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. However, prevailing methods typically employ static advantage estimation, neglecting the dynamic utility of training samples over time. This limitation often results in 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 simple yet effective reinforcement learning technique that dynamically differentiates training data into temporarily advantageous and disadvantageous samples through model rollouts guided by a tailored data differentiation strategy. Instead of static optimization, ADORA adjusts advantage signals on the fly, enabling more efficient policy updates. Extensive evaluations on various tasks demonstrate that ADORA significantly enhances long chain-of-thought reasoning in both mathematical and geometric tasks across large language models and vision–language models, achieving notable performance gains.

1 Introduction

Recently, R1-like reasoning models have attracted significant attention for their remarkable performance on challenging mathematical reasoning tasks through extensive chains of thought in both LLMs (Liu et al., 2025) and VLMs (Shen et al., 2025). The technical report introducing R1 (Guo et al., 2025) has already demonstrated that reinforcement learning (RL) fine-tuning plays a pivotal role in enabling this reasoning capability. In particular, Group Relative Policy Optimization (GRPO) (Zhang and Zuo, 2025), which removes the critic network and replaces Generalized Advantage Estimator (GAE) (Schulman et al., 2015) with a rule-based, outcome-driven reward scheme, has emerged as a promising alternative to traditional methods such as PPO (Schulman et al., 2017) and DPO (Rafailov et al., 2023), primarily due to its efficiency and its intrinsic compatibility with language model training.

However, existing GRPO implementations still face substantial limitations. One key issue is that the static computation of sample utility implicitly assumes that the informativeness of each training example remains constant throughout policy optimization, thereby ignoring the dynamic nature of learning and severely hindering both training efficiency and the performance ceiling of RL. 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 use of data, as also noted by observations that current methods lack robust mechanisms for handling samples of varying utility during training (Ye et al., 2025).

To address this limitation, we propose that a sample's advantage should evolve alongside the policy. We introduce **ADORA** (**A**dvantage **D**ynamics via **O**nline **R**ollout **A**daptation), a novel RL framework designed to dynamically calibrate advantage estimation. 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 reweights 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 conducted extensive controlled experiments on both LLMs for mathematical reasoning and VLMs for geometry reasoning. Empirically, ADORA significantly improves long chain-of-thought reasoning and task generalization. For instance, on the Qwen-7B-base model, ADORA achieved an average of 3.4 percentage points improvement over standard GRPO on math tasks. For VLMs, using fewer than 2,000 samples and no warm-starting, the Qwen2.5-VL-7B-instruct model achieved 73.5% accuracy on MathVista with ADORA.

Our key contributions and findings include:

- **The ADORA framework**: A plug-and-play method for dynamically calibrating advantage estimation weights in RL based on live rollout statistics.
- **Task-specific differentiation strategies**: We designed and validated distinct strategies for distinguishing TAS and TDS across different reasoning domains, consistently demonstrating improvements over GRPO.
- **Comprehensive empirical analysis**: Extensive experiments statistically evaluate ADORA's impact on reflective token frequency, CoT length, generalization ability, and Pass@K scaling laws, providing insights into its mechanisms of action.

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 capabilities. 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., 2025) 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, adaptive, and model-specific 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. This approach requires no cold-start, leverages the entire dataset effectively, and has demonstrated steady improvements in both response length and performance for LLMs and VLMs.

3 Method

This section details ADORA , our proposed framework for dynamically guiding reinforcement learning. ADORA achieves this by classifying training samples into TAS or TDS categories based on the model's live rollouts. The core idea is to make the model focus its learning effort on TAS, with this classification evolving dynamically as training progresses. We first briefly review GRPO, the baseline upon which ADORA builds.

