Leveraging Error Diversity in Group Rollouts for Reinforcement Learning

Wenpu Liu[1,2*] **Yuqi Xu**[1,2*] **Weichu Xie**[1,2*] **Yongfu Zhu**[2] **Shuai Dong**[2,3] **Ziyue Wang**[1,2]
Wenqi Shao[2] Xiaoying Zhang[2] Tong Yang[1†] Nan Duan[2] Jiaqi Wang[2]
[1]Peking University [2]JD.COM [3]Shanghai Innovation Institute
[*]Equal contribution. [†]Corresponding author: yangtong@pku.edu.cn

Abstract

Reinforcement Learning from Verifiable Rewards (RLVR) typically samples multiple responses per prompt and assigns binary rewards based on individual correctness, yet the collective structure of the group output—specifically, the distribution of errors—is largely discarded. We identify this as a missed opportunity: empirical analysis reveals that error diversity within a group is a strong predictor of training success, with problems eliciting diverse wrong answers benefiting substantially more from RLVR than those producing homogeneous failures. Motivated by this observation, we propose **Error Diversity Advantage Shaping (EDAS)**, a lightweight, algorithm-agnostic technique that modulates the advantage signal for incorrect rollouts based on intra-group error diversity. EDAS amplifies penalties for dominant, repeated errors and attenuates penalties for rare, exploratory ones, thereby encouraging the model to maintain diverse reasoning paths and discouraging error perseveration. Crucially, EDAS operates as a simple post-hoc adjustment that can be seamlessly integrated into any RLVR algorithm. We validate EDAS on top of several mainstream RLVR methods across a series of models and seven challenging math benchmarks, demonstrating positive average gains across all evaluated settings. Notably, EDAS yields an average improvement of 6.29 points over DAPO on Qwen3-8B across seven benchmarks, confirming that exploiting the latent information in group rollouts is a broadly effective strategy for strengthening RLVR.

1 Introduction

The rapid advance of large language models (LLMs) has been propelled by increasingly sophisticated post-training strategies, from supervised instruction tuning to reinforcement learning from human feedback [Schulman et al., 2017, Jaech et al., 2024, Liu et al., 2024]. More recently, Reinforcement Learning from Verifiable Rewards (RLVR) has emerged as a powerful paradigm for domains where solutions are unknown but correctness is programmatically checkable—such as formal mathematics and competitive programming [Shao et al., 2024, Guo et al., 2025]. In the RLVR pipeline, the model simultaneously acts as generator and learner: it samples candidate reasoning paths, receives binary correctness feedback from a deterministic verifier, and refines its policy accordingly, enabling continuous self-evolution beyond human-provided examples.

A defining feature of modern RLVR algorithms is group-based rollout: for each training prompt, the policy generates a group of responses, and the reward signal is constructed from the correctness labels of individual outputs within the group. However, current methods treat these labels in isolation, using only the binary correct/incorrect distinction per sample to compute advantages [Shao et al., 2024, Yu et al., 2025]. The collective structure of the group output—in particular, the distribution of errors

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

across the incorrect responses—is entirely discarded. We argue that this represents a significant, overlooked information source.

Our empirical analysis (§2) reveals that the **diversity of wrong answers within a rollout group** is a strong predictor of whether RLVR training will improve the model's performance on that problem—problems with diverse errors exhibit substantially higher improvement rates, whereas those with homogeneous failures see limited gains. Motivated by this observation, we propose **Error Diversity Advantage Shaping (EDAS)**, a lightweight, algorithm-agnostic technique that reshapes the advantage signal for incorrect rollouts based on intra-group error diversity: it amplifies penalties for dominant, repeated errors and attenuates penalties for rare, exploratory ones, thereby encouraging broader exploration and discouraging error perseveration. Crucially, EDAS operates as a simple post-hoc advantage adjustment that can be applied on top of any group-relative RLVR method without modifying the core training algorithm.

AIME 2026 #21 Find the sum of all r such that the circle of radius r centered at $(4, 39)$ is tangent to $2y = x^2 - 8x + 12$. **Ground Truth: 50**

Shared setup. All 32 samples define the squared-distance function $f(x)$ and find two critical values: f=81 (at x=4±4$\sqrt{5}$) and f=1681 (at x=4), corresponding to tangency radii r=9 and r=41. The correct answer requires recognizing *both* as valid: $9+41 = 50$. Samples diverge at how they interpret these critical values:

Answer = 9 (9/32) — Keeps only r=9, discards r=41. Treats the global minimum as the *only* tangency.

*"The **minimum value** of $f(x)$ is 81 ... So $r = \sqrt{81} = \boxed{9}$."*

Answer = 41 (4/32) — Keeps only r=41, discards r=9. Misidentifies $f(4)$=1681 as the global minimum.

"The minimum value of $f(x)$ is 1681 ... the only tangent radius is r=$\boxed{41}$."

Answer = 0 (6/32) — Identifies only *one* radius (either 9 or 41) but allows $r < 0$, so $+r$ and $-r$ cancel to 0.

"Since r^2=81 implies $r = \pm 9$ ... sum $= 9 + (-9) = \boxed{0}$."

Figure 1: Case study on Qwen3-8B. Samples producing the same wrong answer follow near-identical reasoning paths.

We validate EDAS on mathematical reasoning and code generation tasks, integrating it into both GRPO [Shao et al., 2024] and DAPO [Yu et al., 2025] and comparing against several baselines. EDAS yields substantial average gains, with an average improvement of 6.29 points over DAPO on Qwen3-8B across seven math benchmarks and 1.45 points on code generation. The key contributions of this work are as follows:

- **Empirical Insight into Error Diversity:** We identify intra-group error distribution as a critical, yet previously underexplored, determinant of training efficacy in RLVR. Our analysis reveals that environments eliciting diverse reasoning failures yield substantially richer learning signals, whereas homogeneous error distributions severely bottleneck policy optimization.
- **Error Diversity Advantage Shaping:** We propose a lightweight and algorithm-agnostic framework that dynamically calibrates advantage signals based on information-theoretic error diversity. By amplifying penalties for dominant, entrenched failures and attenuating them for rare errors, EDAS discourages error perseveration and incentivizes the exploration of alternative reasoning trajectories.
- **Comprehensive Empirical Validation:** Extensive evaluations demonstrate the robustness of EDAS across models of different scales and two foundational RL algorithms on math reasoning and code generation tasks. Notably, our method establishes positive average gains across all evaluated settings, yielding an average absolute improvement of 6.29 points on math reasoning and 1.45 points on code generation over the SOTA baseline.

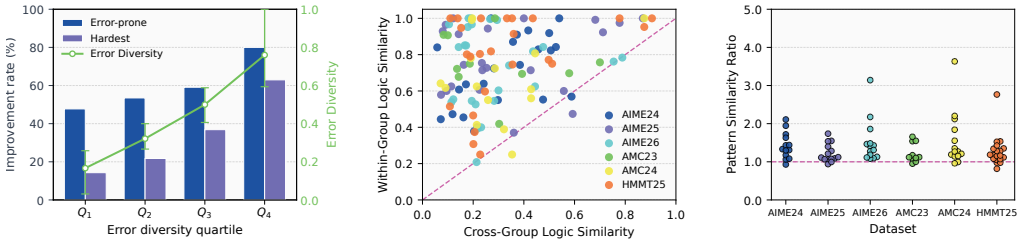

Figure 2: **(a) Error diversity predicts RLVR effectiveness.** We find that Improvement Rate—the ratio of problems showing increased Average Pass Rate (APR)—positively correlates with initial error-set diversity. We collect the error-prone set (initial APR < 100%) and the hardest subsets (initial APR = 0%), and partition them equally into four quartiles according to error diversity. The results suggest that higher error diversity provides a richer learning signal for RLVR. See Appendix C for details. **(b) Shared wrong answers indicate shared reasoning roots.** We compare the probability of sharing a fundamental logical error between trajectories yielding the same wrong answer (y-axis) versus different wrong answers (x-axis). With 94% of evaluated problems lying above the $y = x$ diagonal, the results demonstrate that identical incorrect answers predominantly stem from similar reasoning paths, whereas differing wrong answers reflect distinct logical failures. See Appendix D for details. **(c) Shared wrong answers indicate shared language pattern.** The ratio is calculated as within-group similarity divided by between-group similarity for each problem. Points above the dashed line indicate that models generate more consistent 3-gram language pattern when producing the same incorrect answer. This systematic trend persists across all tested AIME, AMC, and HMMT benchmarks.

2 Motivation

In this section, we present empirical evidence that the diversity of incorrect answers within a rollout group is a strong predictor of RLVR training effectiveness, and validate that distinct wrong answers genuinely reflect distinct underlying reasoning paths.

2.1 Error Diversity Predicts RLVR Effectiveness

Our empirical investigation reveals a striking pattern (Figure 2a): the **diversity of wrong answers within a rollout group** is a strong predictor of whether RLVR training will improve the model's performance on that problem—problems with diverse errors exhibit substantially higher improvement rates, whereas those with homogeneous failures see limited gains.

This observation rests on a key premise: that distinct wrong answers genuinely reflect distinct underlying reasoning paths, rather than mere surface variation. We provide converging evidence from two independent angles. Using an LLM judge to assess logical roots, Figure 2b shows that trajectories yielding the same wrong answer predominantly share the same fundamental reasoning error, while those yielding different wrong answers reflect distinct logical failures. This is independently confirmed at the language pattern level: Figure 2c shows trajectories with the same incorrect answer exhibit significantly higher 3-gram similarity in their reasoning steps than those with different answers. A detailed case in Figure 1 further establishes that wrong answer diversity is a reliable proxy for reasoning-path diversity, explaining why it predicts RLVR effectiveness and motivating the exploitation of intra-group error structure.

2.2 From Observation to Method

The above findings suggest that the intra-group error distribution carries rich, actionable information that current RLVR algorithms discard. A natural strategy is to reshape the advantage signal so that dominant, repeated errors receive amplified penalties—pushing the model away from entrenched failure modes—while rare, exploratory errors receive attenuated penalties, preserving the model's diverse reasoning attempts. We formalize this intuition as **Error Diversity Advantage Shaping (EDAS)** in the next section.

3 Method

In this section, we describe Error Diversity Advantage Shaping (EDAS), a lightweight advantage adjustment module that can be applied on top of any group-relative RLVR algorithm. EDAS modifies

Algorithm 1 EDAS Advantage Redistribution Subroutine (ARS)

Require: Incorrect trajectories subset $\mathcal{W}$, Baseline advantages $\{A_i^{\text{orig}}\}_{i\in\mathcal{W}}$.
Require: Hyperparameters: α (diversity), β (collapse penalty).
Ensure: Advantage adjustments $\{\Delta_i\}_{i\in\mathcal{W}}$.

