Rethinking the Sampling Criteria in Reinforcement Learning for LLM Reasoning: A Competence-Difficulty Alignment Perspective

Deyang Kong[1,2*], **Qi Guo**[1,2*], **Xiangyu Xi**[2†],
Wei Wang[2], **Jingang Wang**[2], **Xunliang Cai**[2], **Shikun Zhang**[1], **Wei Ye**[1†]
[1]National Engineering Research Center for Software Engineering, Peking University, Beijing, China
[2]Meituan Group, Beijing, China
{kong.deyang@foxmail.com, qguo@stu.pku.edu.cn, xixy10@foxmail.com, wye@pku.edu.cn}

Abstract

The low sampling efficiency during the rollout phase poses a significant challenge to scaling reinforcement learning for large language model reasoning. Existing methods attempt to improve efficiency by scheduling problems based on problem difficulties. However, these approaches suffer from unstable and biased estimations of problem difficulty and fail to capture the alignment between model competence and problem difficulty in RL training, leading to suboptimal performance. To address these challenges, we introduce **C**ompetence-**D**ifficulty **A**lignment **S**ampling (**CDAS**). This approach allows for accurate and stable estimation of problem difficulties by aggregating historical performance discrepancies across problems. Subsequently, model competence is quantified to adaptively select problems whose difficulties align with the model's current competence using a fixed-point system. Extensive experiments in mathematical RL training show that **CDAS** consistently outperforms strong baselines, achieving the highest average accuracy of 45.89%. Furthermore, **CDAS** reduces the training step time overhead by 57.06% compared to the widely-used Dynamic Sampling strategy, verifying the efficiency of **CDAS**. Additional experiments on different tasks, model architectures, and model sizes demonstrate the generalization capability of **CDAS**.

Code — https://github.com/DeyangKong/CDAS

Introduction

Advanced large language models (LLMs) exemplified by DeepSeek-R1 (Guo et al. 2025) and OpenAI O1 (Jaech et al. 2024) demonstrate remarkable performance in challenging tasks like mathematics. As a core technology in their reports, the Reinforcement Learning (RL) algorithm, such as Proximal Policy Optimization (Schulman et al. 2017) (PPO) and Group Relative Policy Optimization (Shao et al. 2024) (GRPO), is employed to amplify the reasoning capabilities of the models. It works by utilizing a verifier as the reward model to guide the generation of high-quality reasoning chains without the need for data annotation. Despite the promise, the RL training is costly and hard to scale, particularly due to its low sample efficiency during the rollout phase. Recent studies (Yu et al. 2025; Zeng et al. 2025; Bae et al. 2025) indicate that sampling overly difficult problems often results in no correct chains, while sampling overly simple problems contributes little to model capabilities, leading to computational waste. Consequently, a host of efforts are devoted to exploring sampling strategies for more efficient and stable RL training.

Existing strategies draw inspiration from Curriculum Learning (CL) (Bengio et al. 2009; Narvekar et al. 2020), scheduling data based on problem difficulty to enhance training stability and efficiency. Curriculum Sampling Strategy (Team et al. 2025) relies on prior difficulty labels, which are excessively offline, neglecting the inherent capabilities of the model. Dynamic Sampling used by DAPO (Yu et al. 2025) demonstrates promising results by oversampling and filtering out problems with the pass rate equal to 1 and 0, which incurs substantial rollout overhead and compromises the training efficiency. Prioritized Sampling Strategy used by Kimi k1.5 (Team et al. 2025) records the latest pass rate of each problem during training and adaptively assigns a higher sampling probability to those with lower pass rates. Overall, the pass rate has been widely adopted as a proxy for modeling problem difficulty.

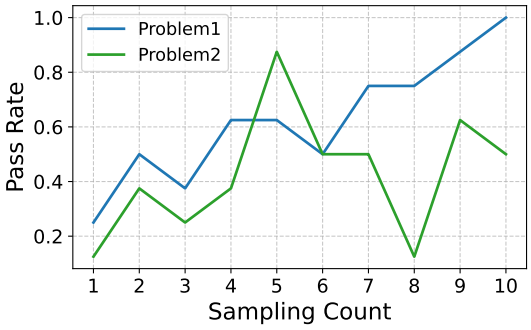


Figure 1: Pass rate variations of two problems in plain GRPO on Qwen2.5-7B using MATH dataset.

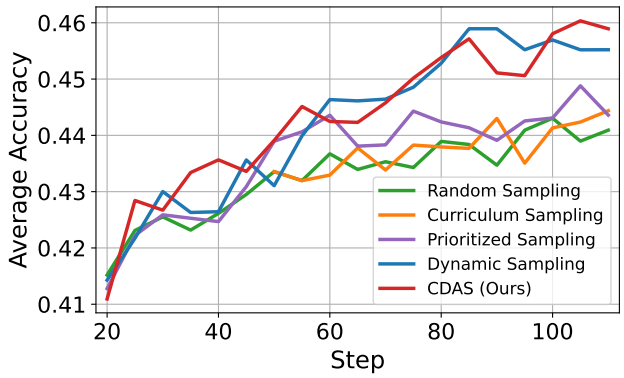
However, these strategies tend to be suboptimal due to two main issues: (1) **Unstable and Biased Estimations of Problem Difficulty Using Single-Step Pass Rate**. As our experiment of training Qwen2.5-7B (Yang et al. 2024) with MATH dataset (Hendrycks et al. 2021b) shows (Figure 1),

*Equal contribution.
†Corresponding authors.