3.1 GRPO

Due to the success of Deepseek-R1 (Guo et al., 2025), GRPO (Shao et al., 2024) becomes the de facto approach with zero-RL training. Unlike standard PPO, GRPO eliminates the need for a separate value network by computing sample-wise advantages directly from normalized reward scores across multiple sampled trajectories. Let $\mathcal{D} = \{(x_i, r_i)\}_{i=1}^N$ be prompts with scalar rewards. Denote the trainable policy by π_θ and a frozen reference by π_{ref}. Given a question q, a set of sampled responses $\{\tau_i\}_{i=1}^N$ generated by the old policy $\pi_{\theta_{\text{old}}}$, and a reward function $R(\tau_i)$, GRPO computes the per-sample advantage A_i as:

$$A_i = \frac{R(\tau_i) - \text{mean}\left(R(\tau_1), \ldots, R(\tau_N)\right)}{\text{std}\left(R(\tau_1), \ldots, R(\tau_N)\right)}, \quad (1)$$

This normalization is done globally over the current batch, meaning all A_i values within one epoch are determined solely by the reward values and are independent of the specific token-level predictions. Crucially, during training, $R(\tau_i)$ is indirectly shaped by the updated policy through its effect on the token-level probabilities.

To estimate the policy update, GRPO uses token-level importance weights:

$$r_{i,t}(\theta) = \frac{\pi_\theta(\tau_{i,t}|q, \tau_{i,<t})}{\pi_{\theta^{\text{old}}}(\tau_{i,t}|q, \tau_{i,<t})}, \quad (2)$$

The GRPO training objective is then defined as:

$$\mathcal{J}_{\text{GRPO}}(\theta) = \frac{1}{\sum_{i=1}^N |\tau_i|} \sum_{i=1}^N \sum_{t=1}^{|\tau_i|} \min\left[r_{i,t}(\theta) A_i,\right. \\ \left.\text{clip}\left(r_{i,t}(\theta); 1-\epsilon, 1+\epsilon\right) A_i\right] - \beta \cdot \mathbb{D}_{\text{KL}}, \quad (3)$$

In the RL objective, GRPO follows PPO, using the importance sampling ($r_{i,t}$ in Eq. (2)) to calibrate the gradient as the rollouts are generated by $\pi_{\theta_{\text{old}}}$. The clipping threshold ϵ stabilizes training by preventing large deviations in token probability ratios. This makes A_i a fixed scalar weight across all tokens in a sample, and its interaction with $r_{i,t}(\theta)$ ensures that updates are localized to high-reward trajectories. Crucially, in standard GRPO, the per-sample advantage A_i is computed based on rewards and remains static throughout an epoch or even the entire training process for that sample. This static nature, as discussed in Section (1), limits its adaptability to the model's evolving capabilities.

3.2 ADORA

While GRPO normalizes scalar rewards into sample-level advantages A_i, it treats all samples equally during training. To better leverage the heterogeneous quality and utility of training trajectories, we propose **ADORA**, which dynamically calibrates the advantage estimates by assigning per-sample weights depending on whether a rollout is deemed temporarily advantageous or disadvantageous within the current epoch.

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

$$\tilde{A}_i = w_i \cdot A_i, \quad (4)$$

where A_i is the normalized reward-based advantage as in GRPO. The ADORA training objective is given by:

$$\mathcal{J}_{\text{ADORA}}(\theta) = \frac{1}{\sum_{i=1}^N |\tau_i|} \sum_{i=1}^N \sum_{t=1}^{|\tau_i|} \min\Big[r_{i,t}(\theta)\tilde{A}_i, \\ \text{clip}\left(r_{i,t}(\theta);\ 1-\epsilon,\ 1+\epsilon\right)\tilde{A}_i\Big] - \beta \cdot \mathbb{D}_{\text{KL}}, \quad (5)$$

Since w_i is trajectory-level and independent of sampled actions, this modification preserves the unbiasedness of the policy gradient.