$N_w \leftarrow |\mathcal{W}|$
if $N_w \leq 1$ **then**
  return $\Delta_i = 0 \quad \forall i \in \mathcal{W}$ ▷ Branch A: Insufficient samples
end if
Compute dynamic scale $S \leftarrow \frac{1}{N_w}\sum_{i\in\mathcal{W}} |A_i^{\text{orig}}|$
Partition $\mathcal{W}$ into K equivalence classes $\mathcal{C} = \{C_1, \ldots, C_K\}$
if $K = 1$ **then**
  return $\Delta_i = -\beta \cdot S \quad \forall i \in \mathcal{W}$ ▷ Branch B: Error perseveration penalty
else
  Compute empirical probabilities $p_k \leftarrow |C_k|/N_w$ for $k = 1, \ldots, K$
  Compute subset entropy $H \leftarrow -\sum_{k=1}^{K} p_k \ln p_k$
  for each error trajectory $i \in \mathcal{W}$ **do** ▷ Branch C: Healthy diversity
    Find class C_k such that $i \in C_k$
    Compute self-information $I_i \leftarrow -\ln p_k$
    Calculate normalized relative surprisal $T_i \leftarrow (I_i - H)/\ln N_w$
    $\Delta_i \leftarrow \alpha \cdot S \cdot T_i$
  end for
  return $\{\Delta_i\}_{i\in\mathcal{W}}$
end if

only the advantage values assigned to incorrect trajectories; correct trajectories and the rest of the training pipeline remain untouched.

3.1 Intra-Group Error Partitioning

Let $\mathcal{W} \subseteq \{1, \ldots, N\}$ denote the index set of trajectories that yield incorrect answers, with $N_w = |\mathcal{W}|$. Rather than committing to a fixed notion of error similarity, EDAS is built around a **domain-specific error equivalence function** $\mathcal{E} : \mathcal{Y} \to \mathcal{L}$, which maps each trajectory $y_i \in \mathcal{Y}$ to a discrete *error label* $\ell_i = \mathcal{E}(y_i)$ drawn from a label set $\mathcal{L}$. Two incorrect trajectories are treated as belonging to the same error class if and only if they share the same label:

$$y_i \sim y_j \iff \mathcal{E}(y_i) = \mathcal{E}(y_j).$$

The choice of $\mathcal{E}$ is the sole domain-specific design decision in EDAS; all downstream computation—entropy estimation, advantage redistribution, and clipping—is identical regardless of domain. Concretely, we describe the designs for the two most common reasoning scenarios:

- **Mathematical reasoning.** $\mathcal{E}(y_i)$ extracts the canonicalized numerical answer from the `\boxed{}` environment and applies symbolic normalization via the Math-Ruler library. Two trajectories are co-classified only if their extracted values are provably numerically equivalent. We show in Appendix H that embedding-based alternatives [Chen et al., 2025, Li et al., 2026, Wei and Huang, 2026] are fundamentally unsuitable for this purpose.
- **Code generation.** $\mathcal{E}(y_i)$ maps each incorrect trajectory to its Python exception type (e.g., `SyntaxError`, `TypeError`, `WrongAnswer`), obtained by compiling and executing the generated code in a sandboxed environment against test cases (detailed in §4.3). Exception types provide a natural semantic proxy for the underlying algorithmic failure mode.

This relation partitions the set of incorrect trajectories into K disjoint equivalence classes (i.e., unique error types), denoted as $\mathcal{C} = \{C_1, C_2, \ldots, C_K\}$, such that $\sum_{k=1}^{K} |C_k| = N_w$. For each class C_k, we define its empirical intra-group probability as $p_k = |C_k|/N_w$; for any trajectory $i \in C_k$, we use p_k as its class probability.

To ensure that EDAS adjustments scale proportionally with the current learning dynamics—avoiding over-adjustment when advantages are small (e.g., early training) or under-adjustment when they are large—we define a dynamic scaling factor S as the mean absolute advantage of the incorrect subset:

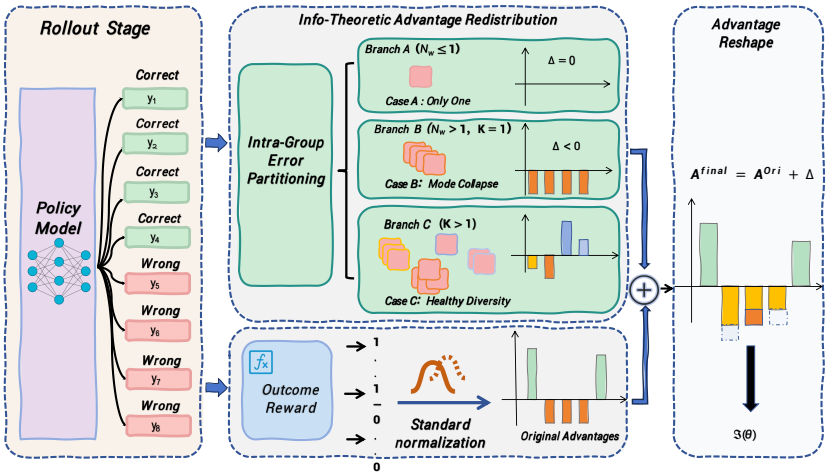

Figure 3: Overview of the proposed method.

$$S = \frac{1}{N_w} \sum_{i \in \mathcal{W}} \left| A_i^{\text{orig}} \right| \tag{1}$$

Here A_i^{orig} denotes the scalar base advantage assigned to rollout i by the underlying group-relative algorithm, which is subsequently broadcast to all valid tokens of that rollout; under binary outcome rewards, all incorrect rollouts within a group share the same base advantage, so S serves to calibrate the adjustment to the current advantage scale.

3.2 Information-Theoretic Advantage Redistribution

The core idea of EDAS is to penalize homogeneous failure modes while attenuating penalties for diverse, exploratory errors. We formalize this using the self-information of each trajectory $I_i = -\ln p_k$ where C_k is the class containing y_i and the empirical Shannon entropy of the group's error distribution $H = -\sum_{k=1}^{K} \frac{|C_k|}{N_w} \ln\left(\frac{|C_k|}{N_w}\right)$.

Depending on the cardinality of the error partition K, we formulate the advantage adjustment Δ_i for each $i \in \mathcal{W}$ as a piecewise function:

$$\Delta_i = \begin{cases} 0, & \text{if } N_w \leq 1 \\ -\beta \cdot S, & \text{if } N_w > 1 \text{ and } K = 1 \\ \alpha \cdot S \cdot \left(\frac{I_i - H}{\ln N_w}\right), & \text{if } K > 1 \end{cases} \tag{2}$$

Here, $\alpha, \beta > 0$ control the strength of diversity encouragement and collapse penalty.

The three branches capture distinct regimes:

- **Insufficient Statistics** ($N_w \leq 1$): Too few errors to estimate diversity; the original advantage is preserved.
- **Error Perseveration** ($K = 1$): All wrong answers are identical—the policy concentrates all errors on a single answer. EDAS applies a uniform negative shift $-\beta \cdot S$ to all incorrect trajectories, amplifying their negative advantage. This increases the magnitude of log-probability decrease for these tokens during the policy gradient update, accelerating the model's departure from the entrenched error and freeing probability mass for alternative reasoning paths.
- **Healthy Diversity** ($K > 1$): Multiple error types coexist. EDAS redistributes penalties based on normalized relative surprisal $T_i = (I_i - H)/\ln N_w$, which is bounded in $[-1, 1]$

Table 1: **Main Results on Math Benchmarks.** Comparison of our **EDAS** enhancement across three Qwen3 models. Δ denotes the improvement of our best variant (**EDAS w/ DAPO**) over the strongest existing baseline. The best results are **bolded**, and our methods are highlighted in light red. Due to the significant increase in inference demands resulting from the use of Dynamic Sampling, we evaluate DAPO as a separate comparison.

Method	AMC23	AMC24	AIME24	AIME25	AIME26	HMMT	Olympiad	Avg.
Qwen3-8B								
Baseline	67.03	44.24	23.85	20.00	15.10	10.42	54.75	33.63
+ GRPO	68.67	48.75	28.85	23.75	18.43	11.25	57.57	36.75
+ PKPO	71.95	45.07	**29.16**	20.00	18.85	12.08	58.90	36.57
+ **GRPO w/ EDAS**	**73.52**	**49.38**	27.81	**26.98**	**19.06**	**12.50**	**59.20**	**38.35**
Δ	+1.57	+0.63	-1.35	+3.23	+0.21	+0.42	+0.30	+1.60
+ DAPO	84.45	56.88	41.46	38.12	39.17	15.94	58.75	47.82
+ DAPO w/ En Adv	81.02	55.62	31.81	38.23	33.23	15.21	58.31	44.78
+ **DAPO w/ EDAS**	**90.39**	**59.38**	**48.85**	**45.52**	**48.65**	**20.10**	**65.88**	**54.11**
Δ	+5.94	+2.50	+7.39	+7.29	+9.48	+4.16	+7.13	+6.29
Qwen3-4B								
Baseline	67.03	44.31	22.08	20.01	18.02	10.83	53.41	33.68
+ DAPO	85.7	57.15	**47.50**	43.54	40.73	22.40	65.88	51.84
+ DAPO w/ En Adv	77.27	49.58	34.58	31.87	26.77	13.65	59.35	41.87
+ **DAPO w/ EDAS**	**88.59**	**58.19**	46.46	**48.12**	**44.27**	**23.12**	**67.36**	**53.73**
Δ	+2.89	+1.04	-1.04	+4.58	+3.54	+0.72	+1.48	+1.89
Qwen3-4B-Base								
Baseline	31.56	12.08	10.31	5.52	2.6	1.56	26.85	12.93
+ DAPO	60.00	33.68	12.81	18.02	10.73	8.33	46.44	27.14
+ DAPO w/ En Adv	55.08	34.24	16.98	17.40	12.71	6.67	45.25	26.90
+ **DAPO w/ EDAS**	**72.73**	**48.68**	**24.17**	**25.21**	**20.97**	**10.94**	**56.5**	**37.02**
Δ	+12.73	+14.44	+7.19	+7.19	+8.26	+2.61	+10.06	+9.88

(see Appendix A). Frequent errors ($T_i < 0$) receive amplified penalties; rare errors ($T_i > 0$) receive attenuated penalties.

In the healthy diversity regime, this redistribution satisfies a strict zero-sum constraint: $\sum_{i \in \mathcal{W}} T_i = 0$ as proved in Appendix A. Consequently, EDAS reallocates the advantage signals without altering their arithmetic mean, preserving the unbiased nature of the base estimator. Since all incorrect rollouts within a group share the same base advantage under binary outcome rewards, this branch is more precisely characterized as structured variance injection among otherwise identical incorrect advantages than as reallocation of pre-existing spread. We summarize the advantage redistribution of EDAS in Algorithm 1.

3.3 Monotonicity-Preserving Clipping

To prevent a positive adjustment from inverting the sign of A_i^{orig}, which would falsely reward an incorrect trajectory, we clip the adjustment magnitude to $|A_i^{\text{orig}}|/\kappa$ for a margin $\kappa > 1$:

$$A_i^{\text{final}} = A_i^{\text{orig}} + \text{sgn}(\Delta_i) \cdot \min\left(|\Delta_i|, \frac{|A_i^{\text{orig}}|}{\kappa}\right) \tag{3}$$

This clipping mechanism guarantees no sign inversion of the advantage for incorrect trajectories, ensuring they always receive non-positive gradient updates, with the full formal proof provided in Appendix B.

4 Experiment

In this section, we evaluate the effectiveness of EDAS across two challenging domains: mathematical reasoning and code generation. We aim to examine whether EDAS consistently improves RLVR performance across different model scales and task types.

