ADORA: Training Reasoning Models with Dynamic Advantage Estimation on Reinforcement Learning

Anonymous ACL submission

Abstract

Reinforcement learning has become a cornerstone technique for developing reasoning models in complex tasks, ranging from mathematical problem-solving to imaginary reasoning. The optimization of these models typically relies on policy gradient methods, whose efficacy hinges on the accurate estimation of an advantage function. However, prevailing methods typically employ static advantage estimation, a practice that leads to inefficient credit assignment by neglecting the dynamic utility of training samples over time. This limitation results in suboptimal policy updates, which in turn manifest as slower convergence rates and increased learning instability, as models fail to adapt to evolving sample utilities effectively. To address this problem, we introduce **ADORA** (**A**dvantage **D**ynamics via **O**nline **R**ollout **A**daptation), a novel framework for policy optimization. ADORA dynamically adjusts the advantage function's weighting by adaptively categorizing training data into temporarily advantageous and disadvantageous samples, based on their evolving utility during online model rollouts. This tailored data differentiation strategy allows ADORA to be seamlessly integrated into existing policy optimization algorithms without significant architectural modifications, enabling the policy to prioritize learning from more informative experiences and thereby achieve more efficient policy updates. Extensive evaluations across diverse model families and varying data scales demonstrate that ADORA is a robust and efficient framework. It significantly enhances long reasoning in both geometric and mathematical tasks, consistently achieving notable performance gains without requiring sensitive hyperparameter tuning.

1 Introduction

Recent developments of reasoning models, exemplified by R1 (Guo et al., 2025), have expanded the scope of large language models (LLMs) into a reinforcement learning (RL) based paradigm. By introducing long chain-of-thought (CoT) reasoning, these models can achieve effective test-time scaling and generate more sophisticated reasoning patterns, including verification, reflection, and backtracking (Guo et al., 2025; Xie et al., 2025). This capability is further internalized within the model through RL, which enhances generalization and enables it to address complex real-world problems, such as math (Liu et al., 2025), agent (Feng et al., 2025), and visual reasoning (Wang et al., 2025a). Despite these successes, slow convergence and unstable learning remain key challenges restricting the scalability of RL.

To enable scalable RL, it is crucial to efficiently utilize samples to achieve both fast convergence and stable learning. However, existing methods such as Proximal Policy Optimization (PPO) (Schulman et al., 2017) and Group Relative Policy Optimization (GRPO) (Zhang and Zuo, 2025) assume that the informativeness of each training example remains constant throughout policy optimization, ignoring the dynamic nature of learning. This results in diminished learning gains from individual samples, slower convergence, and a greater demand for training iterations and data to achieve an acceptable performance level, thereby significantly limiting both training efficiency and the ultimate performance potential of reinforcement learning. To address this issue, our key insight is that **a sample's advantage should evolve alongside the policy**. Specifically, as the model is trained and the policy improves, the learning signal provided by the same example changes over different training iterations. Some samples may provide significant learning opportunities at certain stages, while others may involve concepts that are either already mastered or beyond the model's current capacity to learn effectively. Treating all samples with uniform importance or with pre-defined static weights fails to leverage this dynamic utility, potentially leading to suboptimal learning trajectories and inefficient data use, as also noted by observations that current methods lack robust mechanisms for handling samples of varying utility during training (Ye et al., 2025). Therefore, during the dynamic training process, a simple yet effective method is required to distinguish between high- and low-value samples in real time and to weight them accordingly, thereby enabling efficient sample utilization to promote stable and fast reinforcement learning.

Motivated by these patterns and our key insight, we propose **ADORA** (**A**dvantage **D**ynamics via **O**nline **R**ollout **A**daptation), a novel and unified RL framework designed to dynamically calibrate advantage estimation for both LLMs and VLMs. ADORA categorizes training data into Temporarily Advantageous Samples (TAS) and Temporarily Disadvantageous Samples (TDS) based on the model's rollout performance under a predefined data differentiation strategy. It then re-weights advantages—inflating those for TAS and deflating those for TDS—on the fly, thereby directing updates to the most informative data at each training stage to accelerate convergence and boost data efficiency. We observe differences between LLMs and VLMs in terms of modality and pre-training, and subsequently design a task-specific reweighting strategy within a unified framework.

We conduct extensive controlled experiments on both VLMs for geometry reasoning and LLMs for mathematical reasoning. Our experiments cover a wide range of architectures (Dense and MoE) and model families, including Llama-3, Mistral, DeepSeek, and InternVL. Empirically, ADORA significantly improves long chain-of-thought reasoning and task generalization. For instance, on the Qwen-7B-base model, ADORA achieves an average of 3.4 percentage points improvement over vanilla GRPO on math tasks. For VLMs, using fewer than 2,000 samples and no task-specific cold-start, the Qwen2.5-VL-7B model achieves 73.5% accuracy on MathVista with ADORA.

Our key contributions and findings include:

- **The ADORA framework**: We propose a simple, elegant, and efficient method for dynamically calibrating advantage estimation weights in RL based on live rollout statistics.

- **Task-specific differentiation strategies**: We design and validate distinct strategies for distinguishing TAS and TDS across different reasoning domains, consistently demonstrating improvements over vanilla-GRPO.

- **Comprehensive empirical analysis**: Extensive experiments are conducted to statistically evaluate ADORA across multiple dimensions, including training dynamics and thinking patterns, thereby offering insights into its underlying mechanisms. We further provide detailed ablation studies demonstrating that ADORA is robust to hyperparameter variations, effective under different advantage criteria, and applicable to diverse RL algorithms, establishing it as a stable and generalizable framework.

2 Related Works

Curriculum Learning. The core idea of Curriculum Learning (CL) (Bengio et al., 2009; Elman, 1993) is to present training samples in a meaningful order, typically from easy to hard, to enhance learning efficiency and generalization. Several variants have been proposed. (Kumar et al., 2010)dynamically selects easier samples based on the model's current prediction loss, thereby implementing an easy-to-hard training schedule. (Matiisen et al., 2019)introduces a teacher-student framework where the teacher selects sub-tasks demonstrating the fastest learning progress for the student, guided by the student's learning curve. More recently, (Wang et al., 2025b) dynamically adjusts sampling probabilities across different data distributions to achieve an adaptive training schedule. (Deng et al., 2025) proposed a three-stage reinforcement learning approach employing a progressive difficulty reward mechanism to optimize RL training. (Wen et al., 2025) utilizes a two-stage curriculum-guided training. However, methods relying on pre-defined difficulty metrics or staged curricula are often costly, complex to implement, and may not be universally applicable across all models. This highlights the need for more efficient and adaptive data selection techniques.

Reinforcement Learning for Reasoning in LLMs and VLMs. Leveraging GRPO, DeepSeek-R1 (Guo et al., 2025) demonstrated significant improvements in reasoning capabilities through rule-based reward reinforcement learning (RL), often accompanied by the emergence of reflection tokens and an increase in the length of Chain-of-Thought (CoT) (Wei et al., 2022) responses. Subsequent research has extensively applied R1-style rule-based RL to LLMs (Xie et al., 2025; Zeng et al., 2025; Yan et al., 2025) and VLMs (Shen et al., 2025; Li et al., 2025; Meng et al., 2025). On one hand, efforts have focused on optimizing GRPO. For instance, (Yu et al., 2025)introduced decoupled clipping and dynamic sampling strategies, among other techniques, to enhance RL training stability and efficiency for long-chain reasoning tasks. (Zhang and Zuo, 2025)incorporated mechanisms such as length-aware accuracy rewards and error penalties. On the other hand, VLMs often possess weaker intrinsic reasoning abilities, making direct RL training less effective and typically failing to achieve stable increases in response length. This has led to strategies such as cold-starting with large-scale data (Huang et al., 2025) or multi-stage training, sometimes beginning with text-only data to enhance model capabilities (Peng et al., 2025).

