze Dong, Wei Xiong, Nan Jiang, and Tong Zhang. 2025. Optimizing chain-of-thought reasoners via gradient variance minimization in rejection sampling and rl. *arXiv preprint arXiv:2505.02391*.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, and 1 others. 2025a. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*.

Wenhao Yu, Zhenwen Liang, Chengsong Huang, Kishan Panaganti, Tianqing Fang, Haitao Mi, and Dong Yu. 2025b. Guided self-evolving llms with minimal human supervision. *CoRR*, abs/2512.02472.

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Shiji Song, and Gao Huang. 2025. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model? *arXiv preprint arXiv:2504.13837*.

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

A Derivation of Entropy Change Estimation for Gradient-Aware Advantage Modulation

We derive the first-order entropy change estimation for tabular softmax policy through a factorized decomposition approach. Let the actor policy π_θ be a tabular softmax policy where each state-action pair (s, a) is associated with an individual logit parameter $z_{s,a} = \theta_{s,a}$.

A.1 Preliminary Definitions

Definition 1 (Policy Concentration Measure)**.** *For a policy $\pi_\theta(\cdot|s)$ at state s, we define the concentration coefficient:*

$$\Phi(\pi_\theta|s) := \sum_a \pi_\theta(a|s)^2. \tag{16}$$

This measures the degree of policy concentration, with higher values indicating a more deterministic policy.

Definition 2 (Entropy-Weighted Log-Probability)**.** *We define the centered log-probability for action a under policy π_θ at state s:*

$$\Lambda_\theta(a|s) := \log \pi_\theta(a|s) + \mathcal{H}(\pi_\theta|s). \tag{17}$$

This represents the deviation of action a's log-probability from the policy's average entropy, capturing the information-theoretic distance from uniform randomness.

A.2 Entropy Gradient Derivation

We use Taylor's expansion under first-order approximation:

$$\begin{aligned}&\mathcal{H}(\pi_\theta^{k+1} \mid s)\\ &\approx \mathcal{H}(\pi_\theta^{k} \mid s)\\ &\quad + \langle \nabla \mathcal{H}(\pi_\theta^{k} \mid s), (z^{k+1} - z^{k}) \rangle.\end{aligned} \tag{18}$$

To derive $\nabla \mathcal{H}(\pi_\theta^k \mid s)$, we start from the definition of entropy:

$$\begin{aligned}&\nabla_\theta \mathcal{H}(\pi_\theta \mid s)\\ &= \nabla_\theta \left(-\mathbb{E}_{a\sim\pi_\theta(\cdot|s)}[\log \pi_\theta(a \mid s)]\right)\\ &= -\nabla_\theta \mathbb{E}_{a\sim\pi_\theta(\cdot|s)}[\log \pi_\theta(a \mid s)]\\ &= -\mathbb{E}_{a\sim\pi_\theta(\cdot|s)}[\nabla_\theta \log \pi_\theta(a \mid s)]\\ &\quad - \mathbb{E}_{a\sim\pi_\theta(\cdot|s)}[\log \pi_\theta(a \mid s)\\ &\qquad \times \nabla_\theta \log \pi_\theta(a \mid s)]\\ &= 0 - \mathbb{E}_{a\sim\pi_\theta(\cdot|s)}[\log \pi_\theta(a \mid s)\\ &\qquad \times \nabla_\theta \log \pi_\theta(a \mid s)]\\ &= -\mathbb{E}_{a\sim\pi_\theta(\cdot|s)}[\log \pi_\theta(a \mid s)\\ &\qquad \times \nabla_\theta \log \pi_\theta(a \mid s)].\end{aligned} \tag{19}$$

The first term equals zero because $\mathbb{E}_{a\sim\pi_\theta(\cdot|s)}[\nabla_\theta \log \pi_\theta(a \mid s)] = \nabla_\theta \sum_a \pi_\theta(a \mid s) = \nabla_\theta 1 = 0$.