Two key questions must be addressed:

1. How to determine whether a sample is advantageous or not?

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

VLM Case: Length-Based Advantage Attenuation (Subtraction): Visual language models (VLMs) often exhibit weak multi-hop reasoning capabilities in the early stages of RL training. Standard GRPO optimization, with its static advantages, can thus overfit on short, trivial responses if these yield high initial rewards, hindering progress on complex, long-horizon reasoning tasks. ADORA employs an attenuation strategy for VLMs to penalize unpromising samples. Specifically, we define a sample as temporarily disadvantageous if its successful rollout length does not significantly exceed the average length of unsuccessful rollouts:

Specifically, We define a length advantage as:

$$\text{Length\_adv} \iff \max(\text{length}_{\text{correct}}) > \text{mean}(\text{length}_{\text{incorrect}}), \tag{6}$$

Based on this criterion, we assign:

$$w_i = \begin{cases} 1, & \text{if Length\_adv} \\ 0.1, & \text{otherwise} \end{cases}, \tag{7}$$

Temporarily advantageous samples retain their full advantage signal ($w_i = 1$), while temporarily disadvantageous ones are down-weighted ($w_i = 0.1$). This "subtraction" mechanism reduces the negative impact of samples that are currently unhelpful for long-horizon learning.

LLM Case: Length + Difficulty-Based Advantage Amplification (Addition): In contrast, large language models (LLMs) for tasks like math reasoning may already possess moderate reasoning ability at initialization. Under GRPO, response lengths tend to grow, and many samples can provide useful learning signals. Therefore, rather than primarily suppressing low-utility samples, ADORA for LLMs focuses on amplifying the signal from high-value samples—those that are both difficult and demonstrate promising reasoning depth.

We additionally define a difficulty advantage as:

$$\text{Difficulty\_adv} \iff \text{Correct rate} \leq 0.5, \tag{8}$$

and again reuse $Length\_adv$ from above. We then assign:

$$w_i = \begin{cases} 2, & \text{if Difficulty\_adv \& Length\_adv} \\ 1, & \text{otherwise} \end{cases} \tag{9}$$

Temporarily advantageous samples (those that are harder and longer) receive an amplified learning signal ($w_i = 2$), while others retain their original strength ($w_i = 1$). This "addition" effect reinforces learning from challenging and instructive samples, promoting curriculum-style progression.

In summary, ADORA introduces a general and lightweight mechanism to enhance GRPO via dynamic advantage calibration. By re-weighting training samples adaptively, it supports more targeted policy optimization across both weak (VLM) and strong (LLM) model regimes.

3.3 Algorithm

Algorithm 1 shows an ADORA instantiation. ADORA-GRPO replaces the value function with group baselines (Shao et al., 2024).

Algorithm 1 ADORA-GRPO

Input: Policy π_θ, reference π_{ref}, data $\mathcal{D}$

1: Initialise Adam optimiser
2: **repeat**
3: Sample mini-batch $\{x_i\}_{i=1}^B \sim \mathcal{D}$
4: Generate actions $a_i \sim \pi_\theta(\cdot|x_i)$ and rewards r_i
5: Compute advantages $\hat{A}_i$ via GRPO
6: $w_i \leftarrow \text{weight\_func}(\xi_i)$
7: $\tilde{A}_i \leftarrow w_i \cdot \hat{A}_i$ (ADORA)
8: Update $\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}_{\text{ADORA}}$
9: **until** convergence or budget exhausted

Implementation: In practice, the dynamic weighting can be implemented as:

$$w_i = \text{weight\_func}(\text{seqs}, \text{rewards}, \text{aux\_info}), \tag{10}$$

where $weight\_func$ computes w_i from rollout statistics such as CoT length and correct rate.

4 Experiment

To empirically validate the efficacy of ADORA, we conduct a series of controlled experiments across both LLMs for mathematical reasoning and VLMs for geometry reasoning. Our primary goal is to demonstrate that ADORA's dynamic advantage calibration leads to improved performance and training efficiency compared to a baseline GRPO. We select Qwen2.5-7B-Base (Yang et al., 2024) for LLM tasks and Qwen2.5-VL-7B (Bai et al., 2025) for VLM tasks. This experimental setup allows us to directly test our central hypothesis: dynamically adjusting advantage weights based on rollout statistics enhances the learning process for complex reasoning.

