Scale Down to Speed Up: Dynamic Data Selection for Reinforcement Learning

Zhuoyue Chen^{1,2*}, Jihai Zhang^{2*}, Ben Liu^{3}, Fangquan Lin^{2}, Wotao Yin^{2†}
[1]Zhejiang University [2]DAMO Academy, Alibaba group [3]Wuhan University
chenzhuoyue@zju.edu.cn,
{jihai.zjh, fangquan.linfq, wotao.yin}@alibaba-inc.com,
liuben123@whu.edu.cn

Abstract

Optimizing data utilization remains a central challenge in applying Reinforcement Learning (RL) to Large Language Models (LLMs), directly impacting sample efficiency, training stability, and final model performance. Current approaches often rely on massive static datasets, leading to computational inefficiency and redundant gradient updates. In this paper, we propose **ScalingRL**, a data-centric RL framework that dynamically selects the most informative training samples to optimize RL for mathematical reasoning. Specifically, **ScalingRL** introduces the Data Effectiveness Score (DES) that quantitatively ranks prompts according to three complementary factors: problem difficulty, Chain-of-Thought complexity, and reward adaptability. Then, **ScalingRL** employs an adaptive curriculum scheduler that progressively adjusts the overall scale and specific mix of training prompts—balancing exploration of new, challenging data with exploitation of previously learned concepts—thereby tailoring the data distribution to the model's current learning trajectory and performance. Experimental results demonstrate that **ScalingRL** achieves comparable performance to full-data training methods while requiring only 1.5K samples instead of 220K, reducing training time from 13 days to just 4 hours on 8×A800 GPUs.

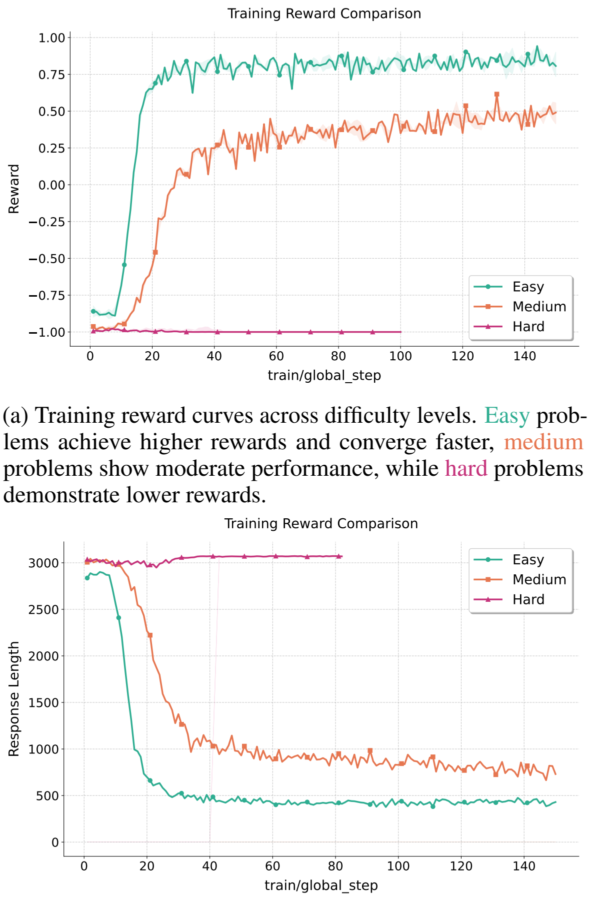
1 Introduction

Recently, R1-like reasoning models (Guo et al., 2025; Team et al., 2025) have attracted significant attention for their unprecedented performance in mathematical reasoning, demonstrating spontaneous emergence of long Chain-of-Thought (CoT) (Wei et al., 2022; Chen et al., 2025a; Liu et al., 2025a) and self-reflection behaviors (He et al., 2024b; Zhang et al., 2024). The central technique driving this revolution is large-scale Reinforcement Learning (RL) (Kaelbling et al., 1996), such as Proximal Policy Optimization (PPO) (Schulman et al., 2017) and Group Relative Policy Optimization (GRPO) (Shao et al., 2024), which enable models to develop sophisticated reasoning capabilities through dynamic reward-based training. However, existing RL training paradigms remain highly sensitive to sample quality and distribution (Liu et al., 2025b; Li et al., 2025), leading to gradient instability issues that fundamentally limit the effectiveness of simply scaling up training data volume.

