

000 001 002 003 004 005 006 007 008 009 010 011 012 013 014 015 016 017 018 019 020 021 022 023 024 025 026 027 028 029 030 031 032 033 034 035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050 051 052 053 ADORA: TRAINING REASONING MODELS WITH DYNAMIC ADVANTAGE ESTIMATION ON REINFORCEMENT LEARNING

Anonymous authors

Paper under double-blind review

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** (Advantage Dynamics 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 on various tasks demonstrate that ADORA significantly enhances long reasoning in both geometric and mathematical tasks across large vision–language models and large language models, achieving notable performance gains.

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 (Guo et al., 2025; Xie et al., 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.

054 Treating all samples with uniform importance or with pre-defined static weights fails to leverage this
 055 dynamic utility, potentially leading to suboptimal learning trajectories and inefficient data use, as also
 056 noted by observations that current methods lack robust mechanisms for handling samples of varying
 057 utility during training (Ye et al., 2025). Therefore, during the dynamic training process, a simple yet
 058 effective method is required to distinguish between high- and low-value samples in real time and
 059 to weight them accordingly, thereby enabling efficient sample utilization to promote stable and fast
 060 reinforcement learning.

061 Motivated by these patterns and our key insight, we propose **ADORA** (Advantage Dynamics via
 062 Online Rollout Adaptation), a novel and unified RL framework designed to dynamically calibrate
 063 advantage estimation for both LLMs and VLMs. ADORA categorizes training data into Temporarily
 064 Advantageous Samples (TAS) and Temporarily Disadvantageous Samples (TDS) based on the
 065 model’s rollout performance under a predefined data differentiation strategy. It then re-weights
 066 advantages—inflate those for TAS and deflate those for TDS—on the fly, thereby directing
 067 updates to the most informative data at each training stage to accelerate convergence and boost data
 068 efficiency. We observe differences between LLMs and VLMs in terms of modality and pre-training,
 069 and subsequently design a task-specific reweighting strategy within a unified framework.

070 We conduct extensive controlled experiments on both VLMs for geometry reasoning and LLMs
 071 for mathematical reasoning. Empirically, ADORA significantly improves long chain-of-thought
 072 reasoning and task generalization. For instance, on the Qwen-7B-base model, ADORA achieves an
 073 average of 3.4 percentage points improvement over vanilla GRPO on math tasks. For VLMs, using
 074 fewer than 2,000 samples and no cold-start, the Qwen2.5-VL-7B-instruct model achieves 73.5%
 075 accuracy on MathVista with ADORA.

076 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 GRPO.
- **Comprehensive empirical analysis:** Extensive experiments are conducted to statistically evaluate ADORA across multiple dimensions, including training dynamics, reflective frequency, overthinking behavior, and generalization ability, thereby offering insights into its underlying mechanisms.

088 2 RELATED WORKS

090 **Curriculum Learning.** The core idea of Curriculum Learning (CL) (Bengio et al., 2009; Elman,
 091 1993) is to present training samples in a meaningful order, typically from easy to hard, to enhance
 092 learning efficiency and generalization. Several variants have been proposed. (Kumar et al., 2010)dy-
 093 namically selects easier samples based on the model’s current prediction loss, thereby implementing
 094 an easy-to-hard training schedule. (Matiisen et al., 2019)introduces a teacher-student framework
 095 where the teacher selects sub-tasks demonstrating the fastest learning progress for the student, guided
 096 by the student’s learning curve. More recently, (Wang et al., 2025b) dynamically adjusts sampling
 097 probabilities across different data distributions to achieve an adaptive training schedule. (Deng et al.,
 098 2025) proposed a three-stage reinforcement learning approach employing a progressive difficulty
 099 reward mechanism to optimize RL training. (Wen et al., 2025) utilizes a two-stage curriculum-
 100 guided training. However, methods relying on pre-defined difficulty metrics or staged curricula are
 101 often costly, complex to implement, and may not be universally applicable across all models. This
 102 highlights the need for more efficient and adaptive data selection techniques.

103 **Reinforcement Learning for Reasoning in LLMs and VLMs.** Leveraging GRPO, DeepSeek-R1
 104 (Guo et al., 2025) demonstrated significant improvements in reasoning capabilities through rule-based
 105 reward reinforcement learning (RL), often accompanied by the emergence of reflection tokens and an
 106 increase in the length of Chain-of-Thought (CoT) (Wei et al., 2022) responses. Subsequent research
 107 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 & 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. This approach requires no cold-start, leverages the entire dataset effectively, and has demonstrated steady improvements in performance for both LLMs and VLMs.

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 $\pi_\theta(o_t | q, o_{<t})$. 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:

$$\mathcal{J}(\theta) = \mathbb{E}_{\mathbf{q} \sim p_Q, \mathbf{o} \sim \pi_\theta(\cdot | \mathbf{q})} \left[R(\mathbf{q}, \mathbf{o}) - \beta \mathbb{D}_{KL}(\pi_\theta(\cdot | \mathbf{q}) \parallel \pi_{ref}(\cdot | \mathbf{q})) \right] \quad (1)$$

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

Group Relative Policy Optimization (GRPO). 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), GRPO 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 $\pi_{\theta_{old}}$ 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:

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

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

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

with token-level importance weights $\rho_{i,t}(\theta) = \frac{\pi_\theta(o_{i,t} | q, o_{i,<t})}{\pi_{\theta_{old}}(o_{i,t} | q, o_{i,<t})}$.

Crucially, the per-sample advantage is computed from rewards and remains static throughout an epoch or even the entire training process for that sample. 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.

162 3.2 ADORA
163

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

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

$$172 \quad \tilde{A}^s = w_s \cdot \hat{A}^s \quad (4)$$

173 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
174 and independent of token-level actions, this modification preserves the unbiasedness of the policy
175 gradient.

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

- 180 1. How to determine whether a sample belongs to TAS or TDS?
- 181 2. How to assign a corresponding weight w_s that reflects its training utility?