Part-1: VLM

ADORA and the baseline GRPO were initialized with Qwen2.5-VL-7B-Instruct and directly trained via Reinforcement Learning (RL) on 2000 samples from the Geometry3K training set (Lu et al., 2021), without a cold-start phase. Detailed training hyperparameter settings are reported at (A). All training was conducted over three independent runs, with average performance reported. For evaluation, VLM performance was primarily assessed

Model	Base model	Cold-Start Data	RL Data
MM-EUREKA-8B (Meng et al., 2025)	InternVL2.5-8B-Ins	54k (open-source)	9.3k (open-source)
MMR1-math-v0 (Leng* et al., 2025)	Qwen2.5-VL-7B-Ins	None	6k (open-source)
Vision-R1-7B (Huang et al., 2025)	Qwen2.5-VL-7B-Ins	200k (synthetic data)	10k (open-source)
ADORA (ours)	Qwen2.5-VL-7B-Ins	None	2k (open-source)

Table 1: Cold-Start and RL training data comparison of multimodal methods with different base models.

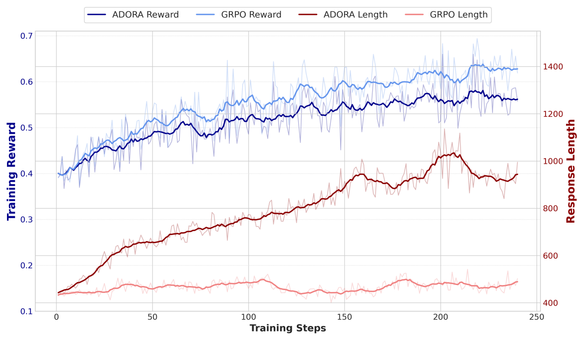


Figure 1: Training dynamics comparison of GRPO vs ADORA on Qwen2.5-VL-7B-Instruct (geometry3k). GRPO exhibits stagnant response length growth with KL/policy loss outliers. ADORA achieves sustained length expansion with stabilized optimization at the cost of slight training reward degradation because it performs "subtraction" on VLMs. Benchmark results demonstrate ADORA's superior in/out-of-domain task performance.

Model	MathVista			MMStar
	avg	id	ood	
Qwen2.5-VL-7B-Instruct	67.3	69.6	65.5	63.9
Claude3.7-Sonnet	66.8	-	-	-
Gemini2-flash	70.4	-	-	-
MM-EUREKA-8B	68.1	73.4	63.8	<u>64.3</u>
MMR1-math-v0	70.2	72.3	68.5	**64.9**
Vision-R1-7B (report)	<u>73.5</u>	**81.9**	66.8	-
GRPO	70.2	71.6	<u>69.1</u>	61.9
ADORA (ours)	**73.5**	<u>76.1</u>	**71.4**	63.8

Table 2: Zero-shot pass@1 performance on benchmarks across various difficulty based on Qwen2.5-VL-7B-Ins. Dashes (–) denote unavailable official scores. Bold and underline represent the 1st and 2nd in performan.

on MathVista (Lu et al., 2023) and MMStar (Chen et al., 2024) datasets using a sampling temperature of 0. MathVista contains 44.7% In-Domain (id) geometric tasks and 55.3% Out-of-Domain (ood) non-geometric samples.The evaluation metric was pass@1, with results averaged over three runs.

(Meng et al., 2025; Leng* et al., 2025; Huang et al., 2025)have reproduced R1 on VLMs. Table (1) presents the resource configurations for ADORA and these works in terms of base model selection, cold-start usage, and the amount of training data at each stage. It demonstrates that ADORA operates without a cold start and utilizes minimal data. The results in Table (2) indicate ADORA's significant improvements over the baseline GRPO on all metrics. Specifically, ADORA achieves 73.5% on MathVista, matching Vision-R1-7B (Huang et al., 2025) and considerably outperforming Claude3.7-Sonnet and Gemini2-flash, alongside stronger OOD capabilities. Although ADORA's performance on MMStar is slightly (0.1%) below Qwen2.5-VL-7B, this is still a substantial gain over GRPO, which underperformed Qwen2.5-VL-7B by 2%. This slight variance is attributed to the potential limitations of employing homogeneous training data.