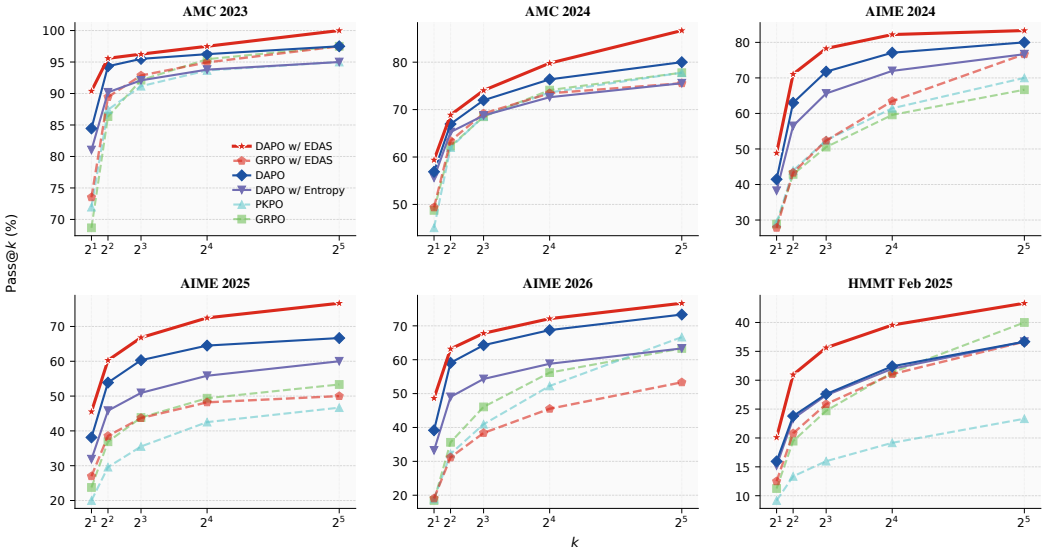

Figure 4: Pass@k curves on six benchmarks for Qwen3-8B. EDAS w/ DAPO consistently achieves the highest Pass@k across all values of k. The advantage is especially pronounced at larger k on harder benchmarks, where diversity in reasoning paths directly translates to higher coverage of correct solutions.

4.1 Experimental Setup

Model. To evaluate the scalability and robustness of EDAS, we employ the Qwen3 series Yang et al. [2025] as our base models, including Qwen3-8B, Qwen3-4B, and Qwen3-4B-Base. These models represent a diverse range of pre-existing reasoning proficiencies and parameter scales.

Implementation Details. We evaluate our method on mathematical reasoning and code generation tasks. Experiments were executed on a system utilizing 8× NVIDIA H200 GPUs. All models are trained under a standard RLVR framework where the reward is derived solely from the correctness of the final output. We maintain a consistent rollout configuration across both domains, utilizing a rollout batch size of 256 with 10 samples per prompt. The EDAS hyperparameters are fixed at $\alpha = 0.4, \beta = 0.2, \kappa = 2.0$ for all experiments; a sensitivity analysis and branch ablations are provided in Appendix N. For the mathematical reasoning tasks, we evaluate performance across seven mainstream benchmarks, including AIME (2024, 2025, 2026) [MAA, 2024, 2025, 2026], AMC (2023, 2024) [MAA, 2023], HMMT 2025 [Balunovic et al., 2025], and OlympiadBench [He et al., 2024]. To assess code generation capabilities, we utilize four representative benchmarks: LiveCodeBench [Jain et al., 2024], Codeforces [Quan et al., 2025], HumanEval+ [Chen et al., 2021, Liu et al., 2023], and MBPP+ [Austin et al., 2021, Liu et al., 2023]. Detailed task-specific configurations, including dataset statistics, training epochs, and evaluation settings, are provided in Appendix F.

Baselines. We compare EDAS against the standard RLVR algorithm GRPO [Shao et al., 2024] and its comparable variant DAPO [Yu et al., 2025]. We also include PKPO [Walder and Karkhanis, 2025] as a baseline, which attempts to directly optimize the Pass@k metric by theoretically leveraging intra-group information inherent in RLVR. Furthermore, we compare our method with entropy-based [Cheng et al., 2026] regularization techniques (En Adv) that promote model exploration by augmenting the reward signal. Notably, as entropy-based exploration encountered training instability on larger models, we report its peak performance sampled at 10-step intervals to ensure a fair comparison. Detailed hyperparameter configurations for these additional baselines are provided in Appendix G.

4.2 Math Reasoning

Results. As illustrated in Table 1, applying EDAS consistently improves average reasoning performance across all three models when combined with mainstream reinforcement learning algorithms. Since DAPO employs Dynamic Sampling—which filters out samples where the model is entirely

correct or entirely incorrect and triggers re-sampling—it consumes significantly more computational resources for the same number of steps. Dynamic Sampling also plays a critical role in EDAS, as discussed in Appendix B. Consequently, we conduct separate comparisons based on the use of Dynamic Sampling. Over the strongest baseline (DAPO), EDAS achieves average improvements of **6.29** points on Qwen3-8B, **1.89** points on Qwen3-4B, and **9.88** points on Qwen3-4B-Base across seven benchmarks, demonstrating the substantial effectiveness of leveraging error diversity information in group rollouts. A one-sided paired bootstrap test over the repeated evaluation samples further confirms that, in the main Qwen3-8B DAPO setting, the gains are statistically significant on six of the seven benchmarks and the average improvement is significant at $p < 10^{-4}$ (see Appendix L); the average gain is also reproduced under three independent training seeds, with all 21 seed–benchmark pairs positive (Appendix M). A preliminary replication on a non-Qwen model family (Gemma-3-4B-IT) is reported in Appendix O.

Pass@k Results. Beyond average accuracy, we evaluate Pass@k to measure the model's coverage of correct solutions across multiple attempts. Figure 4 reports Pass@k for $k \in \{2, 4, 8, 16, 32\}$ on six benchmarks using Qwen3-8B. EDAS w/ DAPO consistently dominates all baselines across all values of k, confirming that the improvement is not limited to greedy decoding but extends to the model's ability to produce at least one correct solution among k attempts. Notably, the gap between EDAS and DAPO widens as k increases on harder benchmarks (AIME 2025, AIME 2026, HMMT), suggesting that EDAS's diversity-promoting mechanism enriches the pool of distinct reasoning paths, making it increasingly likely that at least one succeeds. EDAS w/ GRPO similarly outperforms both GRPO and PKPO, further confirming the generality of the approach. Pass@4 as a function of training step is reported in Appendix K, showing that the coverage advantage emerges during training rather than only at convergence.

4.3 Code Generation

To demonstrate that EDAS generalizes beyond mathematical reasoning, we evaluate it on code generation tasks. The key adaptation lies in the definition of the equivalence relation for error partitioning: instead of grouping incorrect responses by their final numerical answer, we group them by Python exception type, the implementation details of which are provided in Appendix E.

Results. As shown in Table 2, EDAS consistently improves over DAPO across all four benchmarks, with an average gain of **1.45** points. The improvement is most pronounced on Codeforces, which contains algorithmically challenging problems where diverse error exploration is particularly valuable—a model stuck on a single algorithmic approach benefits from EDAS pushing it toward alternative strategies that may yield different failure modes but also open paths to correct solutions.

These results confirm two important properties of EDAS: (1) the core principle—that error diversity is an informative signal for RLVR—extends beyond mathematical reasoning to code generation, and (2) the framework is flexible enough to accommodate domain-specific definitions of error equivalence while preserving the same advantage redistribution mechanism.

Table 2: Code generation results on Qwen3-4B. Δ denotes the improvement of EDAS over DAPO.

Method	LiveCodeBench	Codeforces	HumanEval+	MBPP+	Avg.
+ DAPO	31.21	42.42	79.88	63.23	54.19
+ DAPO w/ EDAS	**32.07**	**45.97**	**80.49**	**64.02**	**55.64**
Δ	+0.86	+3.55	+0.61	+0.79	+1.45

5 Analysis

EDAS Promotes Error Diversity and Enhances RLVR Performance. Figure 5a shows that integrating EDAS into DAPO sustains a higher variety of unique wrong answers during training. Furthermore, EDAS consistently achieves superior reward levels as shown in Figure 5b. These dynamics indicate that EDAS prevents premature convergence to narrow failure modes, encouraging the exploration of diverse reasoning paths and ultimately boosting overall RLVR efficacy.

EDAS Facilitates Breakthroughs on Capability-Frontier Problems. EDAS further demonstrates its utility on capability-frontier problems. Solving these instances requires the discovery of genuinely novel reasoning paths through RL. Table 3 summarizes the "breakthrough" rates for Qwen3-8B; across four benchmarks, 64 problems are identified as hard. EDAS successfully resolves 30 of these, compared to 22 for DAPO, representing a 36% relative improvement. Notably, EDAS uniquely solves 10 problems that DAPO fails to crack entirely (APR = 0%). Detailed analysis is provided in Appendix I.

Table 3: **Bottleneck-breaking analysis on Qwen3-8B.** *Hard* denotes problems where the base model achieved zero accuracy over 32 rollouts. *Breakthrough* indicates the number of such problems solved (APR > 0) after RL training. *EDAS-only* highlights the unique capacity of our method to solve instances that remains unsolved by DAPO.

Benchmark	Hard (Total)	DAPO Break	EDAS Break	EDAS-only
Qwen3-8B (Zero-Shot Hard Subset)				
AIME 2025	15	6	**9**	3
AIME 2026	15	7	**8**	1
AMC 2024	13	5	**7**	3
HMMT Feb 25	21	4	**6**	3
Total Sum	**64**	**22**	**30**	**10**
Success Rate	0.0%	34.4%	**46.9%**	+12.5%

6 Related Work

Group-Level Information in Rollouts. Recent RLVR research increasingly exploits the structured information within rollout groups. PKPO [Walder and Karkhanis, 2025] optimizes Pass@k coverage via low-variance gradient estimators, while RSPO [Zhang et al., 2025a] and Harder-is-Better [Dai et al., 2026] reweight gradients based on trajectory rarity or problem difficulty. To enhance set-level dynamics, SetPO [Li et al., 2026] introduces marginal contribution bonuses, and NGRPO [Nan et al., 2025] addresses gradient vanishing in homogeneous failed groups by injecting negative signals. Distinct from these, EDAS is motivated by a novel empirical observation: the diversity of incorrect answers within a group is a strong predictor of training success, as distinct errors reflect broader exploration across reasoning paths.

Diversity of Reasoning Paths in RLVR. To combat policy entropy collapse, advantage-shaping methods such as UCAS [Xie et al., 2025], EDGE-GRPO [Zhang et al., 2025b], and ProGRPO [Bu et al., 2025] modulate rewards to penalize overconfidence, utilize internal probability structures, or incorporate critique-based feedback [Zhang et al., 2026]. Alternatively, structural approaches intervene via tree-structured branching [Zhao et al., 2026], noise embeddings [Lyu et al., 2025],

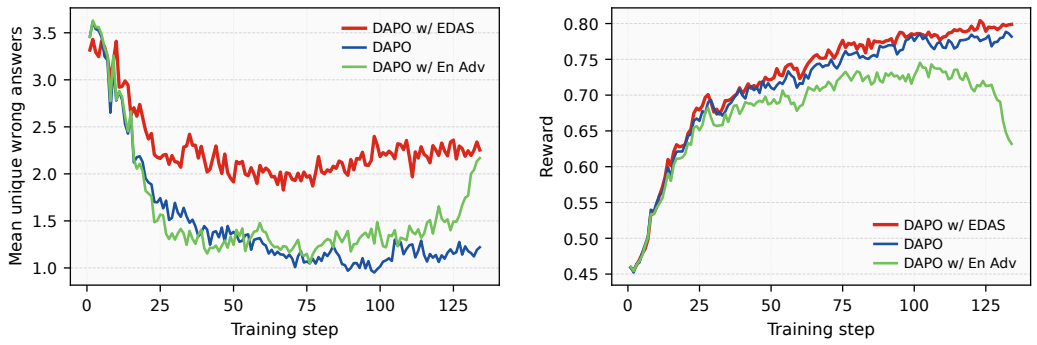

Figure 5: (a) EDAS preserves error diversity throughout training. The trajectories of mean unique wrong answers show that while DAPO's error diversity drops rapidly, applying EDAS maintains a higher variety of failure modes, underpinning enhanced capability to discover successful reasoning paths. (b) EDAS improves optimization dynamics. Reward curves (0.5 exponential smoothing) show that EDAS achieves a higher asymptotic reward and faster growth than DAPO and DAPO w/ Entropy Advantage.

interleaved latent reasoning [Dong et al., 2025], or direct CoT entropy integration [Hao et al., 2025]. Unlike these methods that modify sampling, architectures, or the full distribution, EDAS provides a post-hoc, complementary adjustment. It targets the error structure specifically, using the empirical surprisal of incorrect trajectories to differentially reweight advantages—a lightweight mechanism that leaves all other training components unchanged.