As shown in Figure 1a, the characteristics of training samples significantly influence RL-based reasoning optimization. In particular, the **difficulty level** of the training dataset plays a crucial role in shaping the model's learning trajectory. excessively difficult examples often lead to uniformly incorrect outputs, providing little informative signal for effective policy updates. Conversely, as model capabilities improve, excessively simple problems dilute the informative gradient needed for meaningful policy improvements, thereby reducing sample efficiency (Zhang and Zuo, 2025; Xiong et al., 2025; Liu et al., 2024a). Additionally, due to the surrogate loss function employed in current RL algorithms, **the length of responses** inevitably affects policy gradient updates. In Figure 1b, we identify the optimization bias in current RL algorithms that may lead to progressively longer incorrect responses. Motivated by these considerations, we aim to explore **a data-centric approach** to dynamically schedule RL training samples, substantially enhancing sample efficiency and achieving comparable or superior performance with fewer training samples.

Most existing research efforts focus on enhancing RL training performance through algorithmic innovations (Hu, 2025; Zhang et al., 2025a) or more sophisticated reward models (Liu et al., 2025c), while the data-centric dimension of RL

[⋆]Equal Contribution
[†]Corresponding Author



(a) Training reward curves across difficulty levels. Easy problems achieve higher rewards and converge faster, medium problems show moderate performance, while hard problems demonstrate lower rewards.

(b) Response length evolution during training. Hard problems require longer responses compared to medium and easy problems.

Figure 1: Training dynamics across problem difficulties showing reward performance (a) and response length (b). Results indicate performance decreases as problem complexity increases.

remains notably underexplored. LIMR (Li et al., 2025) introduces the "less is more" concept, demonstrating that RL training on a small but informative subset can match the performance of full-dataset training, highlighting the potential of data-centric perspectives. However, LIMR is fundamentally resource-intensive: it requires complete RL training on the entire dataset before evaluating the effectiveness of individual samples, which undermines its efficiency gains by incurring substantial computational cost and time upfront. Additionally, LIMR exclusively considers reward signals while neglecting critical factors such as difficulty and length of responses, resulting in suboptimal training data selection.

To address these issues, we propose **ScalingRL**, an efficient data-centric framework, which combines progressive data scaling with dynamic sample effectiveness scoring to optimize LLM reinforcement learning. Specifically, we introduce Data Effectiveness Score (DES), an automated quantitative method for evaluating the potential value of RL training samples. DES is a composite function that integrates three key components: difficulty coefficient, CoT complexity, and reward adaptability. This multifaceted approach ensures a comprehensive evaluation of each data sample's utility in the training process. Based on the DES score, ScalingRL implements an adaptive training scheduling strategy that dynamically selects the most informative samples according to the model's evolving capabilities across different training stages. This automatic curriculum adaptation ensures that the training process always prioritizes samples that will contribute most significantly to model improvement, effectively creating a self-optimizing learning schedule that maximizes information gain throughout the entire RL process. Our main contributions can be summarized as follows:

- We introduce an effective data evaluation mechanism that systematically assesses and curates training samples based on their intrinsic properties, particularly difficulty and response length characteristics.

- We develop an adaptive training scheduler that leverages our data evaluation mechanism to dynamically optimize sample selection throughout the training process. This training strategy automatically adjusts the composition of training batches based on the model's evolving capabilities, ensuring that each training phase utilizes the informative samples for continued improvement.

- Experimental results demonstrate that ScalingRL achieves comparable performance on automatically selected 1.5K samples to full-data (220K samples) RL training methods across multiple mathematical reasoning benchmarks. This significant data reduction enables ScalingRL to reduce training time from 13 days to just 4 hours on 8×NVIDIA A800 GPUs without compromising effectiveness.