Part-2: LLM

We conducted RL training directly on Qwen2.5-7b-base with the Math dataset (Hendrycks et al., 2021), which contains 12,000 samples, and used Math-Verify to perform rule-based outcome verification using Math500 (Hendrycks et al., 2021) as the test set. For both GRPO and ADORA, we carried out three separate RL training runs and reported the average performance. Detailed training hyperparameter settings are reported at (A). For evaluation, we mainly focus on seven widely used math reasoning benchmarks, including GSM8K (Cobbe et al., 2021),Gaokao2023 (Zhang et al., 2024), College-Math (Tang et al., 2024), AIME24, AMC23 (Li et al., 2024), OlympiadBench (He et al., 2024), and MATH500 (Hendrycks et al., 2021). For all those benchmarks, we report pass@1, setting the sampling temperature to 0 and repeating the evaluation three times, taking the average result.

Table (3) reports the performance of ADORA

Model	GSM8K	Math500	AMC23	Gaokao2023	CollegeMath	OlympiadBench	AIME24	Avg
Qwen2.5-7B-base	56.3	57.2	37.5	42.0	24.3	26.3	10.0	36.2
GRPO	89.1	73.2	50.0	52.7	28.6	35.1	13.3	48.7
ADORA + L_adv	90.0	74.8	52.5	53.0	29.0	34.6	16.7	50.2
ADORA + D_adv	**90.3**	75.0	55.0	52.7	28.8	35.4	16.7	50.7
ADORA + L_adv + D_adv	89.6	**76.2**	**62.5**	**54.3**	**29.3**	**36.0**	**16.7**	**52.1**

Table 3: Zero-shot pass@1 performance on math benchmarks across various difficulty based on Qwen2.5-7B-base. L_adv and D_adv represent the $Length\_adv$ and the $Difficulty\_adv$ used in data classification. Bold and underline represent the 1st and 2nd in performan.

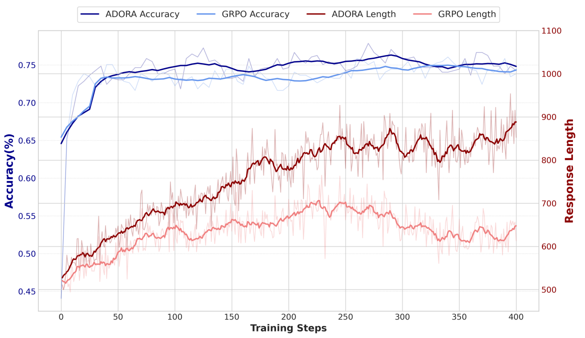


Figure 2: Training dynamics comparison of GRPO vs ADORA on Qwen2.5-7B-base. As training progresses, GRPO exhibits a non-monotonic trend in chain-of-thought (CoT) length, initially increasing and subsequently decreasing. In contrast, ADORA demonstrates a consistently increasing CoT length, with a growth rate approximately three times that of GRPO. Moreover, ADORA achieves a performance improvement over GRPO on the Math500 test set.

and GRPO under various settings indicating that ADORA achieved higher scores compared to GRPO. Specifically, when only the $Length\_adv$ (L_adv) or $Difficulty\_adv$ (D_adv) was applied, the performance improvement was modest. However, when both L_adv and D_adv were used, ADORA consistently outperformed GRPO across all test sets, achieving scores such as 76.2 (+3.0) on Math500 and 62.5 (+12.5) on AMC23, among others. On average, ADORA outperformed GRPO by 3.4 points, demonstrating that dynamically adjusting advantage estimates during training effectively guides the model toward learning from more beneficial samples, thereby enhancing its generalization capability.