7 Conclusion

Our empirical analysis shows that intra-group error diversity is strongly correlated with RLVR effectiveness, suggesting that diverse failure modes can provide useful learning signals beyond binary correctness. Building on this observation, we introduce Error Diversity Advantage Shaping (EDAS), a lightweight advantage-shaping method that redistributes penalties among incorrect rollouts according to their error frequency. Across multiple Qwen3 variants and two domains (math and code), EDAS improves performance over the corresponding baselines and helps maintain error diversity during training. Promising future directions include incorporating diversity signals from intermediate reasoning steps and extending EDAS to other verifiable domains. We discuss limitations in Appendix P.

References

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021.

Mislav Balunovic, Jasper Dekoninck, Ivo Petrov, Nikola Jovanovic, and Martin Vechev. Matharena: Evaluating llms on uncontaminated math competitions, february 2025. *URL https://matharena. ai*, 8, 2025.

Dake Bu, Wei Huang, Andi Han, Atsushi Nitanda, Bo Xue, Qingfu Zhang, Hau-San Wong, and Taiji Suzuki. Post-training as reweighting: A stochastic view of reasoning trajectories in language models. *arXiv preprint arXiv:2511.07368*, 2025.

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

Xiwen Chen, Wenhui Zhu, Peijie Qiu, Xuanzhao Dong, Hao Wang, Haiyu Wu, Huayu Li, Aristeidis Sotiras, Yalin Wang, and Abolfazl Razi. Dra-grpo: Exploring diversity-aware reward adjustment for r1-zero-like training of large language models. *arXiv preprint arXiv:2505.09655*, 2025.

Daixuan Cheng, Shaohan Huang, Xuekai Zhu, Bo Dai, Xin Zhao, Zhenliang Zhang, and Furu Wei. Reasoning with exploration: An entropy perspective. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 30377–30385, 2026.

Yanqi Dai, Yuxiang Ji, Xiao Zhang, Yong Wang, Xiangxiang Chu, and Zhiwu Lu. Harder is better: Boosting mathematical reasoning via difficulty-aware GRPO and multi-aspect question reformulation. *arXiv preprint arXiv:2601.20614*, 2026.

Shuai Dong, Siyuan Wang, Xingyu Liu, Chenglin Li, Haowen Hou, and Zhongyu Wei. Interleaved latent visual reasoning with selective perceptual modeling. *arXiv preprint arXiv:2512.05665*, 2025.

Shuai Dong, Yongfu Zhu, Yuqi Xu, Weichu Xie, Ziyue Wang, Kaiwen Tuo, Congcong Wang, Siyuan Wang, Wenqi Shao, Shuai Yang, et al. RL starts before RL: On policy distillation for better reinforcement learning. *arXiv preprint arXiv:2609.28145*, 2026.

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

Bingguang Hao, Zengzhuang Xu, Maolin Wang, Yuntao Wen, Yicheng Chen, Cunyin Peng, Long Chen, Dong Wang, Xiangyu Zhao, Jinjie Gu, et al. Reasoning through exploration: A reinforcement learning framework for robust function calling. *arXiv preprint arXiv:2508.05118*, 2025.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, et al. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3828–3850, 2024.

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

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. Livecodebench: Holistic and contamination free evaluation of large language models for code. *arXiv preprint arXiv:2403.07974*, 2024.

Chenyi Li, Yuan Zhang, Bo Wang, Guoqing Ma, Wei Tang, Haoyang Huang, and Nan Duan. Setpo: Set-level policy optimization for diversity-preserving LLM reasoning. *arXiv preprint arXiv:2602.01062*, 2026.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.

Jiawei Liu and Lingming Zhang. Code-r1: Reproducing r1 for code with reliable rewards. `https://github.com/ganler/code-r1`, 2025.

Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. Is your code generated by chatgpt really correct? rigorous evaluation of large language models for code generation. *Advances in neural information processing systems*, 36:21558–21572, 2023.

Bohan Lyu, Qixin Xu, Zihan Zhu, Yesai Wu, Zhong Zhang, Haotian Chen, Xin Cong, Xiaojiang Liu, Zhiyuan Liu, and Maosong Sun. Diversity-aware training for test-time scaling. 2025.

MAA. American mathematics competitions - AMC. `https://maa.org/`, 2023.

MAA. American invitational mathematics examination - AIME 2024. `https://maa.org/`, 2024.

MAA. American invitational mathematics examination - aime. `https://maa.org/`, 2025.

MAA. American invitational mathematics examination-AIME 2026. `https://maa.org/`, 2026.

Gongrui Nan, Siye Chen, Jing Huang, Mengyu Lu, Dexun Wang, Chunmei Xie, Weiqi Xiong, Xianzhou Zeng, Qixuan Zhou, Yadong Li, et al. Ngrpo: Negative-enhanced group relative policy optimization. *arXiv preprint arXiv:2509.18851*, 2025.

Shanghaoran Quan, Jiaxi Yang, Bowen Yu, Bo Zheng, Dayiheng Liu, An Yang, Xuancheng Ren, Bofei Gao, Yibo Miao, Yunlong Feng, et al. Codeelo: Benchmarking competition-level code generation of llms with human-comparable elo ratings. *arXiv preprint arXiv:2501.01257*, 2025.

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

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

Christian Walder and Deep Karkhanis. Pass@k policy optimization: Solving harder reinforcement learning problems. *arXiv preprint arXiv:2505.15201*, 2025.

Kangda Wei and Ruihong Huang. MMR-GRPO: Accelerating GRPO-style training through diversity-aware reward reweighting. *arXiv preprint arXiv:2601.09085*, 2026.

Can Xie, Ruotong Pan, Xiangyu Wu, Yunfei Zhang, Jiayi Fu, Tingting Gao, and Guorui Zhou. Unlocking exploration in rlvr: Uncertainty-aware advantage shaping for deeper reasoning. *arXiv preprint arXiv:2510.10649*, 2025.

Yuqi Xu, Rizhen Hu, Zihan Liu, Mou Sun, and Kun Yuan. Grouter: Decoupling routing from representation for accelerated MoE training. *arXiv preprint arXiv:2603.06626*, 2026.

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

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale, 2025. *URL https://arxiv. org/abs/2503.14476*, 1:2, 2025.

Kaichen Zhang, Shenghao Gao, Yuzhong Hong, Haipeng Sun, Junwei Bao, Hongfei Jiang, Yang Song, Hong Dingqian, and Hui Xiong. Rspo: Risk-seeking policy optimization for pass@k and max@k metrics in large language models. *arXiv preprint arXiv:2508.01174*, 2025a.

Xiaoying Zhang, Hao Sun, Yipeng Zhang, Kaituo Feng, Chaochao Lu, Chao Yang, and Helen M Meng. Critique-GRPO: Advancing LLM reasoning with natural language and numerical feedback. In *International Conference on Machine Learning*, 2026.

Xingjian Zhang, Siwei Wen, Wenjun Wu, and Lei Huang. Edge-grpo: Entropy-driven grpo with guided error correction for advantage diversity. *arXiv preprint arXiv:2507.21848*, 2025b.

Ziqi Zhao, Zhaochun Ren, Jiahong Zou, Liu Yang, Zhiwei Xu, Xuri Ge, Zhumin Chen, Xinyu Ma, Daiting Shi, Shuaiqiang Wang, et al. Reinforced efficient reasoning via semantically diverse exploration. *arXiv preprint arXiv:2601.05053*, 2026.

A Proof of Zero-Sum Property in the Diverse Regime

Theorem 1 (Zero-Sum Advantage Redistribution)**.** *In the healthy diversity regime ($K > 1$), the proposed Error Diversity Advantage Shaping (EDAS) guarantees that the arithmetic mean of the normalized relative surprisal T_i over all incorrect trajectories is exactly zero, i.e., $\frac{1}{N_w}\sum_{i\in\mathcal{W}} T_i = 0$.*

Proof. Recall the definition of the normalized relative surprisal for the i-th incorrect trajectory:

$$T_i = \frac{I_i - H}{\ln N_w} \tag{4}$$

where $I_i = -\ln p_k$ is the self-information of trajectory i with C_k denoting the class containing y_i, and H is the empirical Shannon entropy of the group's error distribution. We are interested in the expected adjustment across the subset of incorrect trajectories $\mathcal{W}$, with $|\mathcal{W}| = N_w$.

By substituting T_i into the summation, we have:

$$\frac{1}{N_w}\sum_{i\in\mathcal{W}} T_i = \frac{1}{N_w \ln N_w}\sum_{i\in\mathcal{W}} (I_i - H) \tag{5}$$

To evaluate $\sum_{i\in\mathcal{W}} I_i$, we can group the individual trajectories by their canonical equivalence classes $\mathcal{C} = \{C_1, C_2, \ldots, C_K\}$. For any trajectory $i \in C_k$, its empirical probability is defined as $p_k = \frac{|C_k|}{N_w}$. Summing the self-information over all incorrect trajectories is equivalent to summing over the equivalence classes weighted by their frequencies $|C_k|$:

$$\sum_{i\in\mathcal{W}} I_i = \sum_{k=1}^{K} |C_k| \cdot \left(-\ln \frac{|C_k|}{N_w}\right) \tag{6}$$

Substituting $|C_k| = N_w \cdot p_k$ into the equation yields:

$$\sum_{i\in\mathcal{W}} I_i = \sum_{k=1}^{K} N_w \cdot p_k(-\ln p_k) = N_w \sum_{k=1}^{K} -p_k \ln p_k \tag{7}$$

By the definition of the empirical Shannon entropy H, the summation term $\sum_{k=1}^{K} -p_k \ln p_k$ is exactly H. Thus, we obtain:

$$\sum_{i \in \mathcal{W}} I_i = N_w H \tag{8}$$

Substituting this back into our original equation:

$$\frac{1}{N_w} \sum_{i \in \mathcal{W}} (I_i - H) = \frac{N_w H - N_w H}{N_w} = 0 \tag{9}$$

Hence, $\frac{1}{N_w} \sum_{i \in \mathcal{W}} T_i = 0$. □

This property guarantees that the EDAS adjustment $\Delta_i \propto T_i$ operates strictly as an internal redistribution of advantage within the incorrect subset. It reshapes the gradient landscape to favor minoritarian hypotheses (exploration) and suppress majoritarian traps (error perseveration), without shifting the global baseline of the base RL algorithm.