183 3.2.1 CRITERIA FOR SAMPLE DIFFERENTIATION
184

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

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

$$197 \quad \text{Length}_{\text{adv}} \iff L_{\text{max,succ}}^s > \bar{L}_{\text{fail}}^s \quad (5)$$

198 where $L_{\text{max,succ}}^s$ is the length of the longest successful rollout and \bar{L}_{fail}^s is the average length of
199 unsuccessful rollouts.

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

$$207 \quad \text{Difficulty}_{\text{adv}} \iff R_{\text{succ}}^s \leq \tau \quad (6)$$

208 where R_{succ}^s denotes the proportion of successful rollouts among all rollouts of sample s , and τ is a
209 predefined threshold, set to 0.5. Together, Length and Difficulty Advantages offer complementary
210 perspectives: the former filters out shallow successes, while the latter ensures that training is guided
211 by samples that are both challenging and rich in reasoning content.

213 3.2.2 ADAPTIVE ADVANTAGE FOR WEAK AND STRONG REASONING MODELS
214

215 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

216 reasoning, while stronger models, already equipped with robust capabilities, benefit from strategies
 217 that emphasize challenging and instructive samples. ADORA adapts its advantage calibration to these
 218 differing needs, providing targeted learning signals for models with varying reasoning capabilities.
 219

220 Visual language models (VLMs), representing weak reasoning models, often exhibit limited reasoning
 221 capabilities in the early stages of RL training. During the rollout phase, responses that lack sufficient
 222 reasoning but achieve immediate rewards can dominate the optimization signal, steering the model
 223 toward shallow patterns and hindering the acquisition of advanced reasoning skills. Consequently,
 224 ADORA employs an **attenuation** strategy, treating samples that fail to meet the Length Advantage
 225 criterion as TDS and suppressing their learning signals. Formally, the sample weight is defined as:
 226

$$w_s = \begin{cases} 1, & \text{if Length}_{\text{adv}} \\ 0.1, & \text{otherwise} \end{cases} \quad (7)$$

227 where TAS retain their full advantage signal ($w_s = 1$) and TDS are down-weighted ($w_s = 0.1$).
 228 This attenuation mechanism reduces the influence of unpromising samples that do not contribute to
 229 long-horizon reasoning.
 230

231 In contrast, large language models (LLMs) possess stronger reasoning abilities at initialization,
 232 enabling solid performance on reasoning-intensive tasks. During RL training, models strengthen their
 233 reasoning ability, which naturally leads to longer responses and allows more samples to contribute
 234 meaningful learning signals. Accordingly, the focus shifts from mitigating shallow rollouts to amplifying
 235 the gains derived from difficult and instructive samples. ADORA therefore adopts an **amplification**
 236 strategy, identifying samples that meet both the Length and Difficulty Advantage criteria as TAS, and
 237 strengthening their contribution to the optimization process. We assign:
 238

$$w_s = \begin{cases} 2, & \text{if Length}_{\text{adv}} \& \text{Difficulty}_{\text{adv}} \\ 1, & \text{otherwise} \end{cases} \quad (8)$$

239 where TAS receive an increased weight ($w_s = 2$) and TDS maintain their original weight ($w_s = 1$).
 240 This amplification effect reinforces learning from challenging and instructive samples, promoting
 241 curriculum-style progression.
 242

243 Overall, ADORA introduces a general and lightweight mechanism to enhance RL via dynamic
 244 advantage calibration. By dynamically re-weighting samples according to their utility, it enables
 245 more targeted and effective policy optimization across diverse model regimes.
 246

247 4 EXPERIMENT

248 To empirically validate the efficacy of ADORA, we conduct a series of controlled experiments.
 249 Section 4.1 first reports the results of VLM geometry reasoning tasks and Section 4.2 presents the
 250 results of LLM mathematical reasoning tasks.
 251

252 **Setup.** For VLM tasks, all experiments are conducted on Qwen2.5-VL-7B-Instruct using 2,000
 253 samples from the Geometry3K training set (Lu et al., 2021). For LLM tasks, we conduct RL training
 254 directly on Qwen2.5-7B using the MATH500 training set (Lightman et al., 2023), which contains
 255 12,000 samples. All RL experiments are implemented under the verl framework (Sheng et al., 2024).
 256 We employ Math-Verify for rule-based outcome verification and MATH500 (Lightman et al., 2023)
 257 as the test set. For all models, we conduct three separate runs and report the average performance to
 258 mitigate random variations. Detailed training hyperparameter settings are provided in Appendix A.1.
 259

260 **Evaluation.** For evaluation, VLM performance is primarily assessed on MathVista (Lu et al., 2023),
 261 Math Verse (Zhang et al., 2024) and DynaMath (Zou et al., 2024) datasets. MathVista contains
 262 44.7% In-Domain (ID) geometric tasks and 55.3% Out-of-Domain (OOD) non-geometric samples.
 263 For evaluation on LLM tasks, we mainly focus on seven widely used math reasoning benchmarks,
 264 including GSM8K (Cobbe et al., 2021), MATH500 (Hendrycks et al., 2021), AMC23 (Li et al.,
 265 2024), CollegeMath (Tang et al., 2024), OlympiadBench (He et al., 2024), and AIME24. For all
 266 these benchmarks, we report the pass@1 rate, setting the sampling temperature to 0 and repeating the
 267 evaluation three times to obtain the average result.
 268

270 4.1 VLM
271

272 **Baselines.** Recent works(Meng et al., 2025; Leng* et al., 2025; Huang et al., 2025) have reproduced
273 R1 on VLMs. We take these methods as baselines for comparison and further analyze the amount of
274 training data consumed by different approaches (see Appendix A.2). It demonstrates that ADORA
275 achieves superior performance while operating without a cold start and utilizing minimal data.