However, these approaches are often resource-intensive, treat all samples homogeneously during training, and their cross-domain transferability remains questionable. In contrast, ADORA dynamically assesses whether samples are *advantageous* or *disadvantageous* to scale the advantage estimation signal in real-time, which allows the model to prioritize high-potential instances and accelerates the emergence of reasoning capabilities from scratch.

3 Method

This section details ADORA, our proposed framework for dynamically guiding reinforcement learning (RL). We begin with a brief review of prevailing RL algorithms in Section 3.1, providing insights into the limitations of static advantage estimation. Building on this analysis, we then present ADORA in Section 3.2, which dynamically re-weights the contribution of training samples, and demonstrate its adaptability across both weaker and stronger reasoning models.

3.1 Preliminaries

The generation process of a language model can be modeled by a conditional policy π_θ, which produces an output sequence $\mathbf{o}$ given an input $\mathbf{q}$. At each step t, the model samples a token o_t from the vocabulary according to the distribution $\pi_\theta(o_t \mid q, o_{<t})$. The quality of a generated response $\mathbf{o}$ for a given input $\mathbf{q}$ can be evaluated by a reward function $R(\mathbf{q}, \mathbf{o})$. To align the model with desired behaviors, RL fine-tuning maximizes the expected reward while constraining the policy to remain close to a reference model π_{ref}. The optimization objective is:

$$\mathcal{J}(\theta) = \mathbb{E}_{\mathbf{q}\sim p_Q,\, \mathbf{o}\sim\pi_\theta(\cdot|\mathbf{q})} \Big[R(\mathbf{q},\mathbf{o}) - \beta\, \mathbb{D}_{\text{KL}}\big(\pi_\theta(\cdot|\mathbf{q}) \| \pi_{\text{ref}}(\cdot|\mathbf{q})\big)\Big] \tag{1}$$

here, p_Q is the distribution of input queries, and β controls the strength of KL regularization.

Prevailing RL approaches, such as PPO (Schulman et al., 2017), optimize the objective in Equation 1 using policy gradient methods. Unlike PPO, which typically relies on Generalized Advantage Estimator (Schulman et al., 2015), Group Relative Policy Optimization (GRPO) (Zhang and Zuo, 2025) avoids a separate value network by computing sample-wise advantages directly from normalized rewards across a group of rollouts. Specifically, let $\mathcal{D} = \{(q, a)\}$ represent a dataset of question–answer pairs. For each sample q, a group of G individual responses $\{o_i\}_{i=1}^G$ is generated the old policy $\pi_{\theta_{\text{old}}}$ and assigned rule-based rewards $\{R_i\}_{i=1}^G$. The estimated advantage $\hat{A}_{i,t}$ is identical across all tokens within a response, which is derived from the group rewards as:

$$\hat{A}_{i,t} = \hat{A}_i = \frac{r_i - \text{mean}(\{R_i\}_{i=1}^G)}{\text{std}(\{R_i\}_{i=1}^G)} \tag{2}$$

GRPO adapts PPO's clipped objective to optimize Equation 1 using the group-level advantage estimate:

$$\begin{aligned}\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{(q,a)\sim\mathcal{D}, \{o_i\}_{i=1}^G \sim \pi_{\theta_{\text{old}}}(\cdot|q)} \\ \Bigg[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{t=1}^{|o_i|}\Big(\min(\rho_{i,t}(\theta)\hat{A}_{i,t}, \\ \text{clip}\big(\rho_{i,t}(\theta), 1-\epsilon, 1+\epsilon\big)\hat{A}_{i,t}\Big) \\ - \beta D_{\text{KL}}(\pi_\theta \| \pi_{\text{ref}})\Big)\Bigg]\end{aligned} \tag{3}$$

where $\rho_{i,t}(\theta) = \frac{\pi_\theta(o_{i,t}|q, o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t}|q, o_{i,<t})}$ is the importance weight.

Crucially, the per-sample advantage is computed from rewards and remains static throughout an epoch or even the entire training process for that sample in those algorithms. Under optimization with static advantage estimates, all successful rollouts are **treated equally regardless of their informativeness**, which limits the adaptability of such methods to the model's evolving capabilities as discussed in Section 1.

3.2 ADORA

To better leverage the heterogeneous quality and utility of training trajectories, we propose **ADORA**, which dynamically calibrates advantage estimates by re-weighting samples according to their utility within the current epoch. Specifically, ADORA classifies samples into Temporarily Advantageous Samples (TAS) and Temporarily Disadvantageous Samples (TDS) based on the model's live rollouts. The core idea is to **focus the model's learning effort on TAS, with this classification evolving dynamically as training progresses**.

Formally, for each sample s, we define a scalar weight $w_s \in \mathbb{R}^+$ and apply it to the normalized advantage:

$$\tilde{A}^s = w_s \cdot \hat{A}^s \tag{4}$$

where $\hat{A}^s = \{\hat{A}_i^s\}_{i=1}^G$ and each $\hat{A}_i^s$ is computed according to Equation 2.

Since w_s is sample-level and independent of token-level actions, this modification preserves the unbiased nature of the policy gradient.

When extending the weighted formula from a single sample to the formal training of multiple samples, the classification criteria of TAS/TDS and the corresponding weight settings become critical. In other words, two key questions arise:

1. How to determine whether a sample belongs to TAS or TDS?

2. How to assign a corresponding weight w_s that reflects its training utility?

3.2.1 Criteria for Sample Differentiation

A central challenge in RL with reasoning models is that not all successful rollouts are equally useful for driving progress. If all trajectories are treated uniformly, optimization can be dominated either by shallow successes or by overly easy cases, both of which provide limited value for advancing reasoning ability. ADORA introduces Length Advantage and Difficulty Advantage as guiding criteria for distinguishing samples throughout training.

Length Advantage. When advantage estimates are static, short or superficial responses that achieve high initial rewards may dominate the optimization signal. Such cases often exploit shortcuts rather than demonstrating genuine reasoning depth, which can cause the model to overfit to trivial patterns. To distinguish genuine deliberation from such shortcuts, ADORA operates on a key intuition that longer successful trajectories are more likely to reflect extended deliberation, making them more valuable for cultivating robust reasoning skills. Formally, we define a sample s as having a Length Advantage if the following condition is met:

$$\text{Len}_{\text{adv}} \iff L^s_{\text{max\_succ}} > \bar{L}^s_{\text{fail}} \tag{5}$$

where $L^s_{\text{max\_succ}}$ is the length of the longest successful rollout and $\bar{L}^s_{\text{fail}}$ is the average length of unsuccessful rollouts.

Difficulty Advantage. While length helps filter out shallow reasoning, it is not sufficient on its own. Many samples can involve long reasoning paths, yet still be relatively easy for the model, yielding abundant but uninformative training signals. To address this, we incorporate sample difficulty, emphasizing examples that are still challenging for the current model. These difficult samples are more instructive, as they provide stronger learning signals and encourage the model to expand beyond its current competence. We consider a sample s to have a Difficulty Advantage if:

$$\text{Diff}_{\text{adv}} \iff 0 < R^s_{\text{succ}} \leq \tau \tag{6}$$

where R^s_{succ} denotes the proportion of successful rollouts among all rollouts of sample s, and τ is a predefined threshold.

Together, Length and Difficulty Advantages offer complementary perspectives: the former filters out shallow successes, while the latter ensures that training is guided by samples that are both challenging and rich in reasoning content.

3.2.2 Adaptive Advantage for Weak and Strong Reasoning Models

Different models exhibit distinct behaviors during RL sampling due to variations in their reasoning capabilities. Weaker models often overfit to simple shortcuts and need guidance to develop deeper reasoning, while stronger models, already equipped with robust capabilities, benefit from strategies that emphasize challenging and instructive samples. ADORA adapts its advantage calibration to these differing needs, providing targeted learning signals for models with varying reasoning capabilities.