Efficiency and performance. In addition, we compared the differences between DAPO (Yu et al., 2025) and ADORA in terms of training efficiency and final performance. Specifically, compared to GRPO and ADORA, DAPO requires the training data advantage computation to be non-zero, which leads to more efficient training. However, in terms of final performance, DAPO shows no significant advantage over ADORA. Furthermore, when we incorporate the ADORA method on top of DAPO, both the training efficiency and final performance are further improved.

5 Analysis

Beyond achieving superior aggregate performance, understanding of how ADORA improves reasoning is crucial. This section analyzes ADORA's impact on model behavior and learning dynamics relative to the GRPO baseline. We investigate several facets: the induced cognitive patterns via reflection frequency (Section (5.1)), structural changes in model outputs via length distributions (Section (5.2)), the adaptive learning trajectory (Section (5.3)), and the upper-bound reasoning capabilities through Pass@K analysis (Section (5.4)). These analyses collectively illuminate the mechanisms underlying ADORA's effectiveness.

5.1 Reflection Frequency

A key aspect of understanding how different reinforcement learning strategies influence reasoning capabilities lies in examining the model's explicit thought processes. Section (5.1) delves into the frequency of reflective vocabulary, providing insights into the cognitive behaviors fostered by ADORA compared to the baseline GRPO method across various mathematical benchmarks, as illustrated in Figure 3. This analysis aims to quantify the tendency of models to engage in self-monitoring, verification, and structured thinking during problem-solving.

Two major trends are observed: **Increased use of core reflective terms**: Words that directly indicate verification, evaluation, and deliberate reasoning—such as "verify", "evaluate", "consider",

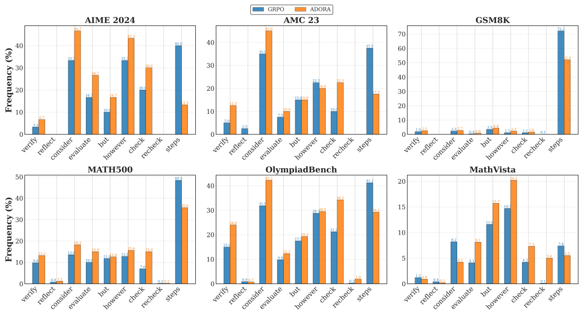


Figure 3: Distribution of Reasoning-Related Keywords for GRPO and ADORA across Various Reasoning Benchmarks.

"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 AIME 2024 benchmark, while "evaluate" shows similar trends on AMC23 and MATH500. **More structured and transitional language**: Terms that signal structured reasoning, such as "but" and "however" are also used more frequently by ADORA-trained models on several benchmarks (e.g., MATH500, OlympiadBench). In addition, compared with GRPO, the frequency of the word "steps" drops significantly in ADORA.

By dynamically calibrating advantage estimates, ADORA preferentially rewards trajectories exhibiting deeper reasoning, cautious verification, and structured expression. Assigning greater learning weight to TAS directly reinforces the associated cognitive behavior patterns and lexical expressions. Compared to the standard GRPO approach, the ADORA training framework more effectively encourages the model to develop a reasoning style characterized by more frequent self-reflection, verification, and structured thinking. This shift in cognitive behavior, as reflected in the model's output text, represents a key underlying factor behind ADORA's performance gains across evaluated mathematical reasoning tasks.

5.2 Distribution of Length Differences

This section examines how reinforcement learning frameworks influence response length, a structural characteristic indicative of Chain-of-Thought (CoT) elaboration and reasoning depth. Figure 4 consistently shows that across multiple benchmarks, ADORA-trained models produce longer responses than GRPO-trained models, evidenced by rightward-shifted and heavier-tailed token length distributions for ADORA.