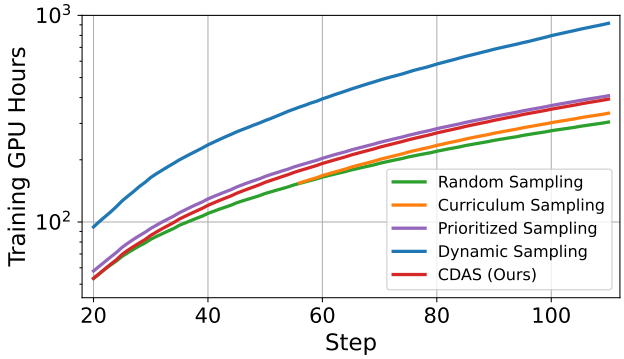
the pass rate of individual problems exhibits considerable fluctuations throughout the training process, leading to unstable estimations of problem difficulty, consistent with several prior works (Zheng et al. 2023; Peng et al. 2023). Moreover, focusing solely on the final pass rate can introduce a difficulty bias. For instance, the pass rate of problem 2 at step 5 happens to surpass that of problem 1, despite their distinct trajectories and difficulty levels. This demonstrates that the pass rate at a single step fails to capture the true complexity and learning dynamics associated with each problem. (2) **Failing to Appropriately Capture the Alignment between Model Competence and Problem Difficulty**. A common strategy, such as in curriculum sampling and prioritized sampling, is to assign higher sampling probabilities to more difficult problems with lower pass rates. A practical consequence is that overemphasizing difficult samples often leads to the selection of many zero-gradient problems with a pass rate of 0 in GRPO training, limiting the training efficiency (Le et al. 2025). However, a more appropriate approach, as advocated in the curriculum learning (Bengio et al. 2009; Narvekar et al. 2020), is to prioritize problems that are more aligned with the current level of competency of the model for a more efficient and effective training.

To address the issues above, we propose **C**ompetence-**D**ifficulty **A**lignment **S**ampling (**CDAS**) for RL training, dynamically sampling problems whose difficulty matches the model competence at the step level. The core intuition behind **CDAS** is twofold: (1) instead of relying solely on the pass rate at a single step, an accumulative estimation that incorporates historical information tends to yield a more stable assessment of problem difficulty; and (2) explicitly modeling model competence to measure its alignment with problem difficulty, thereby enabling more effective sampling decisions. To realize these intuitions, we explicitly define two core concepts: **Model Competence** and **Problem Difficulty**. We then introduce **Competence-Difficulty Alignment** to quantify the gap between them, enabling us to sample problems that are optimally matched to the model's current learning stage. To ensure this difficulty estimation is stable and robust against single-step fluctuations, we aggregate the problem's **historical performance** over time. When integrated into the dynamics of RL training, this entire framework is formulated as a **fixed-point system**, which is theoretically guaranteed to converge, ensuring a stable and principled training process.

To validate the effectiveness of **CDAS**, we conduct mathematical reinforcement learning on Qwen2.5-7B (Yang et al. 2024) with GRPO, the most widely-used RL algorithm, and compare with extensive sampling strategies. The main findings are as follows: (1) **Superior Performance and Efficiency**: Results across six comprehensive mathematical reasoning benchmarks show that **CDAS** consistently outperforms powerful baselines, achieving the highest average accuracy of 45.89%. Compared to Dynamic Sampling, a highly competitive baseline, **CDAS** reduces the training step time overhead by 57.06% (see Figure 2(b)), verifying the efficiency of **CDAS**. (2) **Efficient Sampling Mechanism**: In-depth analysis indicates that **CDAS** successfully samples more problems that align with the model's competence,



(a) Average Accuracy



(b) Total Training GPU Hours

Figure 2: Performance comparison of **CDAS** against baselines. **CDAS** achieves the best performance while demonstrating significant efficiency advantages compared to the strong Dynamic Sampling baseline.

reducing the proportion of zero-gradient problems, significantly less than other sampling strategies. (3) **Proven Generalization and Scalability**: Additional experiments on different tasks (i.e., code generation), model architectures (i.e., LLaMa), and model sizes (i.e., Qwen2.5-14B) demonstrate the generalization capability of **CDAS**.

The contributions of this paper can be summarized as follows:

- We identify and analyze the limitations of existing sampling strategies from a new perspective, highlighting the importance of stable difficulty estimation and dynamic competence-difficulty alignment in RL training.
- We introduce **C**ompetence-**D**ifficulty **A**lignment **S**ampling (**CDAS**), adaptively selecting problems that match the model competence, which is grounded in a theoretically guaranteed fixed-point system. Extensive experiments validate its effectiveness and efficiency.

Preliminary

Group Relative Policy Optimization GRPO utilizes group-based advantage without a value model, thereby reducing computational overhead. Formally, given a problem x, the

correct answer y, and a group of sampled responses $\{\hat{y}_i\}_{i=1}^{G}$ with their corresponding rewards $\{r_i\}_{i=1}^{G}$, GRPO calculates the advantage by normalizing the rewards within each group. The original GRPO objective employs a sample-level loss calculation which potentially introduces length biases (Yu et al. 2025; Liu et al. 2025; Chu et al. 2025), so we utilize a token-level policy gradient loss as our objective function:

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{[x\sim\mathcal{D},\{\hat{y}_i\}_{i=1}^{G}\sim\pi_{\text{old}}(\cdot|x)]} \frac{1}{\sum_{i=1}^{G}|\hat{y}_i|}\sum_{i=1}^{G}\sum_{t=1}^{|\hat{y}_i|}\Bigg(\min\Big(r_{i,t}(\theta)\hat{A}_{i,t}, \text{clip}\Big(r_{i,t}(\theta), 1-\epsilon, 1+\epsilon\Big)\hat{A}_{i,t}\Big) - \beta\mathbb{D}_{\text{KL}}[\pi_\theta \parallel \pi_{\text{ref}}]\Bigg), \tag{1}$$

where

$$r_{i,t}(\theta) = \frac{\pi_\theta(\hat{y}_{i,t} \mid x, \hat{y}_{i,<t})}{\pi_{\theta_{\text{old}}}(\hat{y}_{i,t} \mid x, \hat{y}_{i,<t})}, \tag{2}$$

$$\hat{A}_{i,t} = \frac{r_i - \text{mean}(\{r_i\}_{i=1}^{G})}{\text{std}(\{r_i\}_{i=1}^{G})}. \tag{3}$$