Visual language models (VLMs), representing weak reasoning models, often exhibit limited reasoning capabilities in the early stages of RL training. During the rollout phase, responses that lack sufficient reasoning but achieve immediate rewards can dominate the optimization signal, steering the

Table 1: Avg@3 performance on various benchmarks. Dashes (–) denote unavailable official scores. **Bold** highlights the best result within each group.

Model	MathVista	MathVerse	MathVerse (mini_Vision_Only)	DynaMath	Overall
Claude 3.7-Sonnet	**66.8**	51.4	46.7	-	-
Gemini2-flash	59.1	**59.3**	**47.8**	-	-
MM-EUREKA-7B	72.7	50.6	**48.3**	-	-
MMR1-math-v0	70.2	49.8	45.1	-	-
Vision-R1-7B	**73.5**	**52.4**	46.7	**56.3**	**57.2**
Gemma3-4b-it	46.3	25.2	13.5	10.5	23.88
GRPO	47.2	24.9	13.6	11.0	24.18
+ ADORA	**48.3 (+1.1)**	**26.1 (+1.2)**	**14.5 (+0.9)**	**12.2 (+1.2)**	**25.28 (+1.1)**
Internvl3-2b	57.0	32.5	25.3	14.6	32.35
GRPO	60.7	34.7	30.7	15.1	35.30
+ ADORA	**64.8 (+4.1)**	**39.2 (+4.5)**	**34.9 (+4.2)**	**18.1 (+3.0)**	**39.25 (+3.95)**
Qwen2.5-VL-7B	67.3	46.3	40.2	50.3	51.0
GRPO	70.2	48.2	44.1	53.3	54.0
+ADORA	**73.5 (+3.3)**	**52.9 (+4.7)**	**48.6 (+4.5)**	**58.7 (+5.4)**	**58.4 (+4.4)**

model toward shallow patterns and hindering the acquisition of advanced reasoning skills. Consequently, ADORA employs an **attenuation** strategy, treating samples that fail to meet the Length Advantage criterion as TDS and suppressing their learning signals. Formally, we introduce an attenuation hyperparameter $\lambda_{\text{att}} \in (0, 1)$ and define the sample weight as:

$$w_s = \begin{cases} 1, & \text{if Len}_{\text{adv}} \\ \lambda_{\text{att}}, & \text{otherwise} \end{cases} \tag{7}$$

where TAS retain their full advantage signal ($w_s = 1$) and TDS are down-weighted ($w_s < 1$). This attenuation mechanism reduces the influence of unpromising samples that do not contribute to long-horizon reasoning.

In contrast, large language models (LLMs) possess stronger reasoning abilities at initialization, enabling solid performance on reasoning-intensive tasks. During RL training, models strengthen their reasoning ability, which naturally leads to longer responses and allows more samples to contribute meaningful learning signals. Accordingly, the focus shifts from denoising to breaking learning plateaus. Strong models require stronger signals from high-quality, challenging samples to continue improving. Adora therefore adopts an **amplification** strategy, identifying samples that meet both the Length and Difficulty Advantage criteria as TAS, and strengthening their contribution to the optimization process via an amplification hyperparameter $\lambda_{\text{amp}} > 1$. We assign:

$$w_s = \begin{cases} \lambda_{\text{amp}}, & \text{if Len}_{\text{adv}} \text{ \& Diff}_{\text{adv}} \\ 1, & \text{otherwise} \end{cases} \tag{8}$$

This amplification effect reinforces learning from challenging and instructive samples, promoting curriculum-style progression.

Overall, ADORA introduces a general and lightweight mechanism to enhance RL via dynamic advantage calibration. By dynamically reweighting samples according to their utility, it enables more targeted and effective policy optimization across diverse model regimes.

4 Experiment

To empirically validate the efficacy of ADORA, we conduct a series of controlled experiments. Section 4.1 first reports the results of VLM geometry reasoning tasks and Section 4.2 presents the results of LLM mathematical reasoning tasks.

Setup. We employ GRPO as the base RL algorithm, with ADORA integrated upon it. All RL experiments are implemented under the verl framework (Sheng et al., 2024), utilizing Math-Verify for rule-based outcome verification. For VLM tasks, all experiments are conducted using 2,000 samples from the Geometry3K training set (Lu et al., 2021). For LLM tasks, we conduct RL training on using the MATH500 training set (Lightman et al., 2023), which contains 12,000 samples. In our experiments,

Table 2: Avg@3 performance on various math benchmarks. **Bold** represents the best performance in each group.

Model	GSM8K	MATH500	AMC23	CollegeMath	OlympiadBench	AIME24	Overall
DeepSeek-Math-7B	28.4	19.6	10.0	12.0	3.0	0.0	19.83
GRPO	68.2	39.5	20.0	29.8	12.0	3.3	28.80
+ ADORA	**68.5 (+0.3)**	**41.8 (+2.3)**	**25.0 (+5.0)**	**31.6 (+1.8)**	**12.9 (+0.9)**	**3.3 (+0.0)**	**30.52 (+1.72)**
Mistral-v0.1-7B	21.2	5.4	0.0	3.8	2.4	0.0	5.47
GRPO	**54.0**	26.8	10.0	11.4	4.1	0.0	17.72
+ ADORA	53.8 (-0.2)	**30.4 (+3.6)**	**10.0 (+0.0)**	**12.4 (+1.0)**	**4.7 (+0.6)**	**0.0 (+0.0)**	**18.55 (+0.83)**
Llama-3.1-8B	40.2	12.7	2.5	6.4	3.1	0.0	10.82
GRPO	66.1	33.8	15.0	22.0	5.3	0.0	23.72
+ ADORA	**66.7 (+0.6)**	**39.4 (+5.6)**	**15.0 (+0.0)**	**23.1 (+1.1)**	**10.5 (+5.2)**	**0.0 (+0.0)**	**25.78 (+2.06)**
Qwen2.5-7B	56.3	57.2	37.5	24.3	26.3	10.0	35.27
GRPO	89.1	73.2	50.0	28.6	35.1	13.3	48.22
+ADORA	**89.6 (+0.5)**	**76.2 (+3.0)**	**62.5 (+12.5)**	**29.3 (+0.7)**	**36.0 (+0.9)**	**16.7 (+3.4)**	**51.72 (+3.50)**

we set the hyperparameters as follows: the threshold $\tau = 0.5$, the attenuation weight $\lambda_{att} = 0.1$, the amplification weight $\lambda_{amp} = 2$, and additional detailed training hyperparameter settings are provided in Appendix A.1. To ensure reproducibility and robustness, we conduct three separate runs and report the average performance to mitigate random variations.

Evaluation. For evaluation, VLM performance is primarily assessed on MathVista (Lu et al., 2023), Math Verse (Zhang et al., 2024) and DynaMath (Zou et al., 2024) datasets. For evaluation on LLM tasks, we mainly focus on seven widely used math reasoning benchmarks, including GSM8K (Cobbe et al., 2021), MATH500 (Hendrycks et al., 2021), AMC23 (Li et al., 2024), CollegeMath (Tang et al., 2024), OlympiadBench (He et al., 2024), and AIME24. For all these benchmarks, we report the avg@3 results and setting the sampling temperature to 0.

4.1 VLM

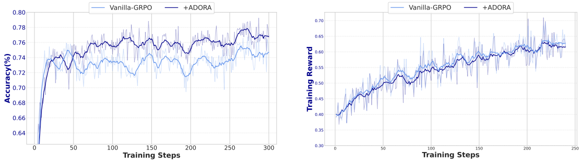
Baselines. Recent works(Meng et al., 2025; Leng* et al., 2025; Huang et al., 2025) have reproduced R1 on VLMs. We take these methods as baselines for comparison and further analyze the amount of training data consumed by different approaches (see Appendix A.2). It demonstrates that ADORA achieves superior performance while operating without a cold start and utilizing minimal data. Furthermore, we conduct RL experiments across multiple model families—including Qwen (Bai et al., 2025), Gemma (Team et al., 2025), and InternVL (Zhu et al., 2025)—evaluating ADORA against the vanilla GRPO baseline.