A.3 Softmax Derivative and Centered Logit Updates

For a softmax policy $\pi_\theta(a \mid s) = \frac{\exp(z_{s,a})}{\sum_{a''} \exp(z_{s,a''})}$, the derivative of the log-probability with respect to any logit parameter $z_{s,a'}$ is:

$$\begin{aligned}\frac{\partial \log \pi_\theta(a \mid s)}{\partial z_{s,a'}} &= \mathbf{1}\{a = a'\}\\ &\quad - \pi_\theta(a' \mid s).\end{aligned} \tag{20}$$

We now introduce a centered form of logit updates that exploits the translation invariance of softmax:

Definition 3 (Centered Logit Change)**.** *Define the centered logit change as:*

$$\begin{aligned}\delta z_{s,a} &:= z_{s,a}^{k+1} - z_{s,a}^{k}\\ &\quad - \mathbb{E}_{a'\sim\pi_\theta^k(\cdot|s)}[z_{s,a'}^{k+1} - z_{s,a'}^{k}].\end{aligned} \tag{21}$$

This removes the global translation component, which does not affect softmax probabilities.

Using Equation (20), we can express the first-order entropy change:

$$\begin{aligned}&\langle \nabla_\theta \mathcal{H}(\theta^k \mid s), (z^{k+1} - z^k) \rangle\\ &= -\mathbb{E}_{a\sim\pi_\theta^k(\cdot|s)}\Big[\log \pi_\theta^k(a \mid s)\\ &\quad \times \sum_{a'} \left(\mathbf{1}\{a = a'\} - \pi_\theta^k(a' \mid s)\right)\\ &\quad \times (z_{s,a'}^{k+1} - z_{s,a'}^{k})\Big]\\ &= -\mathbb{E}_{a\sim\pi_\theta^k(\cdot|s)}\Big[\log \pi_\theta^k(a \mid s)\\ &\quad \times \left(z_{s,a}^{k+1} - z_{s,a}^{k}\right.\\ &\quad \left. - \sum_{a'} \pi_\theta^k(a' \mid s)(z_{s,a'}^{k+1} - z_{s,a'}^{k})\right)\Big]\\ &= -\mathbb{E}_{a\sim\pi_\theta^k(\cdot|s)}[\log \pi_\theta^k(a \mid s) \cdot \delta z_{s,a}].\end{aligned} \tag{22}$$

Expanding further and using the covariance decomposition $\mathbb{E}[XY] = \mathbb{E}[X]\mathbb{E}[Y] + \mathrm{Cov}(X, Y)$:

$$
\begin{aligned}
&\langle \nabla_\theta \mathcal{H}(\theta^k \mid s), (z^{k+1} - z^k) \rangle \\
&= -\mathbb{E}_{a\sim\pi_\theta^k(\cdot|s)}[\log \pi_\theta^k(a \mid s) \\
&\quad \times (z_{s,a}^{k+1} - z_{s,a}^k)] \\
&\quad + \mathbb{E}_{a\sim\pi_\theta^k(\cdot|s)}[\log \pi_\theta^k(a \mid s)] \\
&\quad \times \mathbb{E}_{a'\sim\pi_\theta^k(\cdot|s)}[z_{s,a'}^{k+1} - z_{s,a'}^k] \\
&= -\mathbb{E}_{a\sim\pi_\theta^k(\cdot|s)}\Big[\big(\log \pi_\theta^k(a \mid s) \\
&\quad - \mathbb{E}_{a\sim\pi_\theta^k(\cdot|s)}[\log \pi_\theta^k(a \mid s)]\big) \\
&\quad \times \delta z_{s,a}\Big] \\
&= -\mathbb{E}_{a\sim\pi_\theta^k(\cdot|s)}[\Lambda_\theta^k(a|s) \cdot \delta z_{s,a}],
\end{aligned}
\tag{23}
$$

where we used Definition 2 in the last step, noting that $-\mathbb{E}[\log \pi_\theta] = \mathcal{H}(\pi_\theta|s)$.

A.4 GRPO Update Coefficient

For GRPO, we define the *composite update coefficient*:

$$
\xi_{i,t}(a) := \mathbb{I}_{\text{clip}} \cdot r_{i,t} \cdot A_{i,t}, \tag{24}
$$

where $\mathbb{I}_{\text{clip}}$ is the clipping indicator function, $r_{i,t} = \frac{\pi_\theta(a_{i,t}|s_{i,t})}{\pi_{\text{ref}}(a_{i,t}|s_{i,t})}$ is the importance sampling ratio, and $A_{i,t}$ is the advantage estimate. This coefficient combines three essential components: clipping for stability, importance weighting for off-policy correction, and advantage for policy improvement direction.