276 Table 1: Zero-shot pass@1 performance on various benchmarks based on Qwen2.5-VL-7B-Ins.
277 Dashes (–) denote unavailable official scores. **Bold** denotes the top-performing closed/open-source
278 model.
279

280 281 282 283 284 285 286 287 288 289 290	Model	MathVista			MathVerse	MathVerse (mini_Vision_Only)	DynaMath	Overall
		ID	OOD	Avg.				
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		72.3	68.5	70.2	49.8	45.1	–	–
Vision-R1-7B		81.9	66.8	73.5	52.4	46.7	56.3	57.2
Qwen2.5-VL-7B-Ins.		69.6	65.5	67.3	46.3	40.2	50.3	51.0
GRPO		71.6	69.1	70.2	48.2	44.1	53.3	54.0
+ADORA		76.1	71.4	73.5	52.9	48.6	58.7	58.4

291
292
293 **Results.** The results in Table 1 indicate ADORA’s significant improvements over the baseline
294 GRPO on all metrics. Specifically, ADORA achieves 73.5% on MathVista, matching Vision-R1-
295 7B (Huang et al., 2025) and considerably outperforming Claude 3.7-Sonnet and Gemini2-flash,
296 alongside stronger OOD capabilities. On MathVerse and DynaMath, ADORA outperforms other
297 open-source models and achieves performance comparable to advanced closed-source models with
298 7B parameters. In conjunction with Table 6, ADORA does not rely on the cold-start and achieves
299 state-of-the-art (SOTA) performance on nearly all benchmarks with only 2,000 samples. This provides
300 strong evidence that dynamically adjusting advantage estimates during training effectively guides the
301 model to learn from more beneficial samples, thereby enhancing its generalization capability.

302 4.2 LLM
303

304 Table 2: Zero-shot pass@1 performance on various math benchmarks based on Qwen2.5-7B. Bold
305 represents the best performance.
306

307 308 309 310 311 312 313	Model	GSM8K	MATH500	AMC23	CollegeMath	OlympiadBench	AIME24	Overall
Qwen2.5-7B		56.3	57.2	37.5	24.3	26.3	10.0	35.3
GRPO		89.1	73.2	50.0	28.6	35.1	13.3	48.2
+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)	51.7 (+3.5)
DAPO		92.7	77.4	65	46.1	42.8	20.0	57.3
+ADORA		93.1 (+0.4)	80.4 (+3.0)	72.5 (+7.5)	47.3 (+1.2)	45.0 (+2.2)	20.0 (+0.0)	59.7 (+2.4)

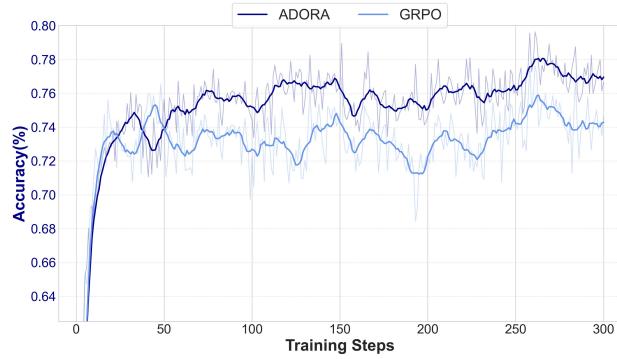
314
315
316 **Results.** As shown in Table 2, ADORA consistently improves the performance of existing RL
317 algorithms across a range of mathematical reasoning benchmarks. For GRPO, ADORA boosts the
318 overall average by 3.5%, with particularly notable gains on challenging benchmarks. Similarly,
319 ADORA enhances the already strong DAPO baseline, raising its overall accuracy from 57.5% to
320 59.9%.

321 Notably, substantial improvements are achieved on AMC23 and MATH500, while the remaining
322 tasks also benefit to varying extents. In summary, these results confirm that ADORA is a versatile
323 and effective plug-and-play enhancement, yielding the greatest improvements on tasks that demand
complex and challenging reasoning when integrated with both GRPO and DAPO.

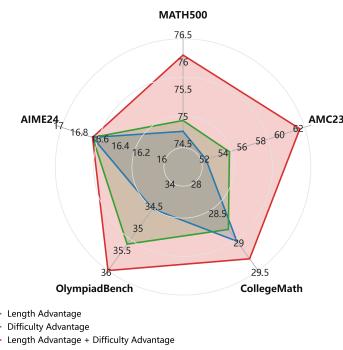
324 5 ANALYSIS

326 Beyond achieving superior aggregate performance, an understanding of how ADORA improves
 327 reasoning is crucial. This section analyzes ADORA’s impacts on model behavior and learning
 328 dynamics relative to the GRPO baseline. First, Section 5.1 compares ADORA and GRPO throughout
 329 training and conducts an ablation study on ADORA’s advantage criteria. Sections 5.2 and Section 5.3
 330 then analyze two notable patterns of ADORA: more frequent reflection and reduced overthinking.
 331 Finally, Section 5.4 explores the impact of ADORA on the trajectory of RL training.

333 5.1 EMPIRICAL STUDY



347 Figure 1: **Comparisons of GRPO vs ADORA.** We use
 348 accuracy to evaluate model performance during training.



347 Figure 2: Ablation study on
 348 ADORA’s advantage criteria.

351 **Training Comparison.** Figure 1 presents the performance comparison between ADORA and
 352 GRPO across the entire training process for LLM tasks. In the early stages of training (0–100 steps),
 353 the performance gap between ADORA and GRPO is not substantial. As training progresses and
 354 model capabilities gradually improve, the uniform weighting of all samples in GRPO can slow or even
 355 stagnate performance. In contrast, ADORA’s differentiated treatment of TAS and TDS strengthens
 356 the reward signal for high-value samples while attenuating it for low-value ones. This design enables
 357 ADORA to achieve substantially higher performance than the baseline and to **enhance the marginal**
 358 **benefits of training.** Once the model reaches a certain capability level, the gains derived from the
 359 same sample under ADORA significantly exceed those under GRPO.