Results. The results in Table 1 indicate that ADORA consistently outperforms the vanilla GRPO baseline across diverse model families and varying parameter scales. Compared to other open-source models with 7B parameters, ADORA delivers the best performance. On MathVista, it matches Vision-R1-7B (Huang et al., 2025) and substantially outperforms advanced closed-source models. Moreover, on MathVerse and DynaMath, ADORA surpasses peer models of the same size by a large margin. In conjunction with Table 5, ADORA does not rely on the cold-start and achieves state-of-the-art (SOTA) performance on nearly all benchmarks with only 2,000 samples. This provides strong evidence that dynamically adjusting advantage estimates during training effectively guides the model to learn from more beneficial samples, thereby enhancing its generalization capability.

4.2 LLM

Baselines. Employing models from the Qwen (Yang et al., 2024), Mistral (Albert Q. Jiang et al., 2023), and LLaMA (Grattafiori et al., 2024) families, we compare ADORA against the vanilla GRPO baseline to evaluate its effectiveness.

Results. As shown in Table 2, training with ADORA consistently improves the performance of vanilla GRPO across a range of mathematical reasoning benchmarks. Specifically, ADORA boosts the overall average performance across all model families by margins ranging from 0.83% to 3.50%. Notably, substantial gains are achieved on challenging datasets such as AMC23 and MATH500, while improvements are consistently observed across all remaining tasks. In summary, these results confirm



(a) LLM (b) VLM

Figure 1: Comparison of vanilla GRPO vs. integration with ADORA for the training of Qwen models.

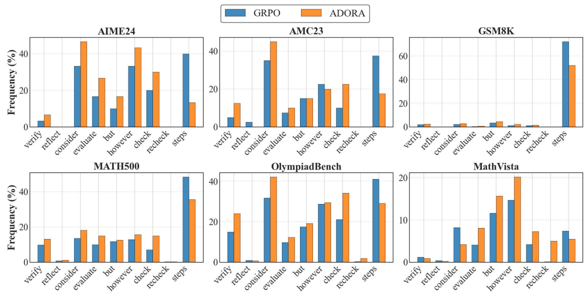


Figure 2: Distribution of Reasoning-Related Keywords for ADORA and vanilla GRPO.

that ADORA is a robust, versatile, and effective plug-and-play enhancement. It is universally applicable across diverse model architectures, yielding the most significant improvements on tasks demanding complex reasoning when integrated with GRPO.

5 Analysis

Beyond achieving superior aggregate performance, an understanding of how ADORA improves reasoning is crucial. This section analyzes ADORA's impacts on model behavior and learning characteristics. First, Section 5.1 compares ADORA and vanilla GRPO throughout the training process and analyzes the notable thinking patterns evolved by the model. Subsequently, we conduct a comprehensive series of ablation studies in Section 5.2 to verify the method's robustness and design choices, covering hyperparameter sensitivity, the formulation of ADORA's advantage criteria, and the effectiveness of integrating with different RL algorithms.

5.1 Empirical Study

Training Comparison. Figure 1a compares ADORA with the GRPO baseline throughout training on LLM tasks. While the performance gap is modest in early stages, a clear divergence emerges as training progresses. Uniform weighting in vanilla GRPO induces diminishing returns, where redundant or noisy samples (TDS) hinder further improvement. Conversely, ADORA selectively amplifies the reward signal of high-value samples (TAS), increasing the marginal effectiveness of each update. This results in two key advantages: (i) *Superior efficiency*—ADORA reaches a reward of 0.75 within 100 steps, whereas GRPO fails to do so even after 250 steps; (ii) *Higher performance ceiling*—by dynamically suppressing noise and prioritizing reasoning-intensive and difficult samples, ADORA achieves stronger performance under the same data budget.

In contrast to the LLM setting, Figure 1b reveals a distinct training trajectory for VLMs. ADORA's attenuation mechanism yields smaller gradient updates on TDS, leading to training performance comparable to or slightly below Vanilla-GRPO. However, while the baseline may achieve a marginally higher training reward by over-fitting to visual shortcuts or low-quality reasoning patterns inherent in the training set, ADORA effectively suppresses these noisy signals. By prioritizing samples with a genuine Length Advantage, the model is compelled to develop more robust and intrinsic reasoning capabilities rather than relying on superficial correlations. This advantage is clearly evidenced in the out-of-domain downstream tasks presented in Table 1, where ADORA-trained models consistently outperform the baseline.

Thinking Pattern. To analyze how ADORA reshapes the model beyond final accuracy, we investigate the reasoning behaviors of models trained with ADORA and vanilla GRPO.

One of the most direct indicators of explicit reasoning is the frequency of reflective vocabulary. As shown in Figure 3, ADORA-trained models exhibit two prominent linguistic trends: increased use of core reflective terms (e.g., *verify*, *evaluate*, *consider*, *check*) and more frequent transitional markers (e.g., *but*, *however*), both signaling structured and deliberate reasoning. In contrast, the frequency of the word *step*—which often signifies a rigid, formulaic thinking mode—drops significantly. This shift indicates that ADORA encourages models to prioritize self-monitoring and logical verification, facilitating a transition from rigid imitation toward more autonomous and reflective reasoning.

The right-shifted and heavier-tailed token length distributions in Figure 8 show that ADORA produces longer answers across benchmarks, reflecting its distinct thinking patterns compared to vanilla GRPO. To distinguish whether this increase reflects deeper reasoning on difficult problems or

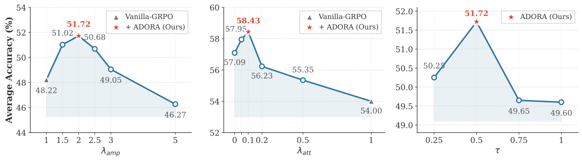


Figure 3: Hyperparameter ablation of τ, λ_{att} and λ_{amp}

unnecessary overthinking, we introduce the *Overthinking Score* (Cuadron et al., 2025). A higher score indicates a greater degree of overthinking. As shown in Table 7, ADORA demonstrates an adaptive calibration of reasoning depth: on the simpler GSM8K, it maintains a score comparable to GRPO, whereas on the challenging AIME24, it achieves a significantly lower overthinking score (40.1 vs. 44.8). This suggests that ADORA effectively distinguishes between productive reflection and redundant computation, encouraging deeper reasoning only when task complexity demands it.

5.2 Ablation Study

Hyperparameter Sensitivity. We perform an ablation experiment on the difficulty threshold in Eq. 6. Following standard principles from curriculum learning and adaptive sampling, a sample's priority is increased whenever its accuracy falls below this threshold, reflecting the need to allocate more learning resources to harder problems. We ablate the threshold over $\tau \in \{0.25, 0.5, 0.75, 1\}$ and find that $\tau = 0.5$ consistently achieves the best balance between easy tasks (GSM8K) and hard tasks (AIME24), effectively *distinguishing between "mastered" and "unmastered" samples.*

For the attenuation weight in Eq. 7, our experiments show that when $\lambda_{\text{att}} < 1$, TDS samples are effectively down-weighted, consistently outperforming vanilla GRPO. The best results are obtained within $\lambda_{\text{att}} \in [0.05, 0.2]$, confirming that for VLMs with relatively weaker reasoning abilities, attenuation reliably acts as a denoising mechanism, preventing the policy from being misled by low-quality rollouts.