Lemma 1 (Centered Logit Update for GRPO)**.** *Under the GRPO update rule, the per-sample stochastic logit update satisfies:*

$$
z_{s,k}^{k+1} - z_{s,k}^k = \eta\, \xi_{i,t}\, \big(\mathbf{1}\{a = k\} - \pi_\theta(k \mid s)\big). \tag{25}
$$

Specifically, for the sampled action a and other actions $a' \neq a$:

$$
\begin{aligned}
z_{s,a}^{k+1} - z_{s,a}^k &= \eta\, \xi_{i,t}\, (1 - \pi_\theta(a \mid s)) \\
z_{s,a'}^{k+1} - z_{s,a'}^k &= -\eta\, \xi_{i,t}\, \pi_\theta(a' \mid s).
\end{aligned}
\tag{26}
$$

Remark 1. *By the definition of advantage functions (Sutton and Barto, 2018) and GRPO's group normalization (Shao et al., 2024), advantages satisfy $\mathbb{E}_{a\sim\pi_\theta(\cdot|s)}[A_{i,t}] = 0$.*

A.5 Factorized Entropy Change Formula

Substituting Lemma 1 into Equation (23):

$$
\begin{aligned}
&\langle \nabla_\theta \mathcal{H}(\theta^k \mid s), (z^{k+1} - z^k) \rangle \\
&= -\sum_k \pi_\theta^k(k \mid s)\, \Lambda_\theta^k(k|s) \big(z_{s,k}^{k+1} - z_{s,k}^k\big) \\
&= -\eta\, \xi_{i,t} \Big[\pi_\theta^k(a \mid s)\, \Lambda_\theta^k(a|s) \\
&\quad - \sum_k \pi_\theta^k(k \mid s)^2\, \Lambda_\theta^k(k|s)\Big].
\end{aligned}
\tag{27}
$$

Taking expectation over $a \sim \pi_\theta^k(\cdot \mid s)$ under GRPO's group normalization, the entropy change is dominated by:

$$
\Delta\mathcal{H}(\pi_\theta^k|s) \approx -\eta\, \mathbb{E}_{a\sim\pi_\theta^k}\Big[\pi_\theta^k(a \mid s) \Lambda_\theta^k(a|s)\, \xi_{i,t}(a)\Big]. \tag{28}
$$

We now state our main result:

Theorem 1 (Factorized Entropy Change)**.** *Following GRPO's design (Shao et al., 2024) with standard clipping, the first-order entropy change admits:*

$$
\boxed{
\begin{aligned}
\Delta\mathcal{H}(\pi_\theta^k|s) \approx -\eta \sum_a \underbrace{\pi_\theta^k(a|s)^2}_{\textit{Concentration}} \\
\times \underbrace{\Lambda_\theta^k(a|s)}_{\textit{Info Deviation}} \\
\times \underbrace{\xi_{i,t}(a)}_{\textit{Update Coeff.}}
\end{aligned}
}
\tag{29}
$$

where $\Lambda_\theta^k(a|s) = \log \pi_\theta^k(a|s) + \mathcal{H}(\pi_\theta^k|s)$ (Definition 2), and:

- $\pi_\theta^k(a|s)^2$*: Policy concentration*
- $\Lambda_\theta^k(a|s)$*: Info-theoretic deviation*
- $\xi_{i,t}(a) = \mathbb{I}_{clip} \cdot r_{i,t} \cdot A_{i,t}$*: Update coefficient (Eq. 24)*

Proof. From Equation (28):

$$
\begin{aligned}
&\Delta\mathcal{H}(\pi_\theta^k|s) \\
&\approx -\eta\, \mathbb{E}_{a\sim\pi_\theta^k}\Big[\pi_\theta^k(a \mid s) \Lambda_\theta^k(a|s)\, \xi_{i,t}(a)\Big] \\
&= -\eta \sum_a \pi_\theta^k(a|s) \cdot \pi_\theta^k(a \mid s) \times \Lambda_\theta^k(a|s)\, \xi_{i,t}(a) \\
&= -\eta \sum_a \pi_\theta^k(a|s)^2 \Lambda_\theta^k(a|s) \cdot \xi_{i,t}(a),
\end{aligned}
\tag{30}
$$

where the second line expands the expectation as $\mathbb{E}_{a\sim\pi}[f(a)] = \sum_a \pi(a|s) \cdot f(a)$, and the third line simplifies the product. Substituting Definition 2 yields the boxed form. □