Remark: Effect of κ-Clipping on the Zero-Sum Property. It is important to note that the zero-sum constraint $\sum_{i \in \mathcal{W}} T_i = 0$ holds exactly *before* the monotonicity-preserving κ-clipping step described in Section 3.3. In practice, the clipping operation

$$\Delta_{\text{final},i} = \text{sgn}(\Delta_i) \cdot \min\Big(|\Delta_i|,\ \tfrac{|A_i^{\text{orig}}|}{\kappa}\Big)$$

may truncate some adjustments asymmetrically—for instance, when a frequent-error trajectory has a very small original $|A_i^{\text{orig}}|$, its penalty term may be truncated. This introduces a *slight imbalance* in the redistributed advantages: the effective sum $\sum_{i \in \mathcal{W}} \Delta_{\text{final},i}$ may deviate marginally from zero.

We emphasize that this deviation is an intentional **engineering trade-off** rather than a theoretical deficiency. The κ-clipping serves a critical purpose: it prevents any single adjustment from inverting the sign of an incorrect trajectory's advantage, which would spuriously reward a wrong answer and destabilize policy updates. In practice, the imbalance induced by clipping is negligibly small—the clipping constraint is rarely active for the majority of trajectories—and does not meaningfully compromise the unbiasedness of the base estimator. The zero-sum property thus characterizes the *ideal* behavior of the redistribution mechanism; clipping constrains that redistribution only at extreme values where unconstrained adjustments would be harmful to training stability.

B Proof of the Sign Non-Inversion Property of Monotonicity-Preserving Clipping

For any incorrect trajectory, the base algorithm guarantees $A_i^{\text{orig}} < 0$. Given clipping margin $\kappa > 1$, the final advantage is

$$A_i^{\text{final}} = A_i^{\text{orig}} + \text{sgn}(\Delta_i) \cdot \min\left(|\Delta_i|,\ \frac{|A_i^{\text{orig}}|}{\kappa}\right).$$

Theorem 2 (Sign Non-Inversion)**.** *For any incorrect trajectory with $A_i^{\text{orig}} < 0$ and $\kappa > 1$,*

1. $A_i^{\text{final}} < 0$ *(no sign inversion),*
2. $|A_i^{\text{final}}| \geq |A_i^{\text{orig}}|\big(1 - \frac{1}{\kappa}\big) > 0$ *(strict lower bound on magnitude).*

Proof. Let $M = \min\Big(|\Delta_i|,\ \frac{|A_i^{\text{orig}}|}{\kappa}\Big) \geq 0$.

Case 1: $\Delta_i \leq 0$ (penalty amplification). $A_i^{\text{final}} = A_i^{\text{orig}} - M \leq A_i^{\text{orig}} < 0$, so (1) holds trivially. Moreover, $|A_i^{\text{final}}| = |A_i^{\text{orig}}| + M \geq |A_i^{\text{orig}}| > |A_i^{\text{orig}}|(1 - \frac{1}{\kappa})$, so (2) holds.

Case 2: $\Delta_i > 0$ **(penalty attenuation).** $A_i^{\text{final}} = A_i^{\text{orig}} + M$. Since $M \leq \frac{|A_i^{\text{orig}}|}{\kappa} < |A_i^{\text{orig}}|$ (as $\kappa > 1$), the adjustment cannot overcome the negative base, so

$$A_i^{\text{final}} = A_i^{\text{orig}} + M < A_i^{\text{orig}} + |A_i^{\text{orig}}| = 0,$$

establishing (1). For the magnitude bound, $|A_i^{\text{final}}| = |A_i^{\text{orig}}| - M \geq |A_i^{\text{orig}}| - \frac{|A_i^{\text{orig}}|}{\kappa} = |A_i^{\text{orig}}|(1 - \frac{1}{\kappa}) > 0$, establishing (2). □

Hence, the clipping mechanism ensures that incorrect trajectories always carry a strictly negative advantage, preventing any spurious reward while bounding the residual gradient magnitude away from zero. In practice, while the clipping margin κ safeguards the advantage sign, we recommend enabling Dynamic Sampling [Yu et al., 2025] when applying EDAS. This is particularly crucial in edge cases where all sampled sequences are incorrect, as it avoids potential scenarios where incorrect sequences might otherwise receive positive reward signals, ensuring stable policy updates across all group configurations.

C Correlation between Error Diversity and RLVR Gains

In this section, we give a detailed description of Figure 2a. To investigate the relationship between the baseline model's failure modes and the effectiveness of RLVR, we conduct a fine-grained analysis on Qwen3-8B-Instruct across six mainstream mathematical benchmarks: AIME (2024–2026), AMC (2023–2024), and HMMT Feb 2025.

Experimental Setup: We apply the state-of-the-art RLVR algorithm, DAPO, training the model on the DAPO-MATH-17K dataset for two epochs. We then analyze the performance shifts by comparing the checkpoints before and after the RLVR process.

Metrics: We define error diversity as the ratio of unique incorrect responses to the total number of incorrect trials during 32-sample repeated testing. The primary evaluation metric is the Improvement Rate, which tracks the portion of problems where the RLVR-tuned model achieves a higher Average Pass Rate (APR) compared to its predecessor.

Grouping: Problems are partitioned into four equal quartiles (Q_1 to Q_4) based on their diversity index. We specifically analyze two cohorts: (1) Error-prone problems, where the base model achieved an $APR < 100\%$; and (2) Hardest Subsets, representing the most challenging cases where the base model's initial APR was exactly 0%.

Observations: As shown in Figure 2a, there is a consistent upward trend in improvement across both cohorts as diversity increases. Notably, for problems in Q_4 (highest diversity), the improvement rate reaches approximately 80%. The error bars in the plot further illustrate the specific range of diversity values within each quartile for the Error-prone set. This trend indicates that problems with multiple distinct failure modes offer more informative feedback during the RLVR process, leading to more robust policy optimization even on tasks where the model initially showed no signs of success.

D Validation of Final Answer as a Proxy for Reasoning Path Consistency

To investigate the logical origins of model failures, we designed an evaluation pipeline to determine if different reasoning trajectories leading to the same incorrect answer share a fundamental logical root. We collected reasoning samples from six math competition benchmarks: AIME (2024–2026), AMC (2023–2024), and HMMT (Feb 2025), sampling each problem 32 times to generate a diverse set of reasoning paths and final answers.

The experimental pipeline first groups all incorrect trajectories for a given problem by their final answer strings. To analyze the internal consistency of these errors, we enumerate all possible pairs within each wrong-answer group (within-group pairs) and employ a judge model (Gemini 3 Flash Preview) to determine if both trajectories fail due to the same fundamental logical error, regardless of phrasing differences. For comparison, we establish a baseline by selecting a representative trajectory from each distinct wrong-answer group to evaluate pairs that produced different incorrect results (cross-group pairs). We are using the prompt shown below:

JUDGE_PROMPT_TEMPLATE

I will provide a math problem, its ground truth answer, and two incorrect reasoning trajectories (Trajectory A and Trajectory B).

Your task is to determine if both trajectories fail due to the **exact same fundamental logical error or misunderstanding**, or if their root causes of failure are **substantively different**. Ignore differences in phrasing, variable naming, or formatting. Focus solely on the mathematical logic.

Problem: `{problem}`
Ground Truth: `{ground_truth}`
Trajectory A: `{traj_a}`
Trajectory B: `{traj_b}`
Output JSON format:

```
{
  "is_same_error_root": true/false
}
```

The resulting metrics—the within-group same-root rate (y-axis) and the cross-group same-root rate (x-axis)—are calculated as the fraction of same-root judgments within their respective categories. To ensure a robust comparison, we filter for questions that contain at least two distinct wrong-answer groups, with at least one group containing multiple trajectories, resulting in $N = 127$ valid problems. The strong tendency for points to cluster above the diagonal validates our hypothesis: shared wrong answers serve as a reliable proxy for shared reasoning error roots, indicating that the diversity of final answers effectively reflects the diversity of underlying logical failures.

E EDAS Implementation of Code Generation

For each incorrect rollout, the error classification follows a hierarchical process. We first identify compilation failures (e.g., `SyntaxError`), assigning the exception name as the label. Remaining rollouts are executed in a sandbox; those failing test cases are labeled `WrongAnswer`, while those triggering runtime exceptions are categorized by their specific exception types (e.g., `TypeError`, `IndexError`). This taxonomy naturally captures the diversity of model failures: a distribution spanning multiple runtime exceptions indicates a more diverse exploration of the task compared to a model limited to repetitive structural or syntactic errors.

Empirical Distribution of Error Classes. A natural concern with this taxonomy is that the coarse `WrongAnswer` class (syntactically valid code that fails test cases) might dominate the incorrect rollouts and thereby suppress the diversity signal. To quantify this, we analyze the complete training logs of the code run (Appendix F), which contain 212,544 incorrect rollouts spanning 26 distinct error classes. Table 4 reports the distribution: `WrongAnswer` accounts for only 19.7% of incorrect rollouts and is the third-largest class, while runtime and exception errors jointly account for 80.1%.

Table 4: Distribution of error classes over all 212,544 incorrect code rollouts in the training logs.

Error Class	Count	Fraction
`IndexError`	82,469	38.8%
`EOFError`	47,773	22.5%
`WrongAnswer`	41,954	19.7%
`ValueError`	31,383	14.8%
Others (22 classes)	8,965	4.2%

At the group level where EDAS operates, 55.3% of groups contain multiple error classes ($K > 1$) and therefore activate the diversity-aware Branch C, covering 59.4% of all incorrect rollouts; these groups contain 2.41 error classes on average, with a mean Shannon entropy of 0.649. Homogeneous `WrongAnswer` groups account for only 12.7% of all groups. The training dynamics also show no progressive collapse toward `WrongAnswer`: after the initial stage its proportion remains between 16.2% and 24.8%, while the fraction of multi-class groups increases from 46.9% at step 10 to 60.9% at

step 98. The diversity-aware signal therefore remains active throughout training rather than gradually disappearing.

F Experimental Details

This appendix provides a comprehensive summary of the hyperparameters and experimental configurations used for the mathematical reasoning and code generation tasks.

F.1 Evaluation

Math Reasoning. The maximum reasoning length for evaluation is set to 32,768 tokens. For datasets with limited samples (AIME 24-26, AMC 23-24, and HMMT 25), we employ a repeat-32 strategy with a sampling temperature of $T = 0.7$, top-p of 0.95, and top-k of 40 to report the average accuracy. For OlympiadBench, we utilize greedy decoding to ensure deterministic results. To ensure the robustness of answer extraction and matching, we leverage the Math-Ruler library[1]. To assess how the model's capabilities evolve, we employ the unbiased estimator proposed by Chen et al. [2021] to calculate Pass@k:

$$\text{Pass@}k = \mathbb{E}_{x\sim\mathcal{D}}\left[1 - \frac{\binom{n-c}{k}}{\binom{n}{k}}\right] \tag{10}$$

where n ($n \geq k$) denotes the total number of generated sequences per problem, and c represents the number of correct sequences among them.

Code Generation. For code-generation evaluation, we set the maximum generation length to 16,384 tokens. We adopt benchmark-specific repeated sampling and report Pass@8 for LiveCodeBench v6 and Codeforces, and Pass@1 for HumanEval+, MBPP+. For repeated sampling, we use temperature $T = 0.7$, top-$p = 0.8$. A submission is considered correct only when it passes all test cases for a given problem. For robustness, code outputs are post-processed with benchmark-specific extraction and then validated through sandboxed execution or official evaluation protocols (e.g., EvalPlus for HumanEval+/MBPP+).