360 This yields two key benefits. First, the model achieves faster convergence: for instance, ADORA
 361 reaches an average reward of 0.75 within 100 steps, whereas GRPO fails to achieve this level
 362 even after 250 steps. Second, with the same amount of training data, ADORA achieves higher
 363 scores by dynamically balancing different samples to fully exploit their potential, leading to a
 364 higher performance ceiling. As marginal gains diminish in the later stages of training, the model’s
 365 improvement rate slows; nevertheless, ADORA consistently outperforms the baseline.

366 **Ablation Study.** We investigate the impact of different advantage criteria in ADORA by comparing
 367 the effects of using Length Advantage, Difficulty Advantage, and their combination on LLM tasks.
 368 As shown in Figure 2, the joint criterion consistently outperforms either single criterion across
 369 multiple benchmarks. Among the single criteria, the Length Advantage tends to perform better on
 370 moderately difficult tasks, while the Difficulty Advantage shows clear benefits on harder benchmarks.
 371 However, on the highly challenging AIME24 task, the three criteria show no significant difference,
 372 yet all outperform vanilla GRPO. These results demonstrate the effectiveness of ADORA’s advantage
 373 criteria and show that their combination yields more robust performance across tasks.

375 5.2 REFLECTION FREQUENCY

376 One of the most direct indicators of explicit reasoning is the **frequency of reflective vocabulary usage**.
 377 Accordingly, we delve into the frequency of reflective vocabulary, providing insights into the cognitive

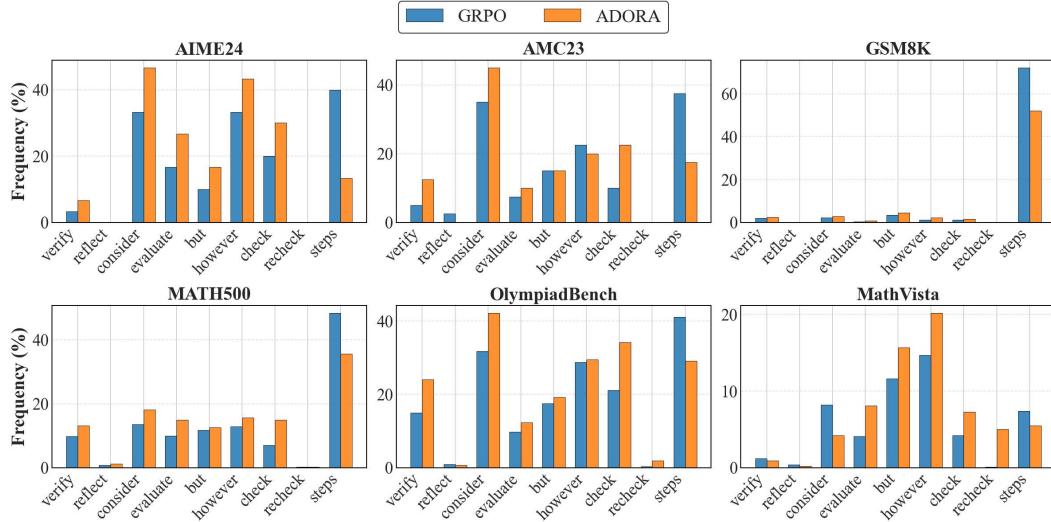


Figure 3: Distribution of Reasoning-Related Keywords for GRPO and ADORA across Various Reasoning Benchmarks.

behaviors fostered by ADORA compared to the baseline GRPO method across various mathematical benchmarks and quantify the tendency of models to engage in self-monitoring, verification, and structured thinking during problem-solving.

As illustrated in Figure 3, two major trends are observed: **increased use of core reflective terms** and **more structured and transitional language**. Words that directly indicate verification, evaluation, and deliberate reasoning—such as “verify”, “evaluate”, “consider”, “reflect”, and “check” appear more frequently in the outputs of models trained with ADORA across most benchmarks. For instance, the use of “verify” is markedly higher on the AIME24 benchmark, while “evaluate” shows similar trends on AMC23 and MATH500. Moreover, ADORA increases the frequency of terms that signal structured reasoning, such as “but” and “however,” across several benchmarks (e.g., MATH500, OlympiadBench). Notably, compared with GRPO, the frequency of the word “steps” drops significantly in ADORA. Unlike reflective and structural terms, “steps” represents a rigid mode of thinking that offers little benefit for answering questions. Overall, the reduced frequency of this word signifies a shift in the ADORA model’s reasoning, moving from rigid imitation of training data toward autonomous reflection and verification.

Although the presence of such words does not directly reflect model strength, it signals a certain capacity for explicit inference, which increases the likelihood of generating structured and logical answers to difficult questions. Compared to vanilla GRPO, the ADORA framework more effectively encourages the model to develop a reasoning style characterized by more frequent self-reflection, verification, and structured thinking.

5.3 OVERTHINKING ISSUES

While longer responses can foster more refined and structured reflection, they may also introduce unnecessary overthinking. The right-shifted and heavier-tailed token length distributions in Figure 7 show that ADORA produces longer answers across benchmarks, highlighting the need to assess whether this tendency results in overthinking. To investigate this phenomenon, we introduce an “Overthinking Score” (Cuadron et al., 2025) as a qualitative measure of the model’s tendency to overthink, strongly correlated with both response accuracy and length. A higher score indicates a deeper degree of the model’s overthinking.

Table 3 shows that, compared to the GRPO baseline, ADORA keeps overthinking within a reasonable range. On the simple reasoning task GSM8K, the strong performance of the GRPO baseline creates a

Table 3: Comparison of Overthinking Scores between ADORA and GRPO.