Remark 2. *This factorization reveals three orthogonal mechanisms governing entropy dynamics: (i) concentration amplifies updates for high-probability actions, (ii) information deviation directs change based on distance from maximum entropy, and (iii) update coefficient modulates the magnitude via clipped importance-weighted advantages. The multiplicative structure implies that entropy change vanishes when any factor approaches zero, providing natural regularization.*

B Gradient-Entropy Relationship: Proof of Expected Gradient Norm Upper Bound

This appendix establishes the relationship between the score function's squared L2 norm and policy entropy in softmax policies, then extends it to advantage-weighted updates.

Core Result. For softmax policy $\pi_\theta(\cdot|s) = (\pi_1, \ldots, \pi_{|\mathcal{V}|})$ with logits $z(s) = (z_1(s), \ldots, z_{|\mathcal{V}|}(s))$ where $\pi_k = \exp(z_k)/\sum_j \exp(z_j)$, the gradient of $\log \pi_k$ with respect to logits is $\frac{\partial \log \pi_k}{\partial z_i} = \delta_{ik} - \pi_i$ (Kronecker delta δ_{ik}).

The squared norm for action k is:

$$\begin{aligned} \|\nabla_z \log \pi_k\|^2 &= \sum_{i=1}^{|\mathcal{V}|} (\delta_{ik} - \pi_i)^2 \\ &= (1 - \pi_k)^2 + \sum_{i \neq k} \pi_i^2 \\ &= 1 - 2\pi_k + \sum_{j=1}^{|\mathcal{V}|} \pi_j^2. \end{aligned} \tag{31}$$

Taking expectation over actions sampled from π_θ:

$$\begin{aligned} &\mathbb{E}_{a_k \sim \pi_\theta(\cdot|s)} \left[\|\nabla_z \log \pi_k\|^2 \right] \\ &= \sum_{k=1}^{|\mathcal{V}|} \pi_k \left(1 - 2\pi_k + \sum_{j=1}^{|\mathcal{V}|} \pi_j^2 \right) \\ &= \sum_{k=1}^{|\mathcal{V}|} \pi_k - 2 \sum_{k=1}^{|\mathcal{V}|} \pi_k^2 \\ &\quad + \left(\sum_{j=1}^{|\mathcal{V}|} \pi_j^2 \right) \underbrace{\sum_{k=1}^{|\mathcal{V}|} \pi_k}_{=1} \\ &= 1 - \sum_{k=1}^{|\mathcal{V}|} \pi_k^2. \end{aligned} \tag{32}$$

The term $\sum_{k=1}^{|\mathcal{V}|} \pi_k^2$ is the collision probability (probability that two independent samples from π_θ are identical), ranging from $1/|\mathcal{V}|$ (uniform) to 1 (deterministic).

Entropy Upper Bound. To connect collision probability to Shannon entropy, apply Jensen's inequality. Since $x \mapsto \log x$ is concave, $\sum_k \pi_k \log \pi_k \leq \log(\sum_k \pi_k^2)$. Multiplying by -1 and exponentiating:

$$\begin{aligned} \mathcal{H}(\pi_\theta|s) &= -\sum_{k=1}^{|\mathcal{V}|} \pi_k \log \pi_k \\ &\geq -\log \left(\sum_{k=1}^{|\mathcal{V}|} \pi_k^2 \right) \\ &\Rightarrow \sum_{k=1}^{|\mathcal{V}|} \pi_k^2 \geq e^{-\mathcal{H}(\pi_\theta|s)}. \end{aligned} \tag{33}$$

Thus $\mathbb{E}_{a_k \sim \pi_\theta(\cdot|s)}[\|\nabla_z \log \pi_k\|^2] = 1 - \sum_{k=1}^{|\mathcal{V}|} \pi_k^2 \leq 1 - e^{-\mathcal{H}(\pi_\theta|s)}$. High entropy ($\mathcal{H}(\pi_\theta|s) \gg 0$) yields $e^{-\mathcal{H}} \approx 0$ and gradient norm ≈ 1 (vigorous learning); low entropy ($\mathcal{H}(\pi_\theta|s) \approx 0$) yields $e^{-\mathcal{H}} \approx 1$ and gradient norm ≈ 0 (stable convergence).