Rule-Based Reward The use of reward model usually leads to reward hacking problem (Gao, Schulman, and Hilton 2023; Weng 2024), so we use a rule-based reward function. The reward is computed using the following rule:

$$r(y, \hat{y}) = \begin{cases} 1 & \text{is\_equivalent}(y, \hat{y}) \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

Notably, we do not employ a format reward. Prior research indicates that strict format constraints may limit the upper bound of model performance (Zeng et al. 2025; Singh et al. 2023; Wang et al. 2024). Therefore, we directly use the final accuracy as the reward.

Competence-Difficulty Alignment Sampling

In this section, we introduce the **C**ompetence-**D**ifficulty **A**lignment **S**ampling (**CDAS**) framework. We first motivate the need for a dynamic and stable difficulty metric, then define model *Competence*, problem *Difficulty*, and their *Alignment*. Finally, we present the alignment-based sampling strategy and its convergence properties.

Rethinking Problem Difficulty: The Need for Relativity and Stability

The efficacy of advanced sampling strategies hinges on their ability to assess problem difficulty accurately. We argue that existing methods are limited by their failure to account for two aspects: the *relativity* and *stability* of difficulty.

The Relativity of Difficulty. Static curriculum learning strategies rely on a fixed, absolute measure of difficulty (e.g., predefined labels). However, problem difficulty is not an intrinsic constant; it is inherently relative to the competence of the model attempting to solve it. A problem that is challenging early in training may become trivial as the model improves, so a meaningful difficulty metric must dynamically reflect the evolving relationship between the model's capability and the problem's complexity.

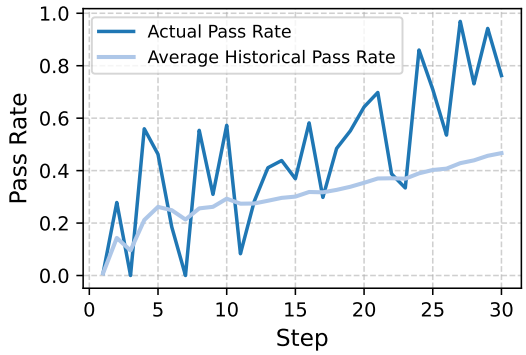


Figure 3: Pass Rate vs Step.

The Stability of Difficulty. Using the instantaneous pass rate as a difficulty proxy introduces high variance. As shown in Figure 3, pass rates fluctuate significantly despite an overall upward trend, making single-step observations noisy and unreliable. To achieve a more reliable estimation, it is necessary to aggregate the problem's **historical performance**, smoothing out short-term noise to better capture the true learning trend.

These limitations often lead to misalignment between model competence and problem difficulty, leading to overly hard or easy samples that yield zero-gradient updates. Motivated by this, we propose a sampling method based on two core concepts: model **Competence** and problem **Difficulty**.

Quantifying the Core Concepts: Competence, Difficulty, and Alignment

To realize our vision of a dynamic and stable sampling method, we introduce three interconnected concepts to ensure a clear logical foundation.

Step 1: Defining Model Competence (C) We define the model's competence C_n as a measure of its overall ability to solve problems at the training step n. Formally, competence is the negative mean of problem difficulty across the dataset $\mathcal{D}$:

$$C_n = -\mathbb{E}_{x\in\mathcal{D}}[D_n(x)] \tag{5}$$

where $D_n(x)$ is the difficulty of problem x, which we will precisely define next. The negative sign indicates that as the average problem difficulty decreases (i.e., the model solves them more effectively), the model's competence score C_n correspondingly increases.

Step 2: Defining Problem Difficulty (D) With the macro-level concept of competence established, we can now define the difficulty of an individual problem. We distinguish between a problem's volatile *instantaneous difficulty* (d) at a single step and its much more robust *stable difficulty* (D), which is informed by its history.

A problem's instantaneous difficulty, $d_n(x)$, at step n is defined as the gap between the model's *expected performance* and its *actual performance*.

- **Actual Performance** is measured by the model's pass rate on problem x at step n, denoted as $s_n(x)$.
- **Expected Performance**, $\hat{P}_{M_n}(x)$, models the probability with which we anticipate the model will solve the problem. This expectation should naturally depend on the gap between the model's prior competence, C_{n-1}, and the problem's established difficulty, $D_{n-1}(x)$. A natural and widely used way to capture this dependency is through a Sigmoid transformation:

$$\hat{P}_{M_n}(x) = \sigma(C_{n-1} - D_{n-1}(x)) \tag{6}$$

The Sigmoid function smoothly maps the real-valued competence–difficulty gap to a probability in $[0, 1]$, a formulation also widely adopted in ability–difficulty modeling such as Item Response Theory and ELO rating systems (Baker 2001; Lord and Novick 2008). If competence far exceeds difficulty ($C \gg D$), the expected probability of success approaches 1. Conversely, if a problem is far too difficult for the model's competence ($C \ll D$), the probability approaches 0. When competence and difficulty are balanced ($C \approx D$), the outcome is most uncertain, with a success probability of 0.5.