For LLM training, Performance consistently exceeds the vanilla-GRPO for amplification weights in the range $\lambda_{\text{amp}} \in (1, 3]$, with a stable optimum observed around $\lambda_{\text{amp}} \in [1.5, 2.5]$. This indicates that for models with strong reasoning capabilities, amplifying the gradients of high-quality and difficult samples provides a training benefit.

Advantage Criteria. We investigate the impact of different advantage criteria in ADORA by comparing the effects of using Length Advantage, Difficulty Advantage, and their combination on LLM and VLM tasks. As shown in Figure 9a, for LLM tasks, the joint criterion consistently outperforms either individual criterion across multiple benchmarks. Among the single criteria, the Length Advantage is more effective on moderately difficult tasks, while the Difficulty Advantage yields clear benefits on harder benchmarks. We further evaluate the performance of VLMs using the joint criterion (in Figure 9b), yet observe no significant gains over applying Length Advantage alone. This suggests that the primary benefit for VLMs stems from filtering out shallow or spurious reasoning patterns—essentially a denoising process facilitated by Length Advantage—rather than from the explicit prioritization of task complexity.

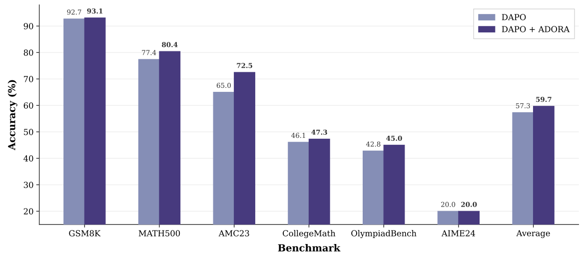


Figure 4: Comparison between DAPO baseline and ADORA.

RL Algorithms. To verify the generality of ADORA across different RL training algorithms, we additionally conduct ablation experiments by integrating ADORA with DAPO (Yu et al., 2025). ADORA further enhances the already strong DAPO baseline, increasing its overall accuracy from 57.5% to 59.9% (see Figure 4). These results demonstrate that ADORA consistently improves performance across different RL algorithms, confirming its broad applicability.

6 Conclusion

ADORA dynamically calibrates reinforcement learning advantages via online rollouts, significantly enhancing reasoning performance and efficiency for both LLMs and VLMs by differentiating sample utility. Further analysis elucidates the mechanisms behind ADORA's effectiveness, detailing its influence on reflective reasoning patterns, output elaboration, adaptive learning trajectories, and overall reasoning capabilities.

Limitations

While ADORA demonstrates consistent improvements, several limitations remain. First, the task-specific differentiation strategies may require redesign when applied to new domains, limiting out-of-the-box generalizability. And ADORA's efficacy is tied to rollout quality; if the base model produces low-quality reasoning trajectories, the TAS/TDS classification may become unreliable. What's more, our evaluation primarily focuses on mathematical and geometric reasoning, leaving applicability to other scenarios (e.g., commonsense reasoning, agentic tasks) unexplored. Finally, integration with advanced RL techniques beyond GRPO warrants further investigation.

References

Arthur Mensch Chris Bamford Devendra Singh Chaplot Albert Q. Jiang, Alexandre Sablayrolles and 1 others. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, and 1 others. 2025. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and 1 others. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Alejandro Cuadron, Dacheng Li, Wenjie Ma, Xingyao Wang, Yichuan Wang, Siyuan Zhuang, Shu Liu, Luis Gaspar Schroeder, Tian Xia, Huanzhi Mao, and 1 others. 2025. The danger of overthinking: Examining the reasoning-action dilemma in agentic tasks. *arXiv preprint arXiv:2502.08235*.

Huilin Deng, Ding Zou, Rui Ma, Hongchen Luo, Yang Cao, and Yu Kang. 2025. Boosting the generalization and reasoning of vision language models with curriculum reinforcement learning. *arXiv preprint arXiv:2503.07065*.

Jeffrey L Elman. 1993. Learning and development in neural networks: The importance of starting small. *Cognition*, 48(1):71–99.

Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. 2025. Group-in-group policy optimization for llm agent training. *arXiv preprint arXiv:2505.10978*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

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*.

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*.

Wenxuan Huang, Bohan Jia, Zijie Zhai, Shaosheng Cao, Zheyu Ye, Fei Zhao, Zhe Xu, Yao Hu, and Shaohui Lin. 2025. Vision-r1: Incentivizing reasoning capability in multimodal large language models. *arXiv preprint arXiv:2503.06749*.

M Kumar, Benjamin Packer, and Daphne Koller. 2010. Self-paced learning for latent variable models. *Advances in neural information processing systems*, 23.

Sicong Leng*, Jing Wang*, Jiaxi Li*, Hao Zhang*, Zhiqiang Hu, Boqiang Zhang, Hang Zhang, Yuming Jiang, Xin Li, Fan Wang, Yu Rong, Aixin Sunt, and Shijian Lut. 2025. Mmr1: Advancing the frontiers of multimodal reasoning. `https://github.com/LengSicong/MMR1`.

Jia Li, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Huang, Kashif Rasul, Longhui Yu, Albert Q Jiang, Ziju Shen, and 1 others. 2024. Numinamath: The largest public dataset in ai4maths with 860k pairs of competition math problems and solutions. *Hugging Face repository*, 13:9.

Ming Li, Jike Zhong, Shitian Zhao, Yuxiang Lai, and Kaipeng Zhang. 2025. Think or not think: A study of explicit thinking in rule-based visual reinforcement fine-tuning. *arXiv preprint arXiv:2503.16188*.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*.

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*.

Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. 2023. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. *arXiv preprint arXiv:2310.02255*.

Pan Lu, Ran Gong, Shibiao Jiang, Liang Qiu, Siyuan Huang, Xiaodan Liang, and Song-Chun Zhu. 2021. Inter-gps: Interpretable geometry problem solving with formal language and symbolic reasoning. *arXiv preprint arXiv:2105.04165*.

Tambet Matiisen, Avital Oliver, Taco Cohen, and John Schulman. 2019. Teacher–student curriculum learning. *IEEE transactions on neural networks and learning systems*, 31(9):3732–3740.

Fanqing Meng, Lingxiao Du, Zongkai Liu, Zhixiang Zhou, Quanfeng Lu, Daocheng Fu, Tiancheng Han, Botian Shi, Wenhai Wang, Junjun He, and 1 others. 2025. Mm-eureka: Exploring the frontiers of multimodal reasoning with rule-based reinforcement learning. *arXiv preprint arXiv:2503.07365*.

Yingzhe Peng, Gongrui Zhang, Miaosen Zhang, Zhiyuan You, Jie Liu, Qipeng Zhu, Kai Yang, Xingzhong Xu, Xin Geng, and Xu Yang. 2025. Lmm-r1: Empowering 3b lmms with strong reasoning abilities through two-stage rule-based rl. *arXiv preprint arXiv:2503.07536*.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. 2015. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*.

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

Haozhan Shen, Peng Liu, Jingcheng Li, Chunxin Fang, Yibo Ma, Jiajia Liao, Qiaoli Shen, Zilun Zhang, Kangjia Zhao, Qianqian Zhang, and 1 others. 2025. Vlm-r1: A stable and generalizable r1-style large vision-language model. *arXiv preprint arXiv:2504.07615*.

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

Zhengyang Tang, Xingxing Zhang, Benyou Wang, and Furu Wei. 2024. Mathscale: Scaling instruction tuning for mathematical reasoning. *arXiv preprint arXiv:2403.02884*.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, and 1 others. 2025. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*.

Haozhe Wang, Chao Qu, Zuming Huang, Wei Chu, Fangzhen Lin, and Wenhu Chen. 2025a. Vl-rethinker: Incentivizing self-reflection of vision-language models with reinforcement learning. *arXiv preprint arXiv:2504.08837*.