Advantage-Weighted Update Magnitude. For policy gradient update with learning rate η and advantage A, the logit update for sampled action a_k is $\Delta z(s) = \eta A(\mathbf{e}_k - \pi_\theta(\cdot|s))$ where $\mathbf{e}_k$ is the one-hot vector. The squared L2 norm is:

$$\|\Delta z(s)\|_2^2 = \eta^2 A^2 \Big(1 - 2\pi_k + \sum_{j=1}^{|\mathcal{V}|} \pi_j^2 \Big). \tag{34}$$

Taking expectation over $a_k \sim \pi_\theta$ gives $\mathbb{E}[\|\Delta z(s)\|_2^2] = \eta^2 \mathbb{E}[A^2(1 - 2\pi_k + \sum_j \pi_j^2)]$. Following GRPO's design (Shao et al., 2024) where advantages are sequence-level, the dependence between A and per-token probabilities π_k is negligible, allowing the approximation:

$$\begin{aligned} &\mathbb{E}_{a_k \sim \pi_\theta(\cdot|s)} \left[\|\Delta z(s)\|_2^2 \right] \\ &= \eta^2 \mathbb{E}[A^2] \left(1 - \sum_{k=1}^{|\mathcal{V}|} \pi_k^2 \right) \\ &\leq \eta^2 \mathbb{E}[A^2] \Big(1 - \exp\big(-\mathcal{H}(\pi_\theta|s) \big) \Big). \end{aligned} \tag{35}$$

This combines logit dynamics with advantage-weighted scaling and entropy bounds. High-confidence tokens ($\pi_k \approx 1$) produce small expected update magnitudes ($\sum_j \pi_j^2 \approx 1$), creating gradient attenuation for confident correct actions—the theoretical basis for gradient compensation in our method.

C Gradient Variance Minimization for Dynamic Rollout Allocation

This appendix provides the complete derivation of the optimal rollout allocation formula.

Optimal Allocation via Lagrange Multipliers. For prompt q_i with n_i rollouts, the gradient estimator $\hat{g}_i = \frac{1}{n_i}\sum_{k=1}^{n_i} g_{i,k}$ (where $g_{i,k} = \frac{1}{G}\sum_{j=1}^{G} \hat{A}_{j,k}\nabla_\theta \log \pi_\theta(o_{j,k}|q_i)$ with $\hat{A}_{j,k}$ the group-normalized advantage) has variance $\text{Var}[\hat{g}_i] = \sigma_i^2/n_i$ where $\sigma_i^2 = \mathbb{E}[\|g_{i,k}\|_2^2]$ is the single-sample gradient variance.

Minimizing total variance $\sum_{i=1}^{N} \sigma_i^2/n_i$ subject to budget constraint $\sum_i n_i = B$, we construct Lagrangian $\mathcal{L} = \sum_i \sigma_i^2/n_i + \lambda(\sum_i n_i - B)$. The first-order condition $\partial\mathcal{L}/\partial n_i = -\sigma_i^2/n_i^2 + \lambda = 0$ gives $n_i = \sigma_i/\sqrt{\lambda}$. Substituting into the constraint $\sum_i \sigma_i/\sqrt{\lambda} = B$ yields $\sqrt{\lambda} = (\sum_k \sigma_k)/B$, thus:

$$n_i^* = B \cdot \frac{\sigma_i}{\sum_{k=1}^{N} \sigma_k}. \tag{36}$$

Variance Decomposition. To implement this principle, we decompose $\sigma_i^2 = \mathbb{E}[\|g_{i,k}\|_2^2]$ where $g_{i,k} = \frac{1}{G}\sum_{j=1}^{G} A_{j,k}\nabla_\theta \log \pi_\theta(o_{j,k}|q_i)$. The squared norm is:

$$\|g_{i,k}\|_2^2 = \frac{1}{G^2}\Big\|\sum_{j=1}^{G} A_{j,k} \times \nabla_\theta \log \pi_\theta(o_{j,k}|q_i)\Big\|_2^2. \tag{37}$$