The **instantaneous difficulty** $d_n(x)$ is thus the difference between these two quantities:

$$d_n(x) = \underbrace{\sigma(C_{n-1} - D_{n-1}(x))}_{\text{Expected Performance}} - \underbrace{s_n(x)}_{\text{Actual Performance}} \tag{7}$$

If $d_n(x) > 0$, the model underperformed relative to expectations, implying the problem was harder than anticipated at this step. If $d_n(x) < 0$, the model overperformed, suggesting the problem was easier.

Finally, to achieve the desired **stability**, we define a problem's stable difficulty, $D_{t_j}(x_j)$, as the centroid of its historical instantaneous difficulties, which smooths out noise over time. In practice, this is computed efficiently via an incremental update each time a problem x_j is sampled (for the t_j-th time):

$$D_{t_j}(x_j) = \frac{t_j - 1}{t_j} \cdot D_{t_j-1}(x_j) + \frac{1}{t_j} \cdot d_n(x_j) \tag{8}$$

Step 3: Quantifying Competence-Difficulty Alignment ($\mathcal{A}$) With definitions for both competence (C) and difficulty (D), we can now naturally quantify their alignment for any given problem. We define the alignment, $\mathcal{A}$, as the absolute difference between the model's competence and the problem's difficulty:

$$\mathcal{A}(x, M_n) = |C_{n-1} - D_{n-1}(x)| \tag{9}$$

This alignment value, $\mathcal{A}$, measures how closely a problem's difficulty matches the model's current learning frontier. A low $\mathcal{A}$ value indicates that a problem is well-matched to the model's present capabilities, making it a prime candidate for efficient and effective training.

Alignment-based Sampling and Training

Symmetric Sampling Strategy A naive approach to leverage our alignment metric might be to simply sample problems with the lowest absolute value of $\mathcal{A}(x, M_n)$. However, this could cause the sampling distribution to become overly biased. To foster a more balanced and robust training dynamic, we introduce a **Symmetric Sampling Strategy**. At each training step, we partition the entire problem set into two groups based on their relationship to the model's current competence:

- **The Slightly Harder Group (B^+):** Problems for which the difficulty is greater than the model's competence ($D_{n-1}(x) > C_{n-1}$). These are problems that currently lie just beyond the model's ability.
- **The Slightly Easier Group (B^-):** Problems for which the difficulty is less than or equal to the model's competence ($D_{n-1}(x) \leq C_{n-1}$). These are problems that are within the model's current grasp.

We then construct the training batch B by selecting the $|B|/2$ problems with the lowest alignment value $\mathcal{A}$ from each group, forming the final batch $B = B^- \cup B^+$. This balanced approach prevents the sampling process from being biased towards only one side. Further study of Symmetric Sampling Strategy is provided in Appendix.

CDAS in Reinforcement Learning: A Stable Fixed-Point System We integrate **CDAS** framework into the GRPO training loop, as detailed in Algorithm 1. We set the initial difficulty $D_0(x)$ for all problems and the initial model competence C_0 to 0, signifying no prior system knowledge.

Notably, the dynamic and interdependent update process, where competence and difficulty co-evolve, can be rigorously analyzed as a **Fixed-Point System**. As training progresses and the model's parameters converge, we expect the system to reach a stable equilibrium state (C^*, D^*, S^*), which satisfies the following conditions:

$$\begin{cases} D^*(x) = \sigma(C^* - D^*(x)) - S^*(x), & \forall x \in \mathcal{D} \\ C^* = -\mathbb{E}_x[D^*(x)] \end{cases} \tag{10}$$

Algorithm 1: Competence-Difficulty Alignment Sampling in GRPO

Input: Training set $\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$, Model M_0
Parameter: Total steps K, batch size $|B|$
Output: Updated model M_K

Initialize $C_0 = 0$, $t_j = 0$, $D_{t_j}(x_j) = 0$ for $j = 1, 2, \ldots, N$
for $n = 1$ to K **do**
  for $j = 1$ to N **do**
    $\mathcal{A}(x_j, M_{n-1}) \leftarrow |C_{n-1} - D_{t_j}(x)|$
  end for
  Sample $B = B^- \cup B^+$ based on $\mathcal{A}$ // GRPO Rollout
  for (x_j, y_j) in B **do**
    Compute pass rate $s_n(x_j)$
    Compute rewards r for each sampled response
    $t_j \leftarrow t_j + 1$
    $D_{t_j}(x_j) \leftarrow \frac{t_j - 1}{t_j} \cdot D_{t_j-1}(x_j) + \frac{1}{t_j} \cdot (\sigma(C_{n-1} - D_{t_j-1}(x_j)) - s_n(x_j))$
  end for
  $C_n \leftarrow -\mathbb{E}_x[D_{t_j}(x)]$
  Update policy model M_{n+1} by maximizing GRPO objective
end for
return M_K

Since the Sigmoid function is a contraction mapping, we can theoretically prove that this system is guaranteed to converge to a unique solution (proof provided in Appendix). The physical meaning of this equilibrium state provides important insight into the model's final capabilities:

- **Converged Pass Rate ($S^*(x)$):** Represents the stable, expected probability of solving problem x once the system stabilizes.
- **Converged Difficulty ($D^*(x)$):** Measures the gap between problem complexity and model capability. Positive values denote unsolved challenges; negative values denote mastery.
- **Converged Competence (C^*):** Represents the model's final, overall capability ceiling on the given task, averaged across all problems.

Experiments

Experimental Setup

Dataset Following Zeng et al. (2025), we utilize the MATH dataset (Hendrycks et al. 2021b) for RL training, including 7,500 training samples and 4,500 test samples. We reserve its MATH500 subset as the validation set for RL training.