F.2 Training Setup

Math Reasoning. We utilized DAPO-Math-17K [Yu et al., 2025] as our training dataset, which is a widely recognized dataset for mathematical reasoning tasks within the RLVR domain. All algorithms were trained for two epochs on this dataset. During the training phase, we set the rollout batch size to 256, with 10 response sequences generated per prompt and a global batch size of 64. For the sampling strategy, both temperature and top-p were set to 1.0 to ensure stochastic diversity. Following standard protocols in mathematical RLVR, the correctness of the final answer served as the sole source of the reward signal.

Rollout and Optimization Budget. Since Dynamic Sampling triggers re-generation whenever a batch contains filtered groups, we verify from the complete training logs of the main Qwen3-8B comparison that the two methods consume essentially the same budget. Over the full run, DAPO performs 411 generation attempts (1,052,160 generated trajectories) and EDAS w/ DAPO performs 412 (1,054,720 trajectories, a 0.24% increase), while the number of trajectories actually used for optimization is exactly identical for the two methods (343,040 each). The comparison reported in Table 1 is therefore sample-matched: the gains of EDAS cannot be attributed to a larger optimization dataset or to any meaningful difference in rollout-generation budget.

Code Generation. We train Qwen3-4B on the Code-R1 [Liu and Zhang, 2025] dataset using DAPO as the base algorithm, with and without EDAS. The maximum prompt and response lengths are set to 2,048 and 4,096 tokens, respectively. We generate 10 rollouts per prompt with the same batch-size setting as in mathematical reasoning and train for one epoch. The EDAS hyperparameters are $\alpha = 0.4$, $\beta = 0.2$, $\kappa = 2.0$, consistent with the math experiments.

[1] https://github.com/hiyouga/MathRuler

G Baseline Setup

In this section, we describe the baselines used in our experiments.

GRPO and DAPO. GRPO serves as a foundational algorithm in the RLVR domain, making it a natural choice for our primary baseline. Building upon this, DAPO [Yu et al., 2025] introduces Clip-Higher and Dynamic Sampling techniques, achieving substantially stronger performance than GRPO.

PKPO. For PKPO, we set the hyperparameter $k = 8$. Given that rewards in the RLVR setting are strictly binary, we adopt the discrete formulation as proposed in the original study. Specifically, let y_i denote the i-th generated response and y^* be the ground-truth answer. The reward assignment is defined as:

$$r_i = \begin{cases} \dfrac{k}{n} & \text{if } \mathbb{I}(y_i = y^*) = 1 \\ \dfrac{k}{n} \cdot \rho(n-1, c, k-1) & \text{if } \mathbb{I}(y_i = y^*) = 0 \end{cases} \tag{11}$$

where the term $\rho(n, c, k)$ is defined as the unbiased estimator of the Pass@k rate:

$$\rho(n, c, k) \equiv 1 - \frac{\binom{n-c}{k}}{\binom{n}{k}} \tag{12}$$

Entropy-Enhanced Advantage. To implement the Entropy-Enhanced Advantage baseline, we follow the original configuration by reshaping the advantage values. This approach augments the learning signal with a weighted entropy term to encourage token-level diversity:

$$A_t^{\text{reshape}} = A_t + \min\left(\alpha \cdot \mathcal{H}_t^{\text{detach}},\ \frac{|A_t|}{\kappa}\right), \text{ where } \alpha > 0 \text{ and } \kappa > 1, \tag{13}$$

In this formulation, $\mathcal{H}_t$ represents the token-level entropy at each position t:

$$\mathcal{H}_t = -\sum_{v \in \mathcal{V}} \pi_\theta(v \mid q, o_{<t}) \log \pi_\theta(v \mid q, o_{<t}). \tag{14}$$

Following the recommended settings in the literature, we employ a scaling factor $\alpha = 0.4$ and a clipping threshold $\kappa = 2$.

H Why Embedding-Based Answer Grouping Cannot Replace Symbolic Equivalence

A natural alternative to the symbolic equivalence check used in EDAS's intra-group error partitioning (§3.1) is to cluster incorrect responses by their *embedding similarity*—representing each generated trajectory as a dense vector and grouping those whose vectors lie close together. This approach is appealing because it does not require answer extraction and could, in principle, capture semantic differences even when answers are expressed in varied surface forms. However, we present empirical evidence that embedding similarity is fundamentally unsuitable for distinguishing mathematically correct from incorrect reasoning, and therefore cannot serve as a reliable basis for the diversity signal central to EDAS.

Setup. We collected 32 repeated rollouts per problem for all 30 problems of AIME 2025, generated by `Qwen3-8B` (960 total trajectories). Final answers were extracted via `\boxed{}` pattern matching. For each problem that elicited at least two numerically distinct answers, we selected up to six representative trajectories (one per distinct answer), embedded each full reasoning chain with `BAAI/bge-large-en-v1.5`, and computed pairwise cosine similarity.

Deceptive pairs. Out of 403 pairs with numerically different answers, **308** (76.4%) achieved cosine similarity >0.95, with an average of 0.978 and a maximum of 1.000. This means that for the vast majority of answer disagreements, an embedding-based clustering method would *merge* the two trajectories into the same group despite their producing different answers.

Deceptive triples. To stress-test the failure in the context most directly relevant to EDAS—where we need to distinguish a *correct* trajectory from *two different wrong* trajectories—we identify naturally occurring triples from the model's own rollouts: for each problem, we pair the trajectory reaching the correct answer with two trajectories reaching *distinct* wrong answers, and measure the minimum pairwise similarity across all three pairs. Table 5 summarizes the distribution over 114 such triples.

Table 5: Minimum pairwise embedding similarity for (correct, wrong_1, wrong_2) triples on AIME 2025 (Qwen3-8B, 32 rollouts/problem).

Condition	**Count (out of 114)**
$\min$ -sim ≥ 0.99	2
$\min$ -sim ≥ 0.98	19
$\min$ -sim ≥ 0.95	76
Average $\min$ -sim	0.956
Maximum $\min$ -sim	0.997

Illustrative example. For AIME 2025 Problem 5 (isosceles trapezoid with inscribed circle of radius 3 and area 72; find $r^2 + s^2$), 27 out of 32 rollouts arrive at the correct answer 504, while isolated rollouts produce 48 and 424 respectively. The three pairwise cosine similarities are 0.997, 0.997, and 1.000 (average 0.998). An embedding-based grouping scheme would classify all three trajectories into the same cluster, completely failing to detect that two of them are erroneous.

Root cause and implications. The fundamental reason for this failure is that general-purpose embedding models encode *surface-level textual features*—vocabulary, sentence structure, topic, and style—rather than the *logical validity of a reasoning chain*. Mathematical reasoning errors often reduce to a single incorrect intermediate step or a one-digit numerical discrepancy, and such differences are absorbed by the high-dimensional embedding space as negligible perturbations.

This finding has a direct implication for EDAS: the equivalence relation used to partition incorrect trajectories into error classes (§3.1) *must* be grounded in symbolic answer extraction and mathematical canonicalization (via tools such as `math-ruler`), not in embedding proximity. Using embedding-based grouping as a drop-in replacement would collapse the diversity signal: answers that are numerically far apart (and hence represent genuinely different failure modes) would be counted as the same error type, eliminating the intra-group entropy that EDAS relies on to modulate exploration.

I Bottleneck-Breaking Analysis

A key motivation for EDAS is to help the model escape problems where it consistently fails under the base policy—problems we term *hard* problems, defined as those for which the base model's Average Pass Rate (APR) across 32 repeated rollouts is exactly 0%. These problems represent the hardest frontier of the model's capability: the base policy has *never* produced a correct solution in 32 attempts, so any improvement must come from genuinely new reasoning paths discovered during RL training. We analyse how many such problems each algorithm "breaks through" after training.

Setup. We evaluate on four challenging benchmarks: AIME 2025, AIME 2026, AMC 2024, and HMMT Feb 2025, using 32 rollouts per problem. For each benchmark, we identify the subset of problems where the base model achieves APR $= 0$, then measure the fraction of these problems for which DAPO and EDAS achieve APR > 0 after training—the *breakthrough rate*. We conduct this analysis across three model configurations to test generality:

- **Qwen3-8B** (Table 3): the primary setting reported in the main paper.
- **Qwen3-4B** with no-think prompting (Table 6): a smaller instruct model where the thinking mode is disabled.

- **Qwen3-4B-Base** (Table 7): a smaller base (pre-trained only) model without instruction tuning.

Results on Qwen3-8B. Table 3 (in the main paper) summarizes the per-benchmark results on Qwen3-8B. Across the four benchmarks, 64 problems are classified as hard for the base model. After training, DAPO breaks through on 22 of these (34.4%), while EDAS breaks through on 30 (46.9%), a relative improvement of **36%** in breakthrough rate. Notably, there are **10 problems** (15.6% of all hard problems) where EDAS achieves APR > 0 but DAPO remains at 0%, demonstrating that EDAS discovers reasoning paths that DAPO never finds.

Results on Qwen3-4B. Table 6 shows the same analysis on Qwen3-4B (instruct, no-think mode). Out of 54 hard problems, DAPO breaks through on 22 (40.7%) while EDAS breaks through on 24 (44.4%), with 3 EDAS-only breakthroughs versus only 1 DAPO-only. The advantage is directionally consistent with Qwen3-8B but more modest (+9.1% relative improvement), likely because the instruction-tuned model already possesses some exploration diversity inherited from supervised fine-tuning, leaving less room for EDAS's diversity mechanism to add value.

Table 6: Bottleneck-breaking analysis on Qwen3-4B (Instruct, no-think). Hard = base model APR $= 0$ over 32 rollouts.

Benchmark	Hard	DAPO Break	EDAS Break	EDAS-only
AIME 2025	10	6	**6**	1
AIME 2026	13	6	**6**	0
AMC 2024	12	5	**6**	1
HMMT Feb 25	19	5	**6**	1
Total	**54**	**22** (40.7%)	**24** (44.4%)	**3**

Results on Qwen3-4B-Base. Table 7 presents the results on Qwen3-4B-Base, the most revealing setting. Of 90 hard problems, DAPO breaks through on only 18 (20.0%), while EDAS breaks through on 31 (34.4%)—a relative improvement of **72.2%**, nearly double the gain observed on Qwen3-8B. The EDAS-only count reaches **14 problems** (15.6% of all hard problems), compared to merely **1** DAPO-only breakthrough. The effect is especially striking on AIME 2025, where DAPO achieves zero breakthroughs while EDAS breaks through on 4 problems, and on HMMT Feb 2025 (DAPO: 1, EDAS: 6).

Table 7: Bottleneck-breaking analysis on Qwen3-4B-Base. Hard = base model APR $= 0$ over 32 rollouts.

Benchmark	Hard	DAPO Break	EDAS Break	EDAS-only
AIME 2025	18	0	**4**	4
AIME 2026	24	4	**7**	4
AMC 2024	25	13	**14**	1
HMMT Feb 25	23	1	**6**	5
Total	**90**	**18** (20.0%)	**31** (34.4%)	**14**

Discussion. The superiority of EDAS on hard problems is consistent across all three model scales and configurations, confirming that the effect is systematic rather than model-specific. Because EDAS actively penalizes mode collapse and promotes rare error types, the policy is continuously nudged away from homogeneous failure modes. When the base model fails all 32 attempts, it typically does so by converging to the same incorrect reasoning chain; DAPO inherits this mode-collapsed behavior and struggles to escape it. EDAS's diversity-encouraging mechanism diversifies the exploration landscape even for the hardest problems, increasing the probability that at least one rollout discovers a correct reasoning path—thereby generating a positive reward signal that DAPO never receives.