Following the treatment in high-dimensional gradient estimation (Schulman et al., 2017; Yao et al., 2025), $\mathbb{E}[\langle \nabla_j, \nabla_{j'} \rangle] \approx 0$, yielding:

$$\mathbb{E}[\|g_{i,k}\|_2^2] \approx \frac{1}{G}\mathbb{E}\Big[A^2 \|\nabla_\theta \log \pi_\theta(o|q_i)\|_2^2\Big]. \tag{38}$$

For GRPO with group-normalized advantages (Shao et al., 2024), the gradient variance exhibits positive correlation with both reward variance and expected gradient magnitude:

$$\mathbb{E}[\|g_{i,k}\|_2^2] \propto \text{Var}(R)\mathbb{E}[\|\nabla_\theta \log \pi_\theta(o|q_i)\|_2^2]. \tag{39}$$

For response $o = \{o_1, \ldots, o_T\}$, the policy gradient decomposes as $\nabla_\theta \log \pi_\theta(o|q_i) = \sum_{t=1}^{T} \nabla_\theta \log \pi_\theta(o_t|q_i, o_{<t})$. Applying the same approximation to token-level gradients:

$$\mathbb{E}[\|\nabla_\theta \log \pi_\theta(o|q_i)\|_2^2] \approx \mathbb{E}\Big[\sum_{t=1}^{T} \|\nabla_\theta \log \pi_\theta(o_t|q_i, o_{<t})\|_2^2\Big]. \tag{40}$$

From Appendix B, the expected squared gradient norm for token t is $\mathbb{E}[\|\nabla_\theta \log \pi_\theta(o_t|q_i, o_{<t})\|_2^2] = 1 - C(P_t)$ where $C(P_t) = \sum_{k=1}^{|\mathcal{V}|} \pi_k^2(q_i, o_{<t})$ is the collision probability. Defining average collision $\bar{C}_i = \mathbb{E}_{o\sim\pi_\theta(\cdot|q_i)}[\frac{1}{|o|}\sum_{t=1}^{|o|} C(P_t)]$:

$$\sigma_i^2 \propto \text{Var}(R) \cdot (1 - \bar{C}_i). \tag{41}$$

Computing $C(P_t) = \sum_{k=1}^{|\mathcal{V}|} \pi_k^2$ for all tokens is expensive. Since high entropy corresponds to low collision probability (uniform distributions have low $C(P)$, concentrated distributions have high $C(P)$), we use entropy as a proxy. Defining average per-token entropy $\bar{\mathcal{H}}_i = \mathbb{E}_{o\sim\pi_\theta(\cdot|q_i)}\Big[\frac{1}{|o|}\sum_{t=1}^{|o|} \mathcal{H}(\pi_\theta|q_i, o_{<t})\Big]$, we approximate:

$$\sigma_i \propto \text{std}(R) \cdot f(\bar{\mathcal{H}}_i) \tag{42}$$

where $f(\cdot)$ is an increasing function. Using $f(\mathcal{H}) = \sqrt{1 - e^{-\mathcal{H}}}$ captures the relationship: higher entropy leads to larger gradient variance.

Connection to Bernoulli Variance. For binary rewards, $\text{Var}(R) = p(1-p)$ where p is success probability. The unbiased estimator using historical statistics (k_i correct responses out of G_i total rollouts) is:

$$P_i = \frac{k_i(G_i - k_i)}{G_i(G_i - 1)}, \quad \mathbb{E}[P_i] = p_i(1 - p_i). \tag{43}$$

This P_i serves as a proxy for σ_i because: (i) it directly estimates $\text{Var}(R)$ through Bernoulli variance, (ii) empirical accuracy $\hat{p}_i = k_i/G_i$ near the variance-maximizing value correlates with high policy entropy (models actively exploring multiple solution paths), and (iii) together, P_i identifies high-σ_i problems per the variance decomposition. The water-level allocation algorithm implements $n_i^* \propto P_i$ with constraints $G_{\min}, G_{\max}$.