Baselines We compare **CDAS** against a range of powerful baselines: (1) **Random Sampling**, which directly samples the problem randomly from the training dataset and can be viewed as plain GRPO training. (2) **Curriculum Sampling** (Team et al. 2025), which uniformly samples easy problems first, then turns to harder ones. Here, we use the difficulty level tags included in the MATH dataset, ranging from 1 to 5, and conduct training from the middle checkpoint in Random Sampling on problems with difficulty level ≥ 4. (3) **Prioritized Sampling**, which is adopted in Kimi k1.5 (Team et al. 2025) that tracks the pass rate s of each problem and then samples problems based on $1 - s$.(4) **Dynamic Sampling**, adopted in DAPO (Yu et al. 2025), which over-samples and filters out problems with the pass rate equal to 1 or 0.

Training Details We experiment with GRPO, the most widely-used algorithm for LLM RL training, on Qwen2.5-7B (Yang et al. 2024) using the veRL framework (Sheng et al. 2024). Following the training recipe in SimpleRL-Zoo (Zeng et al. 2025), we set the batch size $|B| = 1024$, generating 8 rollouts for each problem with temperature 1.0 and maximum response tokens 4096. Refer to Appendix for more training details.

Considering the stability of our iterative system, we adopt a **warm-up strategy** in **CDAS**. For the duration of the first training epoch, we sample problems uniformly at random. This ensures every problem is sampled at least once, allowing us to establish a robust initial value for each problem's difficulty. By doing so, we directly mitigate the risk of amplifying dataset biases before the main adaptive sampling phase begins. Further analysis is available in Appendix.

Evaluation For a comprehensive evaluation, we select 6 mathematical reasoning tasks, including AIME 24&25, MATH500 (Hendrycks et al. 2021b), Minerva Math (Lewkowycz et al. 2022), Olympiad Bench (He et al. 2024) and GSM8K (Cobbe et al. 2021). Since the number of problems in AIME24 and AIME25 is relatively small, we report the Avg@32 metric to reduce randomness. Standard accuracy is reported for the remaining tasks. More details are shown in Appendix.

Main Results

The main results are summarized in Table 1, with training curves shown in Figure 4 and performance comparison curves shown in Figure 2. Besides the final checkpoint, we also report the intermediate checkpoint at the 55-th step. From the results, we have the following findings:

CDAS outperforms plain GRPO and other baselines. As shown in Table 1, CDAS achieves the best average accuracy at both the 55-th step (44.51%) and the 110-th step (45.89%) checkpoints. Meanwhile, we find that the improvements offered by Curriculum Sampling (+0.35% at the 110-th step) and Prioritized Sampling (+0.57% at the 110-th step) over the Random Sampling baseline are limited. This indicates that relying solely on prior difficulty labels or single-step pass rates tends to be suboptimal.

CDAS demonstrates a substantial efficiency advantage. In terms of sample efficiency, **CDAS** achieves a performance of 44.51% at just 55 training steps, surpassing the final performance of Random Sampling (44.09% at 110 steps), reaching comparable capability in half the training time. Moreover, as shown in Figure 2(b), the training time of **CDAS** is comparable to most standard baselines. This stands in stark contrast to Dynamic Sampling, which incurs significantly higher computational costs (2.33× slower).

The reward curve in CDAS demonstrates a dynamic process. As shown in Figure 4, the reward curve for **CDAS** exhibits four distinct phases: (1) **Warm-up Phase:** During the first epoch, the curve mirrors that of Random Sampling, as problems are sampled uniformly to establish an initial, unbiased difficulty assessment. (2) **Rapid Ascent Phase:** Immediately following the warm-up, the reward sharply increases to over 0.8. This is because **CDAS** identifies and prioritizes the problems that have become easy for the model. (3) **Adaptive Descent Phase:** As the model's competence grows, **CDAS** intentionally transitions to sampling more challenging problems that better match the model's new capabilities. This strategic shift causes the average reward to decrease. (4) **Convergence Phase:** The curve finally stabilizes around a median reward of approximately 0.55, which signifies that **CDAS** continuously presents the model with problems within its competence.

Analysis and Discussion

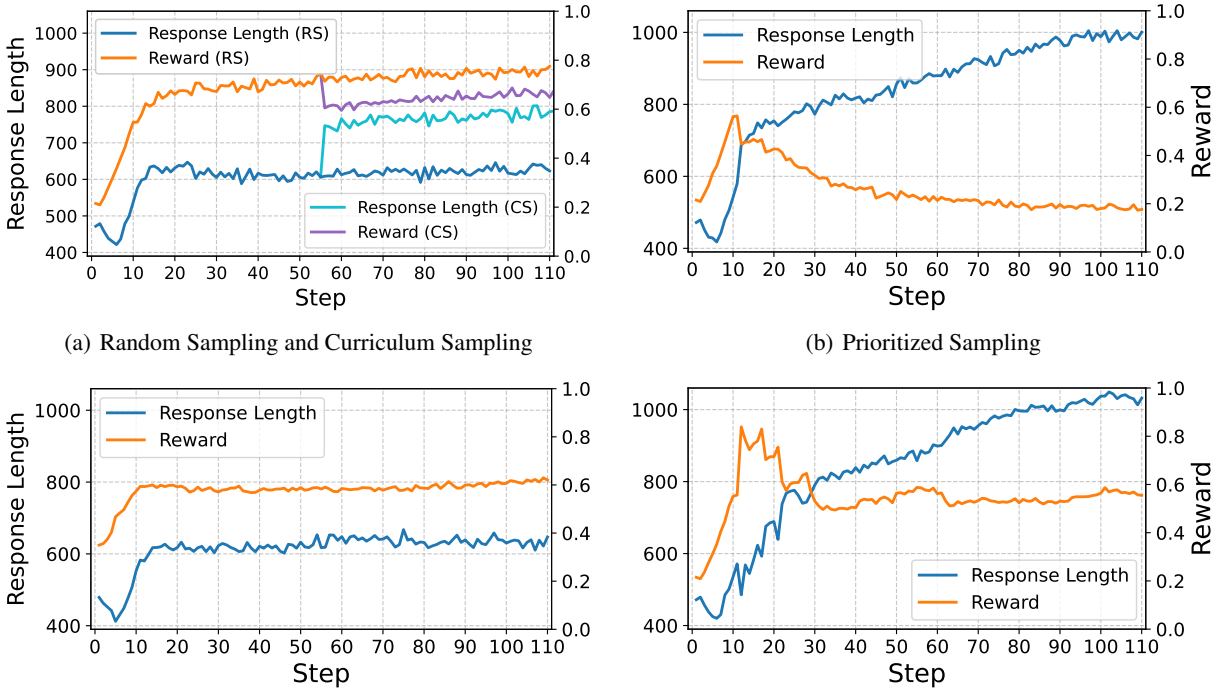
In this section, we conduct a comprehensive analysis and discussion of **CDAS**. We explore the statistical properties of **CDAS** in terms of sample utility, investigate its advantages over pass rate-based methods, and validate its generalization to code generation tasks along with different architectures and model sizes.