Model	GSM8K	AIME24
GRPO	31.5	44.8
+ADORA	32.2	40.1

ceiling effect that limits ADORA’s performance gains. As a result, even though responses generated by ADORA are only slightly longer than those of GRPO, they still lead to a marginally higher overthinking score. In contrast, on the more challenging benchmark AIME24, ADORA introduces a modest increase in reasoning length while promoting more deliberate reasoning patterns, resulting in substantial performance gains. Consequently, ADORA exhibits a substantially lower degree of overthinking compared to the GRPO baseline. These observations suggest that ADORA adaptively calibrates reasoning depth without inducing overthinking.

5.4 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.

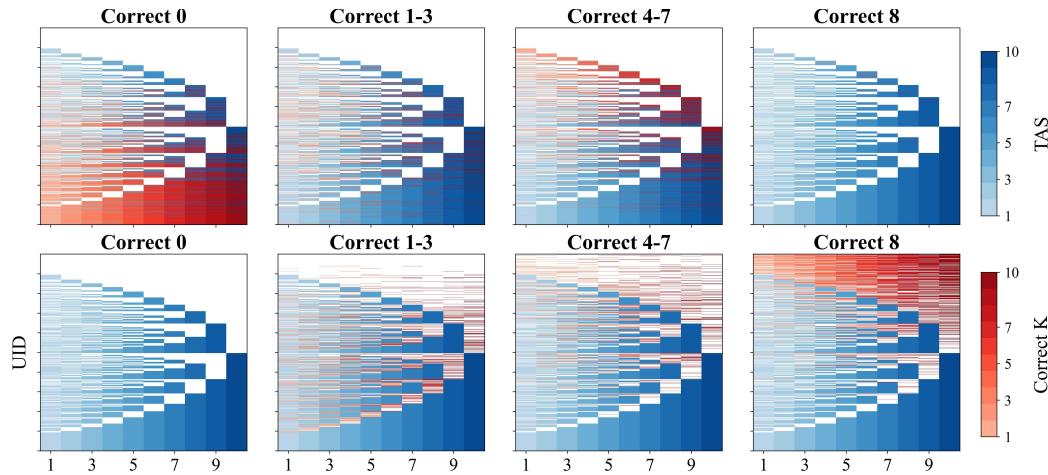


Figure 4: The blue sections represent the samples selected for each epoch (clustered for easier visualization), while the red sections illustrate the distribution of samples under different Correct N settings in one sampling, representing the difficulty of the samples, both of which gradually deepen 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 4 and Figure 8 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 capability 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.

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.

486 ETHICS STATEMENT
487488 We propose a new reinforcement learning framework that leverages open-source datasets in mathematical
489 and geometric reasoning for training. We do not anticipate any inherent negative societal
490 impacts arising from this work.
491492 REPRODUCIBILITY STATEMENT
493494 We strive to ensure the reproducibility of our work. To this end, we use open-source datasets and
495 report average results across multiple experimental runs. For comparative studies, we access open-
496 source models via Hugging Face (<https://huggingface.co/>) and closed-source models through their
497 public APIs, adhering strictly to their terms and conditions. The code and data necessary to reproduce
498 our experiments will be released on GitHub.
499500 REFERENCES
501502 Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In
503 *Proceedings of the 26th annual international conference on machine learning*, pp. 41–48, 2009.
504505 Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser,
506 Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve
507 math word problems. *arXiv preprint arXiv:2110.14168*, 2021.508 Alejandro Cuadron, Dacheng Li, Wenjie Ma, Xingyao Wang, Yichuan Wang, Siyuan Zhuang, Shu
509 Liu, Luis Gaspar Schroeder, Tian Xia, Huanzhi Mao, et al. The danger of overthinking: Examining
510 the reasoning-action dilemma in agentic tasks. *arXiv preprint arXiv:2502.08235*, 2025.511 Huilin Deng, Ding Zou, Rui Ma, Hongchen Luo, Yang Cao, and Yu Kang. Boosting the generalization
512 and reasoning of vision language models with curriculum reinforcement learning. *arXiv preprint
513 arXiv:2503.07065*, 2025.514 Jeffrey L Elman. Learning and development in neural networks: The importance of starting small.
515 *Cognition*, 48(1):71–99, 1993.
516517 Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. Group-in-group policy optimization for llm
518 agent training. *arXiv preprint arXiv:2505.10978*, 2025.519 Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu,
520 Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms
521 via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.522 Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu,
523 Xu Han, Yujie Huang, Yuxiang Zhang, et al. Olympiadbench: A challenging benchmark for
524 promoting agi with olympiad-level bilingual multimodal scientific problems. *arXiv preprint
525 arXiv:2402.14008*, 2024.526 Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song,
527 and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv
528 preprint arXiv:2103.03874*, 2021.529 Wenxuan Huang, Bohan Jia, Zijie Zhai, Shaosheng Cao, Zheyu Ye, Fei Zhao, Zhe Xu, Yao Hu, and
530 Shaohui Lin. Vision-r1: Incentivizing reasoning capability in multimodal large language models.
531 *arXiv preprint arXiv:2503.06749*, 2025.532 M Kumar, Benjamin Packer, and Daphne Koller. Self-paced learning for latent variable models.
533 *Advances in neural information processing systems*, 23, 2010.
534535 Sicong Leng*, Jing Wang*, Jiaxi Li*, Hao Zhang*, Zhiqiang Hu, Boqiang Zhang, Hang Zhang,
536 Yuming Jiang, Xin Li, Fan Wang, Yu Rong, Aixin Sunt, and Shijian Lut. Mmr1: Advancing the
537 frontiers of multimodal reasoning. <https://github.com/LengSicong/MMR1>, 2025.

540 Jia Li, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Huang, Kashif
 541 Rasul, Longhui Yu, Albert Q Jiang, Ziju Shen, et al. Numinamath: The largest public dataset in
 542 ai4maths with 860k pairs of competition math problems and solutions. *Hugging Face repository*,
 543 13:9, 2024.