Variance Reduction Guarantee. The optimal allocation achieves total variance $\text{Var}^* = \frac{1}{B}(\sum_i \sigma_i)^2$, while uniform allocation ($n_i = B/N$) yields $\text{Var}_{\text{uniform}} = \frac{N}{B}\sum_i \sigma_i^2$. Their ratio is:

$$\frac{\text{Var}^*}{\text{Var}_{\text{uniform}}} = \frac{(\sum_i \sigma_i)^2}{N\sum_i \sigma_i^2} \leq 1, \tag{44}$$

where the inequality follows from Cauchy-Schwarz, with equality only when all σ_i are equal. When gradient variances are heterogeneous across problems (typical in RLVR), this adaptive allocation provides substantial variance reduction.

D Dynamic Rollout Allocation: Operational Details.

Our allocation mechanism determines rollout quantities for each prompt in the training batch, operating independently of prompt selection strategies (e.g., random sampling, curriculum learning). Given a batch of N prompts and total rollout budget B, the water-level algorithm distributes n_i rollouts per prompt proportionally to variance proxies

P_i while respecting minimum and maximum constraints.

The allocation process guarantees complete budget utilization through iterative refinement. Initially, each prompt receives a baseline allocation based on its P_i value. When constraints become active (some prompts hit minimum or maximum bounds), residual budget is redistributed among unconstrained prompts using the same proportional rule. This continues until all budget is assigned and $\sum_i n_i = B$ holds exactly. If the total minimum requirement exceeds available budget, we proportionally scale down the minimum allocation to ensure feasibility.

For prompts with insufficient historical data (few prior observations), computing reliable variance proxies P_i is infeasible. In such cases, we apply uniform allocation as a fallback strategy, distributing rollouts equally among data-scarce prompts. As prompts accumulate responses through repeated selection in training batches, their historical statistics become sufficient to support adaptive allocation. This transition from uniform to variance-driven distribution occurs naturally as the P_i estimates stabilize with increased sample counts.

The proportional allocation naturally adapts to estimation errors. If historical P_i overestimates current variance (due to policy improvement), subsequent rollouts reveal lower empirical variance, which updates P_i for future allocations. Conversely, underestimated prompts exhibit higher gradient variance, triggering increased allocation in later iterations. This feedback loop ensures the distribution tracks evolving problem difficulty without manual tuning. In boundary cases where all P_i values become negligibly small (indicating near-deterministic outcomes), the algorithm defaults to uniform distribution to maintain numerical stability.

E Detailed Training Configuration

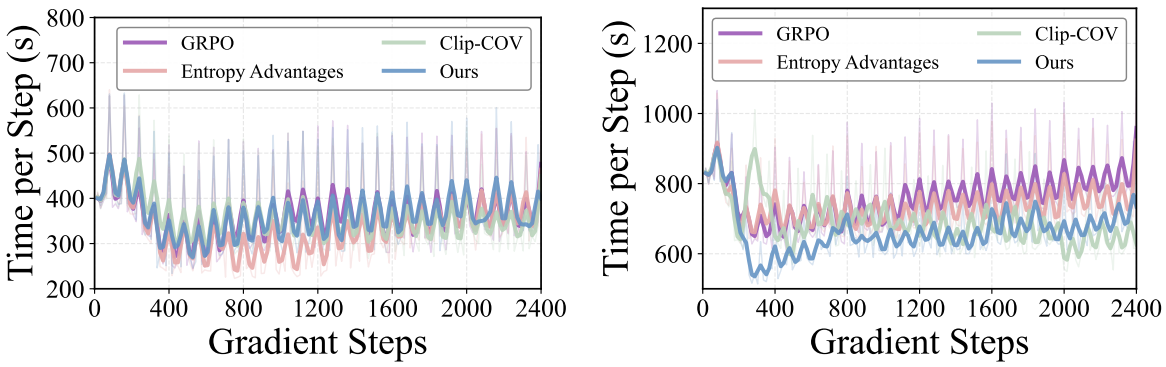
All experiments are conducted on 8× NVIDIA A100 80GB GPUs per node (1-2 nodes depending on model scale) with mixed precision training (bfloat16) and FSDP for distributed training. The actor model uses Tensor Parallelism (TP=2) during rollout generation with vLLM for efficient inference. Training prompts are left-truncated to 2048 tokens if exceeding the maximum length, and we apply Math-Verify (HuggingFace, 2025) for answer verification during training and evaluation. The training dataset is shuffled at the beginning of each epoch, and rollout budget allocation is updated dynamically based on historical success statistics accumulated across iterations. Table 3 presents the complete hyperparameters and configuration details for our experiments.