Utility of the Sampled Problems

From the perspective of the optimization objective of GRPO, the superior performance of Dynamic Sampling can be at-

Steps	Method	AIME24 (Avg@32)	AIME25 (Avg@32)	MATH 500	Minerva Math	Olympiad Bench	GSM8K	Avg.
0	-	0.06	0.00	47.00	17.65	14.52	81.50	24.38
55	Random Sampling	10.00	7.29	74.20	36.76	39.26	91.66	43.20
	Curriculum Sampling	10.00	7.29	74.20	36.76	39.26	91.66	43.20
	Prioritized Sampling	11.77	7.08	74.00	**39.34**	40.15	92.04	44.06
	Dynamic Sampling	**12.19**	**7.92**	75.00	38.97	38.52	91.36	43.99
	CDAS (Ours)	11.56	7.71	**76.20**	38.97	**40.44**	**92.19**	**44.51**
110	Random Sampling	12.29	6.98	75.20	37.13	40.00	**92.95**	44.09
	Curriculum Sampling	12.71	7.19	76.00	38.60	39.56	92.57	44.44
	Prioritized Sampling	**15.10**	9.27	75.00	37.50	39.56	91.51	44.66
	Dynamic Sampling	**15.10**	9.58	**77.20**	39.34	39.56	92.34	45.52
	CDAS (Ours)	14.90	**11.77**	75.40	**40.44**	**40.89**	91.96	**45.89**

Table 1: Performance comparison across different sampling methods on various math benchmarks. Metrics are Avg@32 for AIME and standard accuracy for others. We present the best results in **bold** and the second with underline.



(a) Random Sampling and Curriculum Sampling

(b) Prioritized Sampling

(c) Dynamic Sampling

(d) CDAS

Figure 4: Training curves of different sampling strategies.

tributed to its filtering out of samples that do not contribute to model gradients (Yu et al. 2025) (i.e., those with a pass rate of 0 or 1). Although **CDAS** does not explicitly constrain the pass rate in problem selection, its alignment-based symmetric sampling inherently mitigates the issue of oversampling the zero-gradient problems to some extent. As illustrated in Figure 5, the proportion of such zero-gradient problems within the batches sampled by **CDAS** is consistently lower than that of the other baselines, proving that **CDAS** can effectively improve the utility of sampled problems. We also observe that the proportion of zero-gradient problems in **CDAS** exhibits a rapid decline during the early stages of training, followed by a slight increase in the later stages. The sharp decrease in the initial phase can be attributed to the swift correction of problem difficulty from its initial values. The modest rise in the later phase is mainly due to the increasing proportion of zero-gradient problems in the whole MATH training set, leading to more problems with a pass rate of 0 sampled in the batch B^+.

Problem Difficulty vs. Pass Rate

Since the problem difficulty in **CDAS** derived from the pass rates, we explore the relationship between them. As shown in Figure 6, problem difficulty and pass rate exhibit an over-

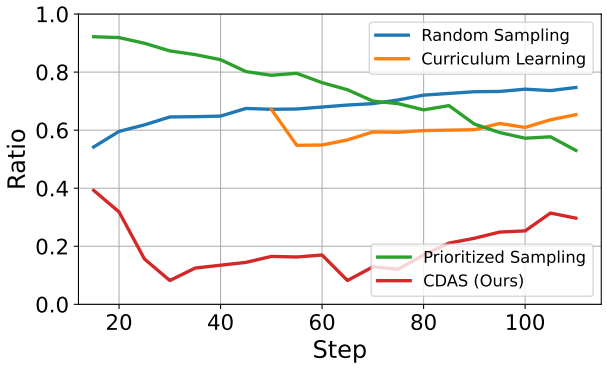


Figure 5: The proportion of zero-gradient problems in the sampled batch.

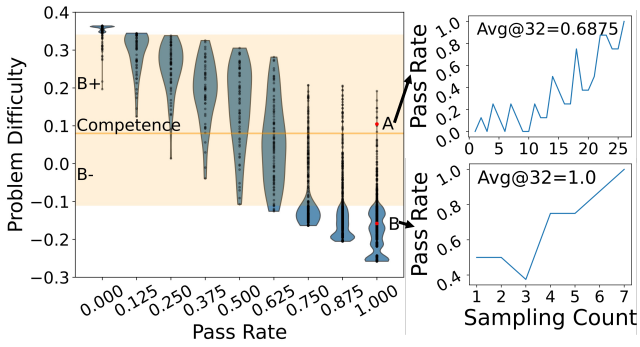


Figure 6: Problem difficulty vs. pass rate in **CDAS**.

all negative correlation where problems with lower difficulty tend to have higher pass rates. Interestingly, we find that even among problems with the same pass rate, there can still be considerable differences in their estimated difficulties. To further investigate this phenomenon, we randomly selected two problems with a pass rate of 1 at the final sampling step. As shown in Figure 6, problem A required 25 samplings to reach a pass rate of 1, whereas problem B achieved a pass rate of 1 after only 6 samplings. Despite both having the same final pass rate in rollout, Problem A is noticeably more difficult than Problem B, as indicated by its average accuracy of 32 inferences being 0.6875, which is much lower than that of Problem B (Avg@32 = 1.0). This validates that **CDAS**, by leveraging historical information, provides a more accurate and robust measure of problem difficulty.