544 Ming Li, Jike Zhong, Shitian Zhao, Yuxiang Lai, and Kaipeng Zhang. Think or not think: A study of
 545 explicit thinking in rule-based visual reinforcement fine-tuning. *arXiv preprint arXiv:2503.16188*,
 546 2025.

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

550 Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min
 551 Lin. Understanding r1-zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*,
 552 2025.

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

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

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

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

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

567 John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional
 568 continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*,
 569 2015.

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

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

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

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

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

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

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

594 Cheng Wen, Tingwei Guo, Shuaijiang Zhao, Wei Zou, and Xiangang Li. Sari: Structured audio
 595 reasoning via curriculum-guided reinforcement learning. *arXiv preprint arXiv:2504.15900*, 2025.
 596

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

600 Jianhao Yan, Yafu Li, Zican Hu, Zhi Wang, Ganqu Cui, Xiaoye Qu, Yu Cheng, and Yue Zhang.
 601 Learning to reason under off-policy guidance. *arXiv preprint arXiv:2504.14945*, 2025.
 602

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

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

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

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

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

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

621
 622
 623
 624
 625
 626
 627
 628
 629
 630
 631
 632
 633
 634
 635
 636
 637
 638
 639
 640
 641
 642
 643
 644
 645
 646
 647

648 A TRAINING DETAILS
649650 A.1 TRAINING HYPERPARAMETERS
651652 The detailed training hyperparameters are provided in Tables 4 and 5, and all experiments are
653 conducted on 8 NVIDIA A100 GPUs, each equipped with 80 GB of memory.
654655 Table 4: Key hyperparameters for VLM training. Table 5: Key hyperparameters for LLM training.
656

Name	Value	Name	Value
Rollout num	8	Rollout num	5
Train batch size	128	Train batch size	256
Rollout temperature 1.0		Rollout temperature 1.0	
Mini batch size	128	Mini batch size	128
Micro batch size per GPU	2	Micro batch size per GPU	2
Learning rate	1.0e-6	Learning rate	1.0e-6
Entropy coefficient	0.0	Entropy coefficient	0.0
KL loss coefficient	0.001	KL loss coefficient	0.001
Max prompt length	8192	Max prompt length	8192
Max response length	4096	Max response length	4096
GPU memory utilization	0.7	GPU memory utilization	0.7

669
670 A.2 COMPARISON OF DATASET SIZES
671672 Table 6 summarizes the training resource configurations of ADORA and other baselines, detailing
673 the amount of data consumed at different post-training stages. The results demonstrate that ADORA
674 achieves competitive effectiveness while maintaining superior data efficiency.
675676 Table 6: Cold-Start and RL training data comparison of multimodal methods.
677

Model	Cold-Start Data	RL Data
MM-EUREKA-7B (Meng et al., 2025)	54k (open-source)	9.3k (open-source)
MMR1-math-v0 (Leng* et al., 2025)	None	6k (open-source)
Vision-R1-7B (Huang et al., 2025)	200k (synthetic data)	10k (open-source)
ADORA (ours)	None	2k (open-source)

B STUDY CASES

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

GRPO In a table tennis tournament every participant played every other participant exactly once. Although there were twice as many right-handed players as left-handed players, the number of games won by left-handed players was 40% more than the number of games won by right-handed players. (There were no ties and no ambidextrous players.) What is the total number of games played?

ADORA  To solve this problem, let's define the variables and set up the equations based on the given information.

Let (L) be the number of left-handed players and (R) be the number of right-handed players. According to the problem, there are twice as many right-handed players as left-handed players, so we have:

$$R = 2L$$

Each player plays every other player exactly once, so the total number of games played is the number of ways to choose 2 players out of ($L + R$) players, which is given by the combination formula:

$$\binom{L+R}{2} = \frac{(L+R)(L+R-1)}{2}$$

Let (W_L) be the number of games won by left-handed players and (W_R) be the number of games won by right-handed players. According to the problem, the number of games won by left-handed players is 40% more than the number of games won by right-handed players, so we have:

$$W_L = 1.4W_R$$

Since each game has exactly one winner, the total number of games won is equal to the total number of games played:

$$W_L + W_R = \binom{L+R}{2}$$

Substituting ($W_L = 1.4W_R$) into the equation, we get:

$$1.4W_R + W_R = \binom{L+R}{2}$$

$$2.4W_R = \binom{L+R}{2}$$

Now, let's substitute ($R = 2L$) into the equation:

$$2.4W_R = \binom{3L}{2}$$

$$2.4W_R = \frac{3L(3L-1)}{2}$$

$$4.8W_R = 3L(3L-1)$$

$$W_R = \frac{3L(3L-1)}{4.8}$$

$$W_R = \frac{3L(3L-1)}{8} = \frac{5L(3L-1)}{8}$$

Since (W_R) must be an integer, ($5L(3L-1)$) must be divisible by 8. Let's check the possible values of (L) that satisfy this condition.

If ($L = 1$): $5 \cdot 1 \cdot (3 \cdot 1 - 1) = 10$ (not divisible by 8)
 If ($L = 2$): $5 \cdot 2 \cdot (3 \cdot 2 - 1) = 50$ (not divisible by 8)
 If ($L = 3$): $5 \cdot 3 \cdot (3 \cdot 3 - 1) = 120$ (divisible by 8)

So, ($L = 3$) is a valid solution. Then ($R = 2L = 6$).

The total number of games played is:

$$\binom{3+6}{2} = \binom{9}{2} = \frac{9 \cdot 8}{2} = 36$$

Let's verify the solution:

$$W_R = \frac{5 \cdot 3 \cdot (3 \cdot 3 - 1)}{8} = \frac{5 \cdot 3 \cdot 8}{8} = 15$$

$$W_L = 1.4W_R = 1.4 \cdot 15 = 21$$

$$W_L + W_R = 21 + 15 = 36$$

The total number of games played is (36) </think>

<answer> 36 </answer>



<think> To solve this problem, let's denote the number of left-handed players as (L) and the number of right-handed players as (R). According to the problem, ($R = 2L$).