Comparing across model configurations reveals an informative gradient. The advantage is largest on Qwen3-4B-Base (+72.2% relative), moderate on Qwen3-8B (+36%), and smallest on Qwen3-4B

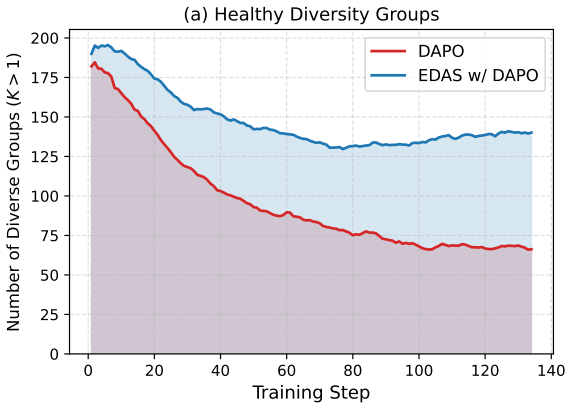

Figure 6: Number of groups with healthy error diversity ($K > 1$) across 2 epochs of training. EDAS (blue) maintains on average 1.61× more diverse groups than DAPO (red). The gap widens from 1.45× in the first epoch to 1.98× in the second epoch, demonstrating that EDAS's diversity preservation strengthens precisely as the remaining problems grow harder.

Instruct (+9.1%). This ordering aligns with the degree of mode collapse in the initial policy: base models without instruction tuning are most prone to homogeneous failure modes, and therefore benefit the most from EDAS's diversity-encouraging objective. Instruction-tuned models already exhibit some exploration diversity from supervised fine-tuning, leaving less headroom for improvement. The consistent EDAS-only breakthroughs across all settings and benchmarks—10 on 8B, 14 on 4B-Base, and 3 on 4B Instruct—confirm that EDAS systematically discovers reasoning paths that DAPO never finds, regardless of model scale.

J Internal Dynamics of EDAS's Advantage Adjustment

We log EDAS's internal statistics throughout training (2 epochs on DAPO-Math-17K) and compare them against the same group-level metrics passively recorded from DAPO on the identical training setup (Qwen3-8B, DAPO-Math-17K, 2 epochs, batch size 256). This provides a controlled comparison of how the two algorithms shape the error landscape during RLVR. Note that for DAPO, these metrics are observed but not acted upon; EDAS actively modulates advantages based on them.

J.1 Metric Definition

For each training step, we count the number of prompt groups exhibiting healthy error diversity ($K > 1$, i.e., at least two distinct wrong answer types coexist among the incorrect trajectories). This metric directly measures the breadth of the model's exploration: more diverse groups mean more varied failure modes, providing richer gradient signal for discovering correct reasoning paths. We compute this identically for both EDAS and DAPO: for each group, we extract incorrect answers from the rollouts, canonicalise them, and count the number of distinct error types K.

J.2 EDAS Preserves Significantly More Diverse Groups

As shown in Figure 6, EDAS consistently maintains far more diverse groups than DAPO throughout training. Averaged across the entire training run (2 epochs), EDAS preserves a mean of 144.1 diverse groups per step compared to DAPO's 89.5—a **1.61**× ratio overall. The advantage grows over training: during the first epoch the ratio averages 1.45×, while during the second epoch it widens to **1.98**×.

This widening gap reflects a fundamental dynamic: as standard RLVR training concentrates the policy on higher-reward patterns, error diversity naturally erodes. DAPO's diverse groups decline from 182 to as low as 54—a 70.3% collapse. EDAS counteracts this erosion: its diverse groups decline more gently (from 189 to a minimum of 112) and partially recover in late training, ending at 146. The recovery is driven by EDAS's Branch B penalty, which forces mode-collapsed groups ($K=1$) to diversify, replenishing the pool of diverse groups even as training progresses.

J.3 Implications

This result provides direct mechanistic evidence for EDAS's effectiveness:

- **Diversity preservation is effective.** Despite both algorithms starting from the same initial diversity (189 vs. 182 at step 1), EDAS's Branch B/C mechanism prevents the severe diversity collapse that DAPO experiences (70.3% decline vs. 40.7%).
- **The effect strengthens when it matters most.** The EDAS/DAPO ratio widens from 1.45× to 1.98× from the first to the second epoch. In the second epoch, the remaining unsolved problems are the hardest, and maintaining diverse error types on these problems is critical for the model to discover new reasoning paths—exactly the mechanism behind the bottleneck-breaking improvements reported in Appendix I.
- **EDAS actively reverses diversity loss.** Rather than merely slowing collapse, EDAS's diverse group count partially *recovers* in the second epoch (from a minimum of 112 back to 146), indicating that Branch B successfully forces collapsed groups to re-diversify.

K Pass@4 Dynamics During Training

To examine whether the error diversity preserved by EDAS (Appendix J) is accompanied by higher coverage of correct solutions *during* training rather than only at the final checkpoint, we track Pass@4 along the two Qwen3-8B training trajectories of DAPO and EDAS w/ DAPO. We use the retained checkpoints of the runs reported in the main paper at a pre-specified 10-step interval, sample $n = 10$ rollouts per problem, and apply the same unbiased Pass@k estimator as in Appendix F with $k = 4$.

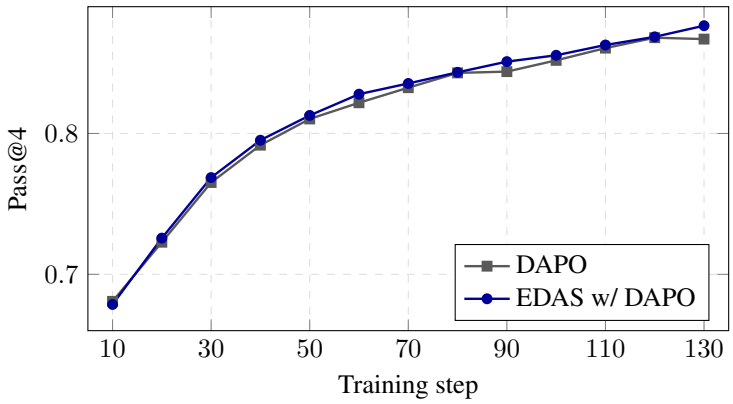

Figure 7: Pass@4 as a function of training step on the Qwen3-8B training trajectories of DAPO and EDAS w/ DAPO, evaluated at a pre-specified 10-step interval from the retained checkpoints.

As shown in Figure 7, EDAS maintains a positive Pass@4 gap at 12 of the 13 reported checkpoints—every checkpoint from step 20 onward—with an average difference of $+0.33$ percentage points and the largest gap ($+0.94$ percentage points, 0.8765 vs. 0.8670) occurring at the final step 130. The absolute gap is modest: both methods reach high training rewards in the later stage, and under the independent-sampling approximation Pass@4 $\approx 1 - (1 - p)^4$ saturates as the single-rollout success probability p grows, so further improvements in p appear only as small absolute changes in Pass@4. Nevertheless, the coverage advantage is directionally consistent throughout training and aligns with the final evaluation results in Table 1. We note that these checkpoints come from the same training trajectories as the main results; this analysis therefore provides process-level evidence consistent with the diversity-to-coverage mechanism, but does not by itself constitute an independent causal proof.

L Statistical Significance of the Main Results

To assess whether the gains reported in Table 1 exceed evaluation-sampling noise, we perform a one-sided paired bootstrap test on the per-sample evaluation results of the main Qwen3-8B setting (DAPO vs. DAPO w/ EDAS). For each benchmark, we draw 10^4 paired bootstrap resamples at the problem level over the evaluation results (repeat-32 sampling for AIME, AMC, and HMMT; greedy

decoding for OlympiadBench) and test the one-sided hypothesis $H_0 : \Delta \leq 0$ against $H_1 : \Delta > 0$, where Δ denotes the performance difference between DAPO w/ EDAS and DAPO. This directly corresponds to the directional claim of interest—whether EDAS brings a positive improvement. We report all benchmarks in full without selective filtering.

Table 8: One-sided paired bootstrap test on Qwen3-8B (DAPO vs. DAPO w/ EDAS). LB denotes the one-sided 95% confidence lower bound of Δ; LB > 0 corresponds to one-sided $p < 0.05$. Gains, lower bounds, and p-values are computed from unrounded per-sample results and may differ by 0.01 from direct subtraction of the rounded scores.

Benchmark	N	**DAPO**	**DAPO w/ EDAS**	Δ	**Δ 95% CI LB**	p-value
AIME 2024	30	41.46	48.85	+7.40	+1.15	0.0283
AIME 2025	30	38.12	45.52	+7.40	+3.33	0.0003
AIME 2026	30	39.17	48.65	+9.48	+4.69	0.0005
AMC 2023	40	84.45	90.39	+5.94	+2.73	0.0005
AMC 2024	45	56.88	59.38	+2.50	−0.97	0.1181
HMMT Feb 2025	30	15.94	20.10	+4.17	+1.04	0.0107
OlympiadBench	674	58.75	65.88	+7.12	+4.75	<0.0001
Average	—	47.82	54.11	+6.29	+4.73	<0.0001

As shown in Table 8, EDAS obtains positive point estimates on all seven benchmarks, six of which reach statistical significance with one-sided $p < 0.05$; for these six benchmarks, the one-sided 95% confidence lower bounds are all strictly positive. On AMC 2024, the point estimate is positive (+2.50) but does not reach significance ($p = 0.1181$), so we interpret this result as positive in direction yet statistically uncertain. The overall average improvement of +6.29 carries a one-sided 95% confidence lower bound of +4.73 with $p < 10^{-4}$: even under this conservative lower bound, the average benefit of EDAS remains clearly positive and substantially exceeds the random variation introduced by repeated evaluation.

M Robustness Across Independent Training Seeds

The paired bootstrap in Appendix L measures the uncertainty of a fixed training checkpoint under repeated evaluation. To additionally assess training randomness, we run two further matched-seed experiments (seeds 42 and 100) in the main Qwen3-8B setting, training DAPO and EDAS w/ DAPO separately under each seed; apart from whether EDAS is enabled, each matched-seed pair uses an identical training configuration. Together with the run reported in the main paper, this yields three matched-seed comparisons.

Table 9: Seven-benchmark averages of DAPO and EDAS w/ DAPO under three matched training seeds (Qwen3-8B). $\pm$ denotes the sample standard deviation across the three seeds.

Training seed	**DAPO**	**EDAS w/ DAPO**	**Paired improvement**	**Improved**
Original run	47.82	54.11	+6.29	7/7
42	47.66	54.39	+6.72	7/7
100	48.17	52.89	+4.72	7/7
Mean $\pm$ std	47.88 $\pm$ 0.26	53.80 $\pm$ 0.80	+5.91 $\pm$ 1.05	21/21

All three matched-seed experiments obtain positive paired improvements in the seven-benchmark average (+6.29, +6.72, and +4.72; mean +5.91 $\pm$ 1.05), and EDAS improves every benchmark under every seed, yielding 21/21 positive seed–benchmark pairs (Table 10). Even for seed 100, which has the lowest overall gain, the seven-benchmark average of EDAS remains +4.72 above DAPO under the same seed, while the low seed-to-seed variation of DAPO itself (47.88 $\pm$ 0.26) indicates that the comparison is not driven by an accidentally favorable EDAS seed or an unfavorable DAPO seed. The two analyses in this and the previous appendix therefore address different sources of uncertainty: the paired bootstrap shows that the benefit of EDAS exceeds repeat-evaluation noise, and the matched-seed experiments show that it is stably reproduced across independent training runs.