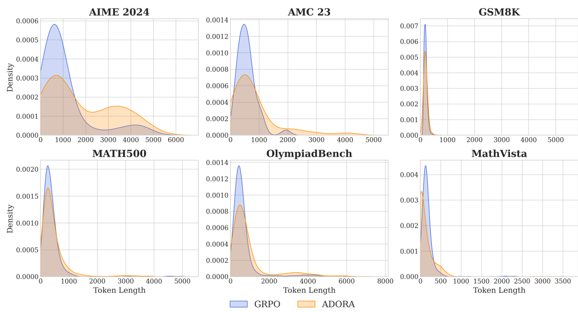


Figure 4: Comparison of Token Length Distributions Generated by GRPO and ADORA across Various Reasoning Benchmarks.

These longer responses under ADORA suggest more elaborate reasoning, clearer articulation, and thorough verification, offering greater "cognitive space" for complex problems; this aligns with Figure 1 and 2, where ADORA's CoT length grows significantly faster during training. Conversely, GRPO's tendency for shorter responses may indicate premature convergence to incomplete solutions due to shallower reasoning. Overall, ADORA impacts not only qualitative aspects (reflective vocabulary) but also quantitatively alters output structure (response length). By encouraging longer, more detailed responses, ADORA better equips models for complex tasks, further supporting its effectiveness through dynamic advantage calibration.

5.3 How ADORA affects the learning trajectory of RL?

To gain deeper insight into how the ADORA framework optimizes the reinforcement learning process through dynamic adjustment of advantage estimation, this section aims to examine ADORA's concrete influence on the model's learning trajectory. The central question is: how does ADORA dynamically select and emphasize training samples of varying difficulty and type based on the model's real-time performance during training? 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 progressively tackle more challenging problems.

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 dif-

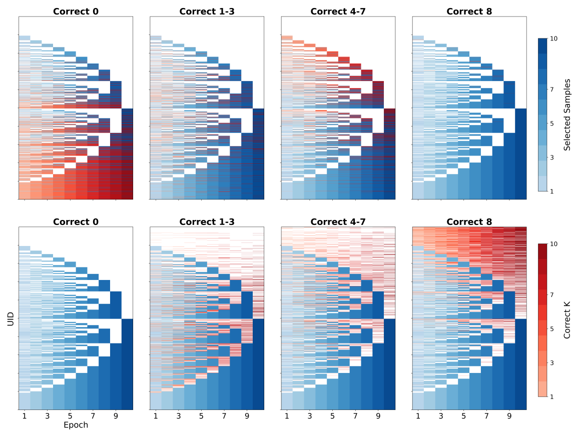


Figure 5: The blue sections represent the samples selected for each epoch (clustered for easier visualization), while the red sections illustrate the distribution of samples under different Correct N settings in once sampling, representing the difficulty of the samples, both of which gradually deepen as epochs progress. The top and bottom rows (from left to right) respectively show the changes in TAP and TDP as the difficulty increases (Correct 0, Correct 1-3, Correct 4-7, Correct 8).

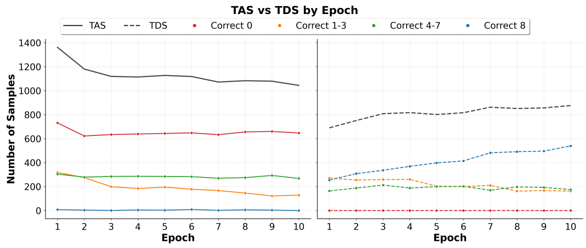


Figure 6: The changes in the number of samples of each difficulty level for the two corresponding categories of samples across epochs.

ficulty, "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 standard 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.

5.4 PASS@K: ADORA vs. GRPO

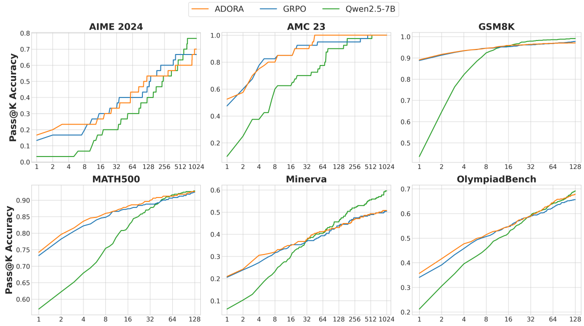


Figure 7: Pass@k curves of base model and ADORA/GRPO across multiple mathematical benchmarks.