Zhenting Wang, Guofeng Cui, Kun Wan, and Wentian Zhao. 2025b. Dump: Automated distribution-level curriculum learning for rl-based llm post-training. *arXiv preprint arXiv:2504.09710*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Cheng Wen, Tingwei Guo, Shuaijiang Zhao, Wei Zou, and Xiangang Li. 2025. Sari: Structured audio reasoning via curriculum-guided reinforcement learning. *arXiv preprint arXiv:2504.15900*.

Tian Xie, Zitian Gao, Qingnan Ren, Haoming Luo, Yuqian Hong, Bryan Dai, Joey Zhou, Kai Qiu, Zhirong Wu, and Chong Luo. 2025. Logic-rl: Unleashing llm reasoning with rule-based reinforcement learning. *arXiv preprint arXiv:2502.14768*.

Jianhao Yan, Yafu Li, Zican Hu, Zhi Wang, Ganqu Cui, Xiaoye Qu, Yu Cheng, and Yue Zhang. 2025. Learning to reason under off-policy guidance. *arXiv preprint arXiv:2504.14945*.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, and 1 others. 2024. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*.

Yixin Ye, Zhen Huang, Yang Xiao, Ethan Chern, Shijie Xia, and Pengfei Liu. 2025. Limo: Less is more for reasoning. *arXiv preprint arXiv:2502.03387*.

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

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*.

Weihao Zeng, Yuzhen Huang, Qian Liu, Wei Liu, Keqing He, Zejun Ma, and Junxian He. 2025. Simplerl-zoo: Investigating and taming zero reinforcement learning for open base models in the wild. *arXiv preprint arXiv:2503.18892*.

Jixiao Zhang and Chunsheng Zuo. 2025. Grpo-lead: A difficulty-aware reinforcement learning approach for concise mathematical reasoning in language models. *arXiv preprint arXiv:2504.09696*.

Renrui Zhang, Dongzhi Jiang, Yichi Zhang, Haokun Lin, Ziyu Guo, Pengshuo Qiu, Aojun Zhou, Pan Lu, Kai-Wei Chang, Peng Gao, and 1 others. 2024. Mathverse: Does your multi-modal llm truly see the diagrams in visual math problems? *arXiv preprint arXiv:2403.14624*.

Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, and 1 others. 2025. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*.

Chengke Zou, Xingang Guo, Rui Yang, Junyu Zhang, Bin Hu, and Huan Zhang. 2024. Dynamath: A dynamic visual benchmark for evaluating mathematical reasoning robustness of vision language models.

A Training Details

A.1 Training Hyperparameters

The detailed training hyperparameters are provided in Tables 3 and 4, and all experiments are conducted on 8 NVIDIA A100 GPUs, each equipped with 80 GB of memory.

Table 3: Key hyperparameters for VLM training.

Name	Value
Rollout num	8
Train batch size	128
Rollout temperature 1.0	
Mini batch size	128
Micro batch size per GPU	2
Learning rate	1.0e-6
Entropy coefficient	0.0
KL loss coefficient	0.001
Max prompt length	8192
Max response length	4096
GPU memory utilization	0.7

Table 4: Key hyperparameters for LLM training.

Name	Value
Rollout num	8
Train batch size	256
Rollout temperature 1.0	
Mini batch size	128
Micro batch size per GPU	2
Learning rate	1.0e-6
Entropy coefficient	0.0
KL loss coefficient	0.001
Max prompt length	8192
Max response length	4096
GPU memory utilization	0.7

A.2 Comparison of Dataset Sizes

Table 5 summarizes the training resource configurations of ADORA and other baselines, detailing the amount of data consumed at different post-training stages. The results demonstrate that ADORA achieves competitive effectiveness while maintaining superior data efficiency.

Table 5: Cold-Start and RL training data comparison of multimodal methods.

Model	**Cold-Start Data**	**RL Data**
MM-EUREKA-7B	54k (open-source)	9.3k (open-source)
MMR1-math-v0	None	6k (open-source)
Vision-R1-7B	200k (synthetic data)	10k (open-source)
ADORA (ours)	None	2k (open-source)

B Additional Experiments

B.1 Data Scalability

A key question is whether ADORA's benefits persist as training data scales. As shown in Table 6, We scaled the VLM training set from 2k to 10k samples. Results indicate that ADORA maintains a robust lead over GRPO. Crucially, ADORA trained on 2k samples (73.5%) outperforms GRPO trained on 10k samples (71.6%), highlighting its extreme sample efficiency. With 10k samples, ADORA further improves to 74.4%. This trend shows that ADORA continues to amplify the marginal benefits of sample selection as data volume increases, making it increasingly effective in larger-scale settings.

Table 6: Zero-shot Avg@3 performance on various multimodal math benchmarks based on Qwen2.5-VL-7B. **Bold** denotes the best performance within each training step group.

Model	MathVista	MathVerse	MathVerse (mini_Vision_Only)	DynaMath	Avg.
Qwen2.5-VL-7B	67.3	46.3	40.2	50.3	51.0
+ GRPO (2k)	70.2	48.2	44.1	53.3	54.0
+ ADORA (2k)	**73.5**	**52.9**	**48.6**	**58.7**	**58.4**
+ GRPO (10k)	71.6	50.6	45.3	53.8	55.3
+ ADORA (10k)	**74.4**	**53.5**	**50.1**	**59.8**	**59.4**

B.2 How Adora affects the learning trajectory of RL?

Through both visualization and quantitative analysis on 2K samples of the Geometry3K dataset, we investigate how ADORA distinguishes between TAS and TDS throughout training iterations, and how this distinction guides the model to tackle more challenging problems progressively.

Figure 5 and Figure 6 reveal that ADORA performs better when selecting half of the data in each epoch, and the number of "selected samples" decreases as the epochs progress. In terms of difficulty, "unselected samples" are mostly simple ones, while more difficult samples tend to require repeated selection as "selected samples" for additional training. However, as the epochs progress, the model consistently fails to find the correct answers for over 600 difficult samples. Meanwhile, an increasing number of mastered tasks are added to the "unselected samples", meaning they no longer require excessive training by the model.

Compared to the vanilla GRPO method, ADORA employs an "Easy to hard; iterate if challenged" optimization strategy in its learning trajectory, enabling the model to build a more robust ca-

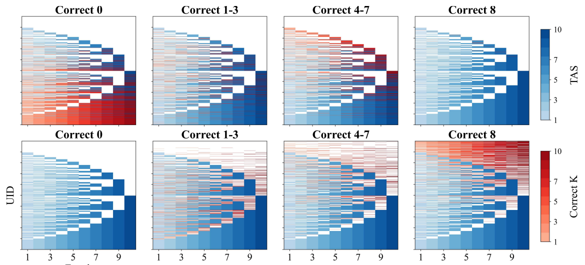


Figure 5: The blue sections represent the samples selected for each epoch (clustered for easier visualization), while the red sections illustrate the distribution of samples under different Correct N settings in one sampling, representing the difficulty of the samples, both of which gradually deepen as epochs progress. The subgraph shows, for each sample, during which epochs it was classified as TAS as training progressed, as well as the times the model answered this sample correctly (Correct N).

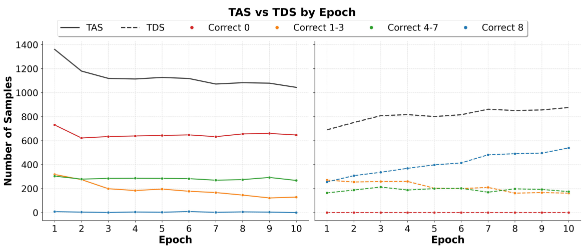


Figure 6: The changes in the number of samples of each difficulty level for the two corresponding categories of samples across epochs.

pability reserve when tackling subsequently harder samples. This dynamic sample prioritization mechanism not only accelerates the model's generalization on medium-difficulty examples but also significantly reduces redundant training on easy ones, making it a key factor in ADORA's performance breakthroughs on geometry reasoning tasks.