Table 10: Per-benchmark means, standard deviations, and paired improvements over the three matched training seeds.

Benchmark	DAPO	EDAS w/ DAPO	Paired improvement	Positive seeds
AIME 2024	42.29 ± 2.39	48.09 ± 1.15	$+5.80 \pm 1.92$	3/3
AIME 2025	38.13 ± 1.45	44.41 ± 1.42	$+6.28 \pm 2.67$	3/3
AIME 2026	39.58 ± 0.81	48.54 ± 0.99	$+8.96 \pm 0.52$	3/3
AMC 2023	84.50 ± 0.64	89.74 ± 1.92	$+5.24 \pm 2.55$	3/3
AMC 2024	57.52 ± 0.78	59.58 ± 0.18	$+2.06 \pm 0.70$	3/3
HMMT Feb 2025	14.88 ± 2.01	20.52 ± 0.48	$+5.64 \pm 2.46$	3/3
OlympiadBench	58.29 ± 1.50	65.68 ± 0.23	$+7.39 \pm 1.61$	3/3
Average	47.88 ± 0.26	53.80 ± 0.80	$+5.91 \pm 1.05$	21/21

N Hyperparameter Sensitivity and Branch Ablations

The configuration $(\alpha, \beta, \kappa) = (0.4, 0.2, 2)$ used throughout the paper was fixed a priori as an untuned default within a reasonable range, before the sweep reported here was conducted. To assess sensitivity, we sweep $\alpha \in \{0.2, 0.3, 0.4, 0.5, 0.6\}$, $\beta \in \{0.1, 0.2, 0.3\}$, and $\kappa \in \{2, 3\}$ around this default in the main Qwen3-8B DAPO setting. We additionally ablate the two branches of Equation 2: setting $\alpha = 0$ removes the diversity-aware redistribution under multiple error modes (Branch C) while retaining the homogeneous-error penalty, and setting $\beta = 0$ removes the homogeneous-error-collapse penalty (Branch B) while retaining the redistribution. Table 11 reports all tested configurations without omission.

Table 11: Hyperparameter sweep and branch ablations on Qwen3-8B (DAPO base). Scores are shown to two decimal places; averages and gains are computed from unrounded results and may differ by 0.01 from direct subtraction of the displayed values. The configuration used in the paper is marked †.

Method	α	β	κ	Avg.	Δ Avg.	AIME25	AIME24	AIME26	AMC23	AMC24	HMMT25	O.Bench
DAPO base	—	—	—	47.82	—	38.12	41.46	39.17	84.45	56.88	15.94	58.75
EDAS	0.2	0.2	2	51.41	+3.59	40.73	45.73	46.98	86.09	57.22	20.52	62.61
EDAS	0.3	0.2	2	55.34	+7.51	43.44	50.73	50.52	89.30	60.56	23.23	69.58
EDAS†	0.4	0.2	2	54.11	+6.29	45.52	48.85	48.65	90.39	59.38	20.10	65.88
EDAS	0.5	0.2	2	55.04	+7.21	47.71	49.58	47.60	90.62	60.21	23.65	65.88
EDAS	0.6	0.2	2	53.37	+5.54	43.12	50.52	49.27	86.41	58.89	20.83	64.54
EDAS	0.4	0.1	2	54.29	+6.46	45.31	50.83	49.79	87.19	60.90	22.19	63.80
EDAS	0.4	0.3	2	52.26	+4.43	44.06	45.83	48.33	86.88	57.29	20.21	63.20
EDAS	0.4	0.2	3	52.07	+4.24	44.48	46.25	45.83	85.47	61.74	17.50	63.20
w/o Branch C ($\alpha = 0$)	0	0.2	2	49.79	+1.96	41.00	46.00	38.50	82.70	57.40	19.00	63.90
w/o Branch B ($\beta = 0$)	0.4	0	2	49.52	+1.69	41.46	44.38	43.85	81.88	58.47	18.12	58.46

Three observations emerge. First, every complete EDAS configuration outperforms DAPO on all seven benchmarks and on the seven-benchmark average, with gains ranging from +3.59 to +7.51; the benefit of EDAS therefore does not depend on a single knife-edge parameter point. Second, the paper's configuration is not the best-performing point in the sweep—$(\alpha, \beta) = (0.3, 0.2)$ and $(0.5, 0.2)$ reach averages of 55.34 and 55.04, both above the 54.11 reported in the main results—so the reported gain does not rely on selecting the sweep's optimum. Third, both single-branch ablations remain above DAPO (+1.96 for $\alpha = 0$; +1.69 for $\beta = 0$) but fall well short of any complete configuration and improve only 5/7 benchmarks each, indicating that the two mechanisms carry complementary training signals and that the full benefit arises from their combination.

O Preliminary Generalization Beyond the Qwen3 Family

To test whether the motivating observation and the gains of EDAS are specific to the Qwen3 family, we repeat the main comparison on Gemma-3-4B-IT, keeping the hyperparameter configuration of the main experiments ($\alpha = 0.4$, $\beta = 0.2$, $\kappa = 2.0$) unchanged.

As shown in Table 12, EDAS improves all seven benchmarks, raising the average score from 22.98 to 24.44 (+1.46). We further replicate the diagnostic analysis of section 2 on Gemma-3-4B-IT,

Table 12: DAPO vs. EDAS w/ DAPO on Gemma-3-4B-IT across the seven math benchmarks, using the same hyperparameters as the main experiments.

Benchmark	**DAPO**	**EDAS w/ DAPO**	Δ
AIME 2025	12.19	13.65	+1.46
AIME 2024	12.08	12.50	+0.42
AIME 2026	9.90	12.19	+2.29
AMC 2023	48.20	48.75	+0.55
AMC 2024	36.60	37.78	+1.18
HMMT Feb 2025	0.94	2.71	+1.77
OlympiadBench	40.95	43.47	+2.52
Average	22.98	24.44	+1.46

stratifying problems by their initial error diversity and measuring the improvement after DAPO training relative to before training.

Table 13: Relationship between initial error diversity and training improvement on Gemma-3-4B-IT, following the analysis protocol of section 2.

Diversity level	**Mean distinct wrong answers**	**Improvement after training**	**Improvement rate**
Level 1	2.3	+0.74	10.5%
Level 2	6.4	+1.39	11.1%
Level 3	10.9	+1.79	17.5%
Level 4	15.2	+2.49	22.0%

As initial error diversity increases, the post-training improvement rises strictly monotonically from +0.74 to +2.49—the highest-diversity group improves 3.4 times as much as the lowest—and the problem-level improvement rate increases from 10.5% to 22.0% (Table 13). On a non-Qwen model family, we thus observe both the positive error-diversity–training-improvement relationship and the performance benefit of EDAS over DAPO. We regard this as preliminary cross-family evidence that the two central observations of the paper transfer beyond Qwen3, without claiming coverage of all model architectures or scales.

P Limitations

While EDAS demonstrates consistent improvements across multiple models and domains, we identify several limitations that contextualize the current results and suggest directions for future work.

Reliance on Observable Output Features for Error Partitioning. EDAS defines error equivalence through observable output features—final numerical answers for mathematical reasoning and exception types for code generation. This design, while effective and computationally efficient, cannot distinguish different reasoning paths that happen to produce the same observable output. For instance, two trajectories may arrive at the same incorrect numerical answer through entirely different logical errors, yet EDAS would classify them as belonging to the same equivalence class. Incorporating process-level diversity signals, e.g., based on intermediate reasoning steps or execution traces, could yield a finer-grained error taxonomy but would introduce significant computational overhead and design complexity.

Domain-Specific Engineering of the Equivalence Function. Although EDAS's downstream computation is fully domain-agnostic, the equivalence function $\mathcal{E}$ must be manually designed for each new domain. For mathematical reasoning and code generation, natural choices exist (canonicalized answers and exception types, respectively). However, extending EDAS to domains such as open-ended text generation, multi-step tool use, or formal theorem proving requires identifying appropriate discrete error labels—a non-trivial design decision that may limit out-of-the-box applicability.

Fixed Hyperparameters Across Training. The current formulation uses fixed hyperparameters (α, β, κ) throughout training. As the policy evolves, the optimal balance between diversity encouragement and collapse penalty may shift: early training may benefit from stronger exploration incentives, while later stages may require more conservative adjustments to avoid destabilizing a near-converged policy. Adaptive scheduling of these hyperparameters remains unexplored.

Sensitivity to Group Size and Sampling Budget. EDAS's information-theoretic quantities (entropy, self-information) are estimated from finite samples within each rollout group. With the group size fixed at $N = 10$ in our experiments, the entropy estimate may be noisy for problems that admit many distinct error types, potentially underestimating the true diversity. Larger group sizes would improve estimation quality but proportionally increase computational cost. The interaction between group size, diversity estimation accuracy, and downstream performance has not been systematically characterized.

Evaluation Scope. Our experiments focus on mathematical reasoning (competition-level problems) and code generation. While these represent two important verifiable domains, the generalizability of EDAS to other settings—such as formal theorem proving, scientific reasoning, or multi-hop question answering with verifiable answers—remains empirically unvalidated. In addition, the main experiments are conducted on models from the Qwen3 family; a preliminary replication on Gemma-3-4B-IT (Appendix O) reproduces both the motivating observation and the performance gains of EDAS, but broader generality across model families has not been fully established. Likewise, our experiments cover neither policies initialized through alternative pre-RL pipelines such as policy distillation [Dong et al., 2026], nor sparse mixture-of-experts architectures [Xu et al., 2026], whose training dynamics may differ from those of the dense models studied here. We also note that the during-training diversity metrics reported in Figure 5a and Figure 6 measure quantities that EDAS explicitly optimizes for, and should therefore be read as mechanistic evidence; the coverage-oriented results (Pass@k in Figure 4 and the breakthrough analysis in Appendix I) constitute the load-bearing evidence that the preserved diversity is productive. The consistently positive results across our tested domains are encouraging, but do not guarantee transferability to domains with fundamentally different error structures.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Refer to Abstract and Introduction.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss limitations in the Conclusion and provide a detailed analysis in Appendix P.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: Refer to Appendix A and Appendix B.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Refer to Experiment Setup in main paper, Appendix F and Appendix G.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The datasets we used are all open-accessed. We will open source our code.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Refer to Appendix F and Appendix G.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: For the main Qwen3-8B setting, we report a one-sided paired bootstrap significance test (10^4 resamples at the problem level over the repeated evaluation samples) with one-sided 95% confidence lower bounds and p-values for all seven math benchmarks and their average (see Appendix L), together with mean and standard deviation over three independent training seeds, under which EDAS improves all 21 seed–benchmark pairs (see Appendix M). To mitigate the effect of stochastic variation in the remaining settings, we conduct repeated evaluations with 32 samples per instance and report Pass@k metrics where applicable; performing full significance testing across all benchmarks and model variants would require substantially more computational resources.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Refer to Experiment Setup.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Our research conforms, in every respect, with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: There is no societal impact of our work.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.

- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: There is no such risk in the paper.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We comply with the licenses of existing assets used in the paper and provide necessary references.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: The proposed algorithm can be used for LLMs post-training. Thus we present some experiments about LLMs post-training. Meanwhile, we use LLMs for editing.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.