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.

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

(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.

References

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.

Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, and 1 others. 2024. Are we on the right way for evaluating large vision-language models? *arXiv preprint arXiv:2403.20330*.

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*.

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.

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*.

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*.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36:53728–53741.

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*.

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

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*.

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

Zhenting Wang, Guofeng Cui, Kun Wan, and Wentian Zhao. 2025. 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*.

Boning Zhang, Chengxi Li, and Kai Fan. 2024. Mario eval: Evaluate your math llm with your math llm–a mathematical dataset evaluation toolkit. *arXiv preprint arXiv:2404.13925*.

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*.

A Training Hyperparameter Details

To conduct a rigorous controlled experiment, we consistently use the same detailed training hyperparameter settings for GRPO and ADORA.

LLM Training Settings. Based on Qwen2.5-7B-Base, we set the KL coefficient to 0.001 and the entropy coefficient to 0. Our rollout batch size is 256, and the update batch size is 128. Rollout generation is performed with a temperature of 1.0, generating 5 responses per sample, with a maximum response length of 4096 tokens. The learning rate is fixed at 1e-6 throughout training.

VLM Training Settings. Based on Qwen2.5-VL-7B-Ins, we set the KL coefficient to 0.001 and the entropy coefficient to 0. Our rollout batch size is 128, and the update batch size is 128. Rollout generation is performed with a temperature of 1.0, generating 8 responses per sample, with a maximum response length of 4096 tokens. The learning rate is fixed at 1e-6 throughout training.

B Study Cases

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

<think> 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 ($W_L = 1.4 W_R$) 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:

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

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 ($L = 1$):

$$5 \cdot 1 \cdot (3 \cdot 1 - 1) = 10 \quad \text{(not divisible by 8)}$$

If ($L = 2$):

$$5 \cdot 2 \cdot (3 \cdot 2 - 1) = 50 \quad \text{(not divisible by 8)}$$

If ($L = 3$):

$$5 \cdot 3 \cdot (3 \cdot 3 - 1) = 120 \quad \text{(divisible by 8)}$$

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 ($\boxed{36}$).</think>

<answer> $\boxed{36}$ </answer>

GRPO

<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:

$$L(2L-1) = \frac{5L(3L-1)}{8}$$
$$8L(2L-1) = 5L(3L-1)$$
$$8(2L-1) = 5(3L-1)$$
$$16L - 8 = 15L - 5$$
$$L = 3$$

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 ($\boxed{36}$). </think>
<answer> $\boxed{36}$ </answer>

Qwen2.5-VL-7B-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 the 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 number.

If ($L = 4$), then ($\frac{5 \cdot 4 \cdot (12-1)}{8} = \frac{5 \cdot 4 \cdot 11}{8} = \frac{220}{8} = 27.5$), which is not an integer.
If ($L = 5$), then ($\frac{5 \cdot 5 \cdot (15-1)}{8} = \frac{5 \cdot 5 \cdot 14}{8} = \frac{350}{8} = 43.75$), which is not an integer.
If ($L = 6$), then ($\frac{5 \cdot 6 \cdot (18-1)}{8} = \frac{5 \cdot 6 \cdot 17}{8} = \frac{510}{8} = 63.75$), which is not an integer.
If ($L = 8$), then ($\frac{5 \cdot 8 \cdot (24-1)}{8} = \frac{5 \cdot 8 \cdot 23}{8} = 115$), 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:

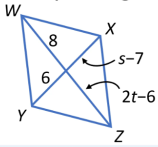
$$\frac{3L(3L-1)}{2} = \frac{24 \cdot 23}{2} = 276$$

The total number of games played is ($\boxed{276}$).

Figure 8: 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.

"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 9: 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.