Generalization

We further investigate the generalization capabilities of **CDAS** across different tasks, model architectures, and model sizes. We compare our method against two representative sampling strategies: **Random Sampling (RS)**, the most widely used baseline, and **Dynamic Sampling (DS)**, a strong but computationally expensive performance baseline.

Generalization to Code Generation Tasks We apply **CDAS** to the code generation task. Specifically, we aggregate 10k open-source competitive programming problems from Apps (Hendrycks et al. 2021a), Taco (Li et al. 2023), and CodeContests (Li et al. 2022), and perform GRPO training on Qwen2.5-Coder-7B (Hui et al. 2024) for 100 steps. As shown in Table 2, CDAS achieves **18.24%** Pass@1 accuracy on LiveCodeBench v5 (Jain et al. 2025), outperforming RS by **+3.00%** and nearly matching DS. This verifies **CDAS**'s strong generalization to different tasks. We leave the comparison of other tasks for future work.

Method	Pass@1 Acc.
RS	15.24
DS	**18.32**
CDAS	<u>18.24</u>

Table 2: Accuracy comparison on LiveCodeBench v5.

Generalization to Different Architectures and Model Sizes To further examine the generalization capability of **CDAS** across architectures and scales, we conduct additional RL experiments on Qwen2.5-14B (Yang et al. 2024) and OctoThinker-8B-Hybrid (Wang et al. 2025). Both models are trained on the MATH dataset used in the main experiments. For Qwen2.5-14B, we train for 200 steps with a batch size of 256, while OctoThinker-8B-Hybrid adopts the same hyperparameter configuration as the main setup. As shown in Table 3, **CDAS** consistently outperforms the RS baseline and achieves performance that matches or exceeds the strong DS baseline on both models, while maintaining comparable training efficiency to RS. These results demonstrate the robust generalization of CDAS to models of larger scale and different architectural designs.

Model	Method	Average Acc.
Qwen2.5-14B	RS	46.28
	DS	<u>46.62</u>
	CDAS	**47.22**
OctoThinker-8B	RS	35.30
	DS	<u>36.47</u>
	CDAS	**36.52**

Table 3: Generalization Performance Across Different Architectures and Model Sizes.

Conclusion

We present **C**ompetence-**D**ifficulty **A**lignment **S**ampling (**CDAS**), a novel sampling strategy for RL training in LLM reasoning. **CDAS** addresses the limitations of existing methods by modeling problem difficulty as a trajectory of performance discrepancies to provide more stable estimations and explicitly aligning it with model competence at each training step throughout a fixed-point system. Extensive experiments on mathematical reasoning benchmarks demonstrate the superiority of **CDAS** in both accuracy and efficiency to powerful baselines. Our results highlight the importance of dynamically matching problem difficulty to model competence for efficient RL training.

References

Bae, S.; Hong, J.; Lee, M. Y.; Kim, H.; Nam, J.; and Kwak, D. 2025. Online difficulty filtering for reasoning oriented reinforcement learning. *arXiv preprint arXiv:2504.03380*.

Baker, F. B. 2001. *The basics of item response theory*. ERIC.

Bengio, Y.; Louradour, J.; Collobert, R.; and Weston, J. 2009. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, 41–48.

Chu, X.; Huang, H.; Zhang, X.; Wei, F.; and Wang, Y. 2025. GPG: A Simple and Strong Reinforcement Learning Baseline for Model Reasoning. *arXiv preprint arXiv:2504.02546*.

Cobbe, K.; Kosaraju, V.; Bavarian, M.; Chen, M.; Jun, H.; Kaiser, L.; Plappert, M.; Tworek, J.; Hilton, J.; Nakano, R.; et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Gao, L.; Schulman, J.; and Hilton, J. 2023. Scaling laws for reward model overoptimization. In *International Conference on Machine Learning*, 10835–10866. PMLR.

Guo, D.; Yang, D.; Zhang, H.; Song, J.; Zhang, R.; Xu, R.; Zhu, Q.; Ma, S.; Wang, P.; Bi, X.; et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

He, C.; Luo, R.; Bai, Y.; Hu, S.; Thai, Z. L.; Shen, J.; Hu, J.; Han, X.; Huang, Y.; Zhang, Y.; et al. 2024. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. *arXiv preprint arXiv:2402.14008*.

Hendrycks, D.; Basart, S.; Kadavath, S.; Mazeika, M.; Arora, A.; Guo, E.; Burns, C.; Puranik, S.; He, H.; Song, D.; et al. 2021a. Measuring coding challenge competence with apps. *arXiv preprint arXiv:2105.09938*.

Hendrycks, D.; Burns, C.; Kadavath, S.; Arora, A.; Basart, S.; Tang, E.; Song, D.; and Steinhardt, J. 2021b. Measuring Mathematical Problem Solving With the MATH Dataset. *NeurIPS*.

Hui, B.; Yang, J.; Cui, Z.; Yang, J.; Liu, D.; Zhang, L.; Liu, T.; Zhang, J.; Yu, B.; Lu, K.; et al. 2024. Qwen2. 5-coder technical report. *arXiv preprint arXiv:2409.12186*.