F Detailed Description of Benchmarks

To comprehensively evaluate mathematical reasoning capabilities, we employ six widely-adopted benchmarks spanning competition-level challenges (AIME, AMC), curriculum-aligned assessments (MATH), and specialized STEM domains (Minerva, OlympiadBench). These benchmarks encompass diverse mathematical subfields and problem formats, enabling rigorous evaluation across multiple dimensions of reasoning proficiency. Table 4 summarizes the key characteristics of each benchmark.

G Efficiency Analysis

Figure 6 presents per-step training time across model scales. DynaMO maintains comparable efficiency to baselines on both 1.5B and 7B models, as the additional computations—Bernoulli variance estimation, water-level allocation, and token-level modulation—involve only lightweight arithmetic operations on existing logits and statistics, incurring negligible overhead relative to model inference.



(a) Qwen2.5-Math-1.5B

(b) Qwen2.5-Math-7B

Figure 6: Per-step training time on Qwen2.5-Math models. Bold lines: smoothed; light lines: raw measurements.

Table 3: Detailed training configuration for DynaMO experiments.

Category	Parameter	Value
	Training Hyperparameters	
Batch Configuration	Generation Batch Size	512
	Update Batch Size	32
	PPO Mini-batch Size	32
Rollout Allocation	Average Rollouts per Prompt	16
	Dynamic Range	[8, 24]
Optimization	Learning Rate (Actor)	1×10^{-6}
	Learning Rate (Critic)	1×10^{-5}
	Weight Decay	0.1 / 0.01
	Gradient Clipping	1.0
	Warmup Steps	10
	Warmup Style	constant
	GRPO-Specific Settings	
Clipping	Clip Ratio	0.2
Entropy	Entropy Coefficient	0
KL Penalty	KL Coefficient	0.0
Advantage Estimator	Type	GRPO
	Normalize by Std	True
	Gamma (Discount Factor)	1.0
	Lambda (GAE)	1.0
	Inference & Rollout	
Sampling Strategy	Temperature	1.0
	Top-p	1.0
	Top-k	-1 (disabled)
Sequence Lengths	Max Prompt Length	2048
	Max Response Length	8192
Rollout Engine	Backend	vLLM
	Tensor Parallel Size	2
	GPU Memory Utilization	0.8
	Max Num Sequences	1024
	Max Batched Tokens	10240

Benchmark	Core Description	Key Characteristics
AIME	American Invitational Mathematics Examination - high school competition	• 15 challenging problems per round • Integer answers (0-999) • Algebra, Geometry, Number Theory, Combinatorics • Multi-step reasoning required • Top AMC performers participate
AMC	American Mathematics Competitions - tiered assessment system	• Three tiers (AMC 8/10/12) for different grades • 25 multiple-choice problems per tier • Curriculum-aligned design • Comprehensive secondary mathematics coverage • Standardized difficulty progression
MATH-500	Curated subset from MATH dataset	• 500 problems from comprehensive collection • Seven domains (Algebra, Geometry, Number Theory, etc.) • Five difficulty levels • Formal mathematical reasoning • Step-by-step solution verification
Minerva Math	Technical STEM benchmark	• Undergraduate to graduate difficulty • Physics, Chemistry, Biology applications • Symbolic manipulation and formula derivation • Domain-specific knowledge integration • Quantitative problem-solving
OlympiadBench	Bilingual Olympiad-level benchmark	• Large-scale competition problem collection • Bilingual (English and Chinese) • Mathematics, Physics, Chemistry, Biology • Theorem-proving and open-ended problems • Multimodal inputs (text, diagrams, equations)

Table 4: Characteristics of mathematical reasoning benchmarks used in our evaluation. These benchmarks provide comprehensive coverage across difficulty levels (secondary to graduate), problem formats (multiple-choice, integer answers, open-ended), and mathematical domains (pure and applied mathematics), enabling thorough assessment of reasoning capabilities.