Each player plays every other player exactly once, so the total number of games played is the number of ways to choose 2 players out of the total ($L + R$) players, which is given by the combination formula ($\binom{L+R}{2} = \frac{(L+R)(L+R-1)}{2}$).

Let's denote the number of games won by left-handed players as (W_L) and the number of games won by right-handed players as (W_R). According to the problem, ($W_L = 1.4W_R$).

Since every game has a winner, the total number of games won is equal to the total number of games played, so ($W_L + W_R = \binom{L+R}{2}$).

Substituting ($R = 2L$) into the equations, we get:

$$W_L + W_R = \binom{3L}{2} = \frac{3L(3L-1)}{2}$$

$$W_L = 1.4W_R$$

Let ($W_R = x$). Then ($W_L = 1.4x$), and the total number of games won is:

$$1.4x + x = 2.4x = \frac{3L(3L-1)}{2}$$

$$x = \frac{3L(3L-1)}{4.8} = \frac{5L(3L-1)}{8}$$

Since ($W_R = x$) and ($W_R = \binom{2L}{2} = \frac{2L(2L-1)}{2} = L(2L-1)$), we can set up the equation:

$$L(2L-1) = \frac{5L(3L-1)}{8}$$

$$8L(2L-1) = 5L(3L-1)$$

$$8L(2L-1) = 5(3L-1)$$

$$16L - 8 = 15L - 5$$

$$L = 3$$

So, ($R = 2L = 6$). The total number of players is ($L + R = 3 + 6 = 9$). The total number of games played is:

$$\binom{9}{2} = \frac{9 \times 8}{2} = 36$$

Thus, the total number of games played is (36). </think>

<answer> 36 </answer>



Qwen2.5-VL
78-instruct

To determine the total number of games played in the tournament, we first need to establish the relationship between the number of left-handed and right-handed players and the number of games won by each group.

Let (L) be the number of left-handed players and (R) be the number of right-handed players. According to the problem, ($R = 2L$). The total number of players is ($L + R = L + 2L = 3L$).

The total number of games played in a round-robin tournament where every participant plays every other participant exactly once is given by the combination formula ($\binom{3L}{2} = \frac{3L(3L-1)}{2}$).

Let (W_L) be the number of games won by left-handed players and (W_R) be the number of games won by right-handed players. According to the problem, ($W_L = 1.4W_R$).

In any individual game, there is a winner and a loser, and the total number of games won by all players must equal the total number of games played. Therefore, we have:

$$W_L + W_R = \frac{3L(3L-1)}{2}$$

Substituting ($W_L = 1.4W_R$) into the equation, we get:

$$1.4W_R + W_R = \frac{3L(3L-1)}{2},$$

$$2.4W_R = \frac{3L(3L-1)}{2},$$

$$W_R = \frac{3L(3L-1)}{2 \cdot 2.4} = \frac{3L(3L-1)}{4.8} = \frac{5L(3L-1)}{8}.$$

Now, we need to check the consistency of these values. Since (W_R) must be an integer, ($5L(3L-1)$) must be divisible by 8. We can test small values of (L) to find a suitable value.

If ($L = 4$), then ($\frac{5+4(12-1)}{8} = \frac{5+411}{8} = \frac{225}{8} = 27.5$), which is not an integer.

If ($L = 5$), then ($\frac{5+5(15-1)}{8} = \frac{5+50}{8} = \frac{55}{8} = 6.875$), which is not an integer.

If ($L = 6$), then ($\frac{5+6(18-1)}{8} = \frac{5+61}{8} = \frac{66}{8} = 8.25$), which is not an integer.

If ($L = 8$), then ($\frac{5+8(24-1)}{8} = \frac{5+192}{8} = \frac{197}{8} = 24.625$), which is an integer.

Thus, ($L = 8$) is a valid solution. Therefore, the total number of players is ($3L = 3 \cdot 8 = 24$), and the total number of games played is:

$$\frac{3L(3L-1)}{2} = \frac{24 \cdot 23}{2} = 276$$

The total number of games played is (276).

Figure 5: 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.

C 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 9 and 10.

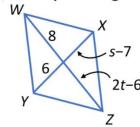
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809

"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 s in the parallelogram.

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



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 t . 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 :

$$\begin{aligned} 8 + 6 &= 2t \\ 14 &= 2t \\ t &= \frac{14}{2} \\ t &= 7 \end{aligned}$$

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:

$$\begin{aligned} 8 + 6 &= 2t \\ 14 &= 2t \\ t &= \frac{14}{2} \\ t &= 7 \end{aligned}$$

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:

$$\begin{aligned} 8 + 6 &= 2t \\ 14 &= 2t \\ t &= \frac{14}{2} \\ t &= 7 \end{aligned}$$

The correct answer is:

$$\boxed{7}$$



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

- 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 $\boxed{6}$.



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:

$$\boxed{A}$$

Figure 6: 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.