Jaech, A.; Kalai, A.; Lerer, A.; Richardson, A.; El-Kishky, A.; Low, A.; Helyar, A.; Madry, A.; Beutel, A.; Carney, A.; et al. 2024. Openai o1 system card. *arXiv preprint arXiv:2412.16720*.

Jain, N.; Han, K.; Gu, A.; Li, W.-D.; Yan, F.; Zhang, T.; Wang, S.; Solar-Lezama, A.; Sen, K.; and Stoica, I. 2025. LiveCodeBench: Holistic and Contamination Free Evaluation of Large Language Models for Code. In *The Thirteenth International Conference on Learning Representations*.

Le, T.-L. V.; Jeon, M.; Vu, K.; Lai, V.; and Yang, E. 2025. No prompt left behind: Exploiting zero-variance prompts in llm reinforcement learning via entropy-guided advantage shaping. *arXiv preprint arXiv:2509.21880*.

Lewkowycz, A.; Andreassen, A.; Dohan, D.; Dyer, E.; Michalewski, H.; Ramasesh, V.; Slone, A.; Anil, C.; Schlag, I.; Gutman-Solo, T.; et al. 2022. Solving quantitative reasoning problems with language models. *Advances in Neural Information Processing Systems*, 35: 3843–3857.

Li, R.; Fu, J.; Zhang, B.-W.; Huang, T.; Sun, Z.; Lyu, C.; Liu, G.; Jin, Z.; and Li, G. 2023. Taco: Topics in algorithmic code generation dataset. *arXiv preprint arXiv:2312.14852*.

Li, Y.; Choi, D.; Chung, J.; Kushman, N.; Schrittwieser, J.; Leblond, R.; Eccles, T.; Keeling, J.; Gimeno, F.; Dal Lago, A.; Hubert, T.; Choy, P.; de Masson d'Autume, C.; Babuschkin, I.; Chen, X.; Huang, P.-S.; Welbl, J.; Gowal, S.; Cherepanov, A.; Molloy, J.; Mankowitz, D.; Sutherland Robson, E.; Kohli, P.; de Freitas, N.; Kavukcuoglu, K.; and Vinyals, O. 2022. Competition-Level Code Generation with AlphaCode. *arXiv preprint arXiv:2203.07814*.

Liu, Z.; Chen, C.; Li, W.; Qi, P.; Pang, T.; Du, C.; Lee, W. S.; and Lin, M. 2025. Understanding r1-zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*.

Lord, F. M.; and Novick, M. R. 2008. *Statistical theories of mental test scores*. IAP.

Narvekar, S.; Peng, B.; Leonetti, M.; Sinapov, J.; Taylor, M. E.; and Stone, P. 2020. Curriculum learning for reinforcement learning domains: A framework and survey. *Journal of Machine Learning Research*, 21(181): 1–50.

Peng, B.; Song, L.; Tian, Y.; Jin, L.; Mi, H.; and Yu, D. 2023. Stabilizing RLHF through advantage model and selective rehearsal. *arXiv preprint arXiv:2309.10202*.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Shao, Z.; Wang, P.; Zhu, Q.; Xu, R.; Song, J.; Bi, X.; Zhang, H.; Zhang, M.; Li, Y.; Wu, Y.; et al. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Sheng, G.; Zhang, C.; Ye, Z.; Wu, X.; Zhang, W.; Zhang, R.; Peng, Y.; Lin, H.; and Wu, C. 2024. HybridFlow: A Flexible and Efficient RLHF Framework. *arXiv preprint arXiv: 2409.19256*.

Singh, A.; Co-Reyes, J. D.; Agarwal, R.; Anand, A.; Patil, P.; Garcia, X.; Liu, P. J.; Harrison, J.; Lee, J.; Xu, K.; et al. 2023. Beyond human data: Scaling self-training for problem-solving with language models. *arXiv preprint arXiv:2312.06585*.

Team, K.; Du, A.; Gao, B.; Xing, B.; Jiang, C.; Chen, C.; Li, C.; Xiao, C.; Du, C.; Liao, C.; et al. 2025. Kimi k1. 5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599*.

Wang, E.; Cassano, F.; Wu, C.; Bai, Y.; Song, W.; Nath, V.; Han, Z.; Hendryx, S.; Yue, S.; and Zhang, H. 2024. Planning in natural language improves llm search for code generation. *arXiv preprint arXiv:2409.03733*.

Wang, Z.; Zhou, F.; Li, X.; and Liu, P. 2025. OctoThinker: Mid-training Incentivizes Reinforcement Learning Scaling. *arXiv preprint arXiv:2506.20512*. Preprint.

Weng, L. 2024. Reward Hacking in Reinforcement Learning. *lilianweng.github.io*.

Yang, A.; Yang, B.; Zhang, B.; Hui, B.; Zheng, B.; Yu, B.; Li, C.; Liu, D.; Huang, F.; Wei, H.; et al. 2024. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*.

Yu, Q.; Zhang, Z.; Zhu, R.; Yuan, Y.; Zuo, X.; Yue, Y.; Fan, T.; Liu, G.; Liu, L.; Liu, X.; et al. 2025. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*.

Zeng, W.; Huang, Y.; Liu, Q.; Liu, W.; He, K.; Ma, Z.; and He, J. 2025. Simplerl-zoo: Investigating and taming zero reinforcement learning for open base models in the wild. *arXiv preprint arXiv:2503.18892*.

Zheng, R.; Dou, S.; Gao, S.; Hua, Y.; Shen, W.; Wang, B.; Liu, Y.; Jin, S.; Liu, Q.; Zhou, Y.; et al. 2023. Secrets of rlhf in large language models part i: Ppo. *arXiv preprint arXiv:2307.04964*.