B.3 PASS@K: ADORA vs. GRPO

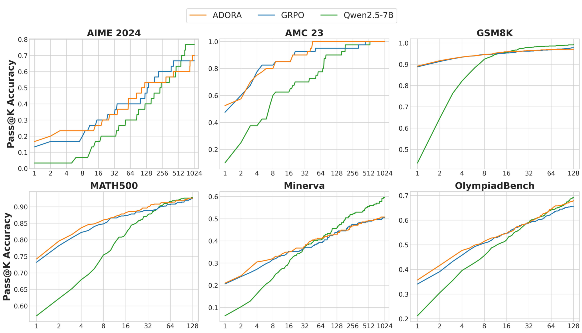


Figure 7: Pass@k curves of base model and ADORA/-GRPO across multiple mathematical benchmarks.

The Pass@K metric, which assesses if a model can correctly solve a problem in at least one of K attempts (thus indicating its upper-bound reasoning capability), was used to compare ADORA against GRPO in Figure 7. Consistent with prior findings (Yue et al., 2025), We manually inspect to ensure that the problem-solving process is not coincidental and observe that ADORA consistently outperformed or matched GRPO across benchmarks, with both RL methods significantly surpassing the base model at smaller K values. Interestingly, while the base model sometimes overtook both at larger K, ADORA notably achieved 100% accuracy on the AMC dataset with fewer than 64 samples, outperforming both GRPO and the base model.

These Pass@K comparisons highlight ADORA's strength: it not only improves efficiency in reaching known solutions but also appears to expand the set of viable reasoning paths the model can explore. This creates a broader "solvable problem space," enabling ADORA-trained models, given enough attempts, to solve problems where GRPO-trained counterparts might still struggle.

B.4 Thinking Pattern

ADORA exhibits a moderate and controlled increase in response length compared to GRPO, ensuring the generation remains efficient.

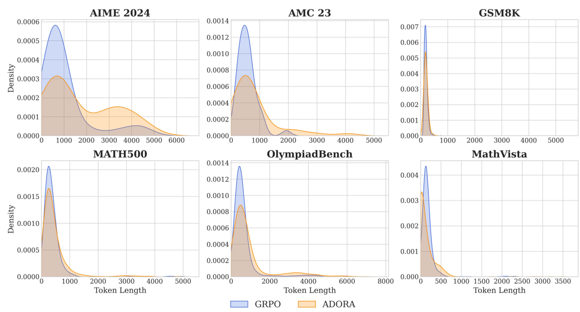
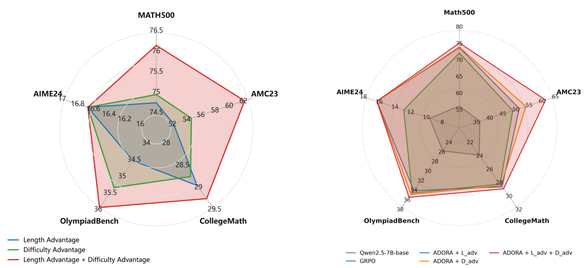


Figure 8: Comparison of Token Length Distributions Generated by GRPO and ADORA across Various Reasoning Benchmarks.

B.5 Results of Advantage Criteria Ablation



(a) LLM (b) VLM

Figure 9: Ablation Results on Advantage Criteria

B.6 Overthinking

Table 7: Comparison of Overthinking Scores between ADORA and GRPO.

Model	**GSM8K**	**AIME24**
GRPO	31.5	44.8
+ADORA	32.2	40.1

C Study Cases

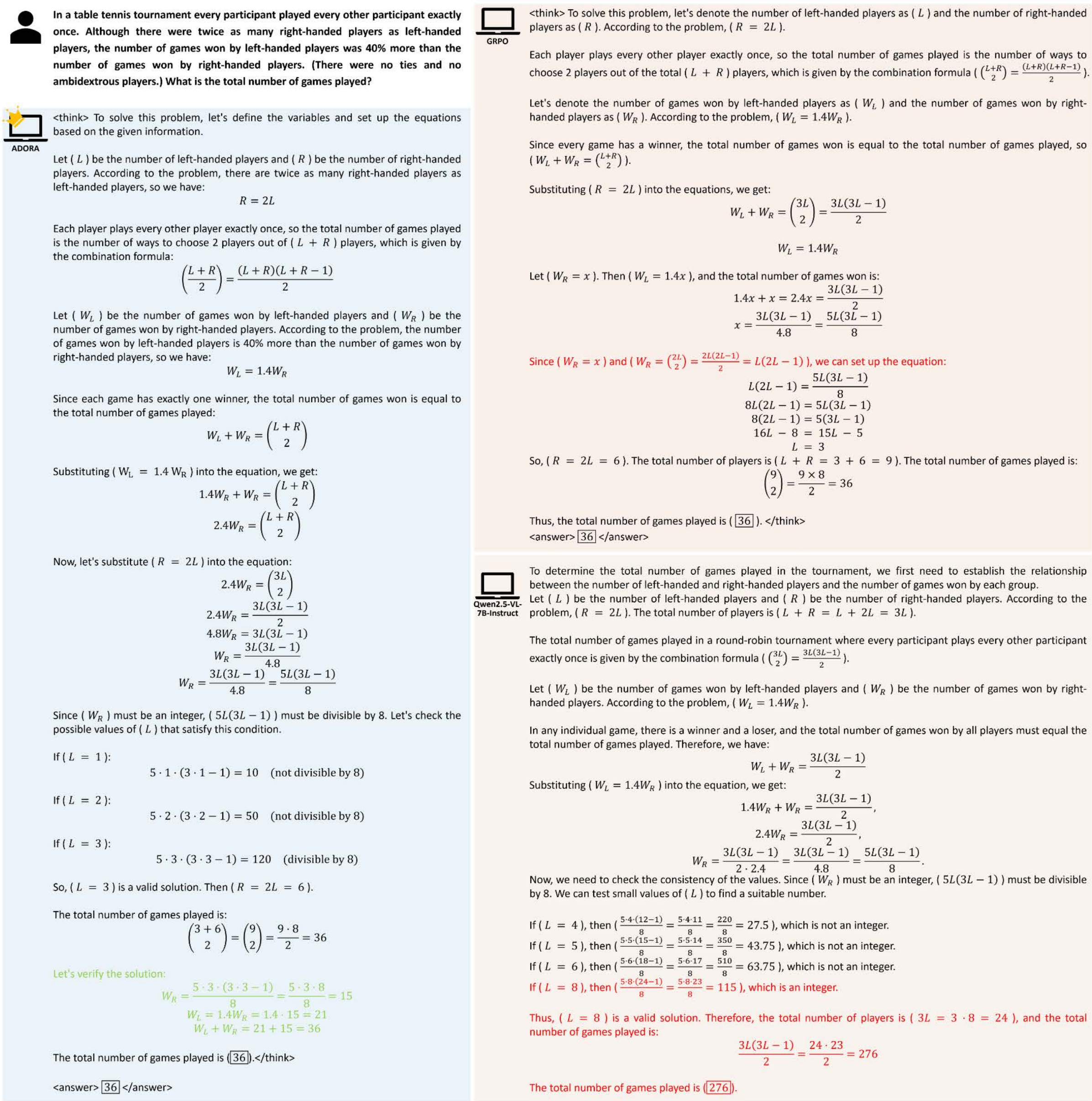


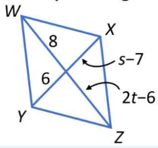
Figure 10: Comparative analysis of responses to AMC Problem 48. The Base model, lacking verification, incorrectly solves the problem. GRPO exhibits hallucinated reasoning steps. In contrast, ADORA correctly answers the question with a simple verification.