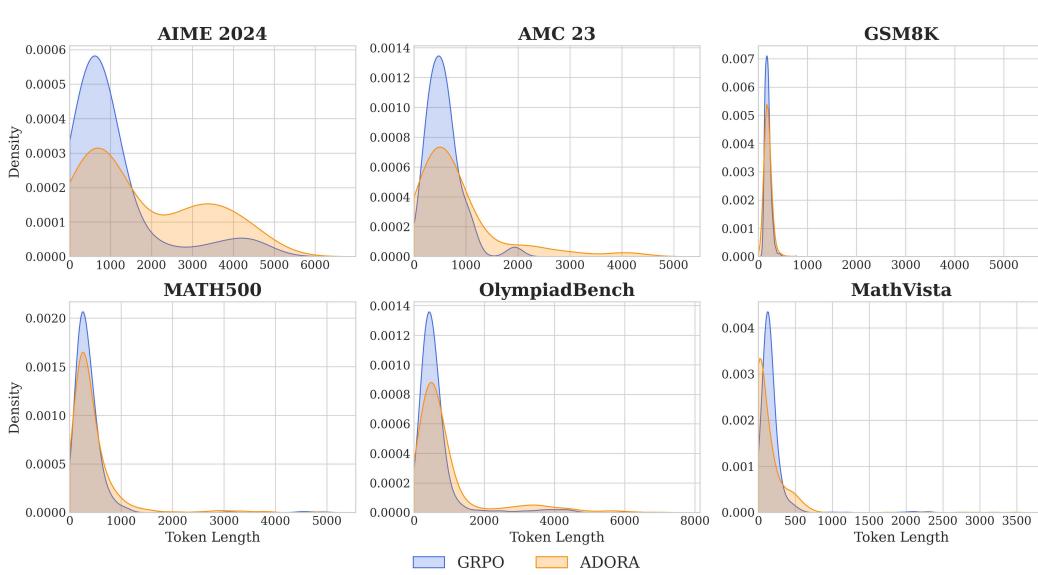


Figure 7: Comparison of Token Length Distributions Generated by GRPO and ADORA across Various Reasoning Benchmarks.

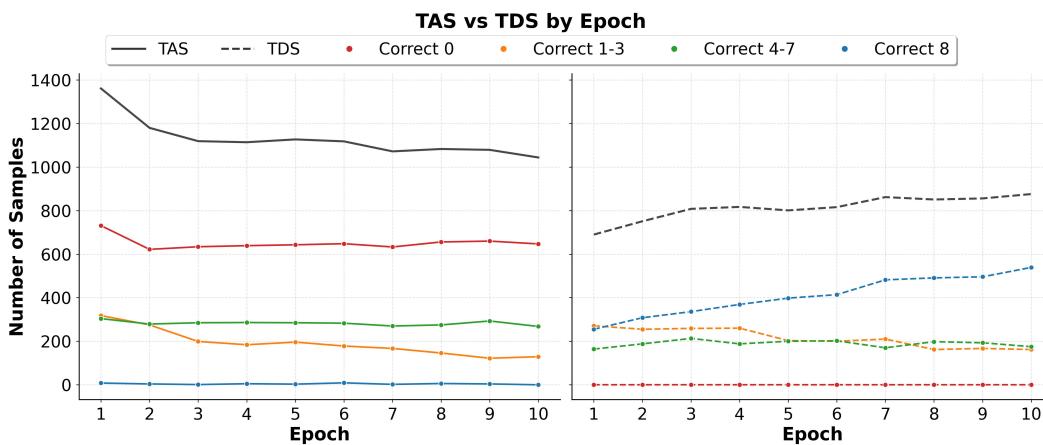


Figure 8: The changes in the number of samples of each difficulty level for the two corresponding categories of samples across epochs.

864 Prompt to Detect Overthinking-1

865

866 **System Prompt:**

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

868

869 {
870 <INTERACTION> trajectory goes here </INTERACTION>
871 }

872 Analyze the <INTERACTION> and determine if the model is preferring their internal reasoning chain over interacting with the environment:

873 How could this be detected?

874 <CORE PRINCIPLE>

875 • The model suffers from Analysis Paralysis; it focuses on heavy planning instead of interacting with the environment.

876 • The model suffers from Rogue actions. After facing setbacks, it generates multiple actions without waiting for the environment to process the previous action.

877 • 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.

878

879 </CORE PRINCIPLE>

880 <SCORING SYSTEM (0-10)>

881 **0-3: Always interacting with the environment**

882

883 • A summary of what has been done so far is good, even if done multiple times.

884 • A brief summary of the steps to take is good if the model interacts with the environment, following steps one by one.

885 • Only one action per turn, finish, and other actions are NOT allowed.

886 • Alternating between two operations is good.

887 • 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.

888 • Repeating similar patterns or configurations is fine as long as the model interacts with the environment between attempts.

889 • Detailed reasoning and planning are good if they lead to concrete actions with environment interaction.

890

891 **4-7: Sometimes relies too much on their internal reasoning chain, but still interacts with the environment.**

892

893 • It engages in heavy planning, but still interacts with the environment.

894 • It NEVER concludes the task without checking with the environment.

895 • It might output multiple steps ONE time, but at subsequent turns, it interacts one step at a time.

896 • Long theoretical discussions are acceptable if they eventually result in concrete actions.

897

898 **8-10: Completely relies on their internal reasoning chain.**

899

900 • Focuses solely on their internal reasoning chain, with no concrete actions following the analysis.

901 • Generates multiple actions without waiting for the environment response.

902 • The model prematurely concludes the task. Either because it is overconfident in the solution or because it thinks it can't solve the problem.

903 • Generates many steps without any environment interaction.

904 • Gets stuck in endless theoretical discussion without attempting solutions.

905

906

907 </SCORING SYSTEM>

908

909

910

911

912

913

914

915

916

917

Figure 9: 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 10: The prompt for overthinking scoring.

D LIMITATIONS

(1) **Specific differentiation strategies may require tuning for new tasks/models:** This implies that the definitions of Temporarily Advantageous Samples (TAS) and Temporarily Disadvantageous Samples (TDS) within ADORA, along with their weight adjustment mechanisms, might not be universally applicable. When applying ADORA to new tasks or models, these strategies may need to be redesigned or adjusted. (2) **Efficacy is tied to rollout quality:** ADORA relies on the outcomes of the model's online rollouts to dynamically assess sample utility. If the quality of these rollouts is low (e.g., the model generates poor-quality reasoning trajectories), then the classification of samples and the subsequent weight adjustments may be inaccurate, consequently impacting overall training effectiveness.

E USE OF LLMs

We use LLMs solely to assist in language polishing and enhancing readability. LLMs are not involved in the generation of data or the preparation of this paper. All technical research content, experiments and analysis are conducted and completed by the authors.