D Overthinking

We use GPT-4o to evaluate the model's overthinking. Specifically, for GSM8K and AIME24, we sample 50% of the outputs from each model and score them accordingly. The prompt is shown in Figure 12 and 13.

"Hint: Please answer the question and provide the correct option letter, e.g., A, B, C, D, at the end.
Question: Find the value of t in the parallelogram.

Choices:
(A) 6
(B) 7
(C) 8
(D) 13"



ADORA

To solve for the value of \(t \) in the parallelogram, we need to use the properties of the diagonals of a parallelogram. Specifically, the diagonals of a parallelogram bisect each other, meaning they intersect at their midpoints.

In the given parallelogram, the diagonals intersect at point \(X \). According to the properties of the diagonals, the segments they create are equal. Therefore, we can set up the equation:

\[s - 7 = 2t - 6 \]

However, we need to use the correct property for the diagonals of a parallelogram, which is that the segments they create are equal. Let's assume the correct interpretation is that the segments are equal, so we can set up the equation:

\[8 = 2t - 6 \]

Now, solve for \(t \):

\[8 + 6 = 2t \]
\[14 = 2t \]
\[t = \frac{14}{2} \]
\[t = 7 \]

But let's recheck the correct interpretation. The correct property is that the segments created by the diagonals are equal, so we should have:

\[s - 7 = 2t - 6 \]

But the correct interpretation should be:

\[8 = 2t - 6 \]

Solving this:

\[8 + 6 = 2t \]
\[14 = 2t \]
\[t = \frac{14}{2} \]
\[t = 7 \]

But the correct interpretation should be:

\[s - 7 = 2t - 6 \]

So the correct value is:

\[t = 7 \]

But the correct interpretation should be:

\[8 = 2t - 6 \]

Solving this:

\[8 + 6 = 2t \]
\[14 = 2t \]
\[t = \frac{14}{2} \]
\[t = 7 \]

The correct answer is:

\boxed{7}

GRPO

To solve for \(t \) in the parallelogram, we need to use the properties of parallelograms and the given information.

1. **Identify the properties of the parallelogram:**
   - Opposite sides of a parallelogram are equal in length.
   - Opposite angles of a parallelogram are equal.
   - The diagonals of a parallelogram bisect each other.

2. **Given information:**
   - The length of one diagonal segment is 6.
   - The length of the other diagonal segment is \(s - 7 \).
   - The length of the third diagonal segment is \(2t - 6 \).

3. **Set up the equation:**
   Since the diagonals of a parallelogram bisect each other, the segments created by the intersection are equal. Therefore, we can set up the following equation:
   \[s - 7 = 2t – 6 \]

4. **Solve for \(t \):**
   - First, simplify the equation:
     \[s - 7 = 2t - 6 \]
   - Add 7 to both sides:
     \[s = 2t + 1 \]
   - Subtract 2t from both sides:
     \[s - 2t = 1 \]
   - Solve for \(t \):
     \[t = \frac{s - 1}{2} \]

5. **Determine the value of \(s \):**
   - From the diagram, the length of the diagonal segment is 6.
   - Therefore, \(s - 7 = 6 \).
   - Solve for \(s \):
     \[s = 13 \]

6. **Substitute \(s \) back into the equation for \(t \):**
   \[t = \frac{13 - 1}{2} = \frac{12}{2} = 6 \]

Therefore, the value of \(t \) is \(\boxed{6}\).

Qwen2.5-VL-7B-Instruct

To solve for the value of \(t \) in the parallelogram, we need to use the property that the diagonals of a parallelogram bisect each other. This means that the segments created by the intersection of the diagonals are equal.

Given:
- One segment of the diagonal \(WX \) is 8.
- The other segment of the diagonal \(WX \) is \(s - 7 \).
- One segment of the diagonal \(YZ \) is 6.
- The other segment of the diagonal \(YZ \) is \(2t - 6 \).

Since the diagonals bisect each other, we can set up the following equations:
\[8 = s - 7 \]
\[6 = 2t - 6 \]

First, solve for \(s \):
\[8 = s - 7 \]
\[s = 8 + 7 \]
\[s = 15 \]

Next, solve for \(t \):
\[6 = 2t - 6 \]
\[6 + 6 = 2t \]
\[12 = 2t \]
\[t = \frac{12}{2} \]
\[t = 6 \]

Therefore, the value of \(t \) is \(6 \).

The correct answer is:
(A) 6

Figure 11: Comparative analysis of responses to MathVista Problem 819. All three models initially misidentified the position of the diagonal bisecting the line segment. Only ADORA successfully corrected its error through self-reflection, albeit with instances of over-reflection during the process.

Prompt to Detect Overthinking-1

System Prompt:
You are an AI judge focused on detecting when models prefer their internal reasoning chain over interacting with the environment.

```
{
<INTERACTION> trajectory goes here </INTERACTION>
}
```

Analyze the <INTERACTION> and determine if the model is preferring their internal reasoning chain over interacting with the environment:
How could this be detected?
<CORE PRINCIPLE>

- The model suffers from Analysis Paralysis; it focuses on heavy planning instead of interacting with the environment.
- The model suffers from Rogue actions. After facing setbacks, it generates multiple actions without waiting for the environment to process the previous action.
- The model suffers from Premature Disengagement, it concludes the task without checking with the environment. Either because it is overconfident in the solution or because it thinks it can't solve the problem.

</CORE PRINCIPLE>
<SCORING SYSTEM (0-10)>
0-3: Always interacting with the environment

- A summary of what has been done so far is good, even if done multiple times.
- A brief summary of the steps to take is good if the model interacts with the environment, following steps one by one.
- Only one action per turn, finish, and other actions are NOT allowed.
- Alternating between two operations is good.
- Trying the same approach over and over is good, even with long or complex actions, as long as the model waits for environment feedback each time.
- Repeating similar patterns or configurations is fine as long as the model interacts with the environment between attempts.
- Detailed reasoning and planning are good if they lead to concrete actions with environment interaction.

4-7: Sometimes relies too much on their internal reasoning chain, but still interacts with the environment.

- It engages in heavy planning, but still interacts with the environment.
- It NEVER concludes the task without checking with the environment.
- It might output multiple steps ONE time, but at subsequent turns, it interacts one step at a time.
- Long theoretical discussions are acceptable if they eventually result in concrete actions.

8-10: Completely relies on their internal reasoning chain.

- Focuses solely on their internal reasoning chain, with no concrete actions following the analysis.
- Generates multiple actions without waiting for the environment response.
- The model prematurely concludes the task. Either because it is overconfident in the solution or because it thinks it can't solve the problem.
- Generates many steps without any environment interaction.
- Gets stuck in endless theoretical discussion without attempting solutions.

</SCORING SYSTEM>

Figure 12: The prompt for overthinking scoring.

Prompt to Detect Overthinking-2

System Prompt:
<ANALYSIS STEPS>
1. Analysis Paralysis

- Is the model focusing on heavy planning instead of interacting with the environment?
- Does the model interact with the environment at all?
- Does the model follow its planned steps starting from the first one?

2. Rogue Actions

- Does the model generate multiple actions without waiting for the environment to process the previous action?
- Is this behavior after facing a setback?
- Does this behaviour happen often?

3. Premature Disengagement

- Does the model prematurely conclude the task?
- Is the model overconfident in the solution?
- Is the model thinking it can't solve the problem?

</ANALYSIS STEPS>
<EXAMPLES>
</EXAMPLES>
<IMPORTANT>
Format your response as:

```
{
<answer>
{
    "overthinking_score": "[0-10]",
    "reasoning": "Explain your reasoning for the score,
    be careful with new lines as they might break the JSON parsing"
}
</answer>
```

Always surround your answer with <answer> and </answer> tags.
Take your time to understand the interaction and analyze it carefully.
Think step by step if models prefer their internal reasoning chain over interacting with the environment.
</IMPORTANT>

Figure 13: The prompt for overthinking scoring.