2 Related work

2.1 Reinforcement Learning for LLMs

Reinforcement Learning (RL) has been increasingly integrated into the training of large language models for enhancing mathematical reasoning capabilities. Starting with PPO (Schulman et al., 2017), which provided a stable optimization framework, the field rapidly evolved with specialized algorithms like GRPO (Shao et al., 2024) that simplified implementation by eliminating separate critic networks while maintaining performance.

Subsequent refinements addressed specific challenges in language model optimization: Dr.GRPO (Liu et al., 2025b) incorporated explicit length considerations to balance thoroughness with conciseness, while DAPO (Yu et al., 2025) introduced dynamic sampling techniques. Recent advancements include value function optimizations (VinePPO (Kazemnejad et al., 2024), VCPPO (Yuan et al., 2025b), VAPO (Yuan et al., 2025a)), stabilization techniques (SRPO (Zhang et al., 2025b), CPG (Chu et al., 2025)), and integrated approaches like REINFORCE++ (Hu, 2025) that combine multiple components. These developments collectively demonstrate both the potential and ongoing challenges of applying reinforcement learning to mathematical reasoning tasks.

2.2 Data Selection for LLM Post-Training

The quality and relevance of training data are crucial determinants of large language models' performance. Data selection for LLM post-training has been extensively studied in prior work (Kumar et al., 2025), with most efforts focusing on supervised fine-tuning. These approaches include LLM-based quality assessment (Ivison et al., 2025), leveraging features from model computation (Chen et al., 2023), gradient-based selection (Ivison et al., 2022), and other methodologies (Xia et al., 2024). A parallel line of research (Das et al., 2024; Muldrew et al., 2024; Liu et al., 2024b) explores data selection for human preference data in Reinforcement Learning from Human Feedback (RLHF). Traditional approaches often emphasize the quantity of data, operating under the assumption that more data leads to better model performance. However, recent studies (Ye et al., 2025; Li et al., 2025; Muennighoff et al., 2025) advocate for the principle of "less is more," highlighting that carefully curated and high-quality data can achieve superior results with fewer samples.

Notably, LIMR (Li et al., 2025) proposes selecting training samples aligned with the model's learning trajectory, demonstrating that a subset of 1,389 samples can surpass the performance of the full 8,523-sample dataset. However, LIMR requires first completing RL training on the entire dataset to assess individual sample utility—an approach that incurs significant upfront computational cost and limits scalability for large-scale settings. Moreover, LIMR focuses solely on reward-based metrics, overlooking other informative factors such as problem difficulty and response length. In contrast, our approach integrates these additional dimensions into an efficient, adaptive sample selection framework, enabling substantial reductions in computational resources while preserving training effectiveness.

3 Methodology

In this section, we introduce our proposed **ScalingRL** framework. As illustrated in Figure 2, **ScalingRL** consists of four key components: (1) Initial Candidate Construction, which refines a diverse 220K mathematics dataset to a curated 8K candidate set through quality, diversity, and difficulty filtering; (2) Dynamic Data Effectiveness Scoring, a novel metric combining difficulty coefficient, Chain-of-Thought complexity, and reward adaptability to evaluate sample utility; (3) Dynamic Sampling Strategy that implements temperature-controlled sample selection to balance exploration and exploitation; and (4) Reinforcement Learning with PPO algorithm and structured reward functions for model optimization. Through this progressive refinement strategy, we significantly reduce training time from 13 days to 4 hours while maintaining model performance with 8×NVIDIA A800 GPUs.

3.1 Initial Candidate Construction

We begin with an extensive initial dataset comprising 220K[1] mathematics-specific problem-answer pairs collected from AIME (1984-2023), AMC (prior to 2023), Mathematical Olympiads, and Art of Problem Solving (AoPS) forum. While this substantial dataset could be directly used for training, we hypothesize that a carefully selected subset of high-quality examples might achieve comparable or even superior results. Thus, we finally select 8K samples focusing on three critical dimensions:

- **Quality**: To maintain the highest data quality standards, we conduct a comprehensive data cleaning and validation process. First, we

[1] https://huggingface.co/datasets/open-r1/OpenR1-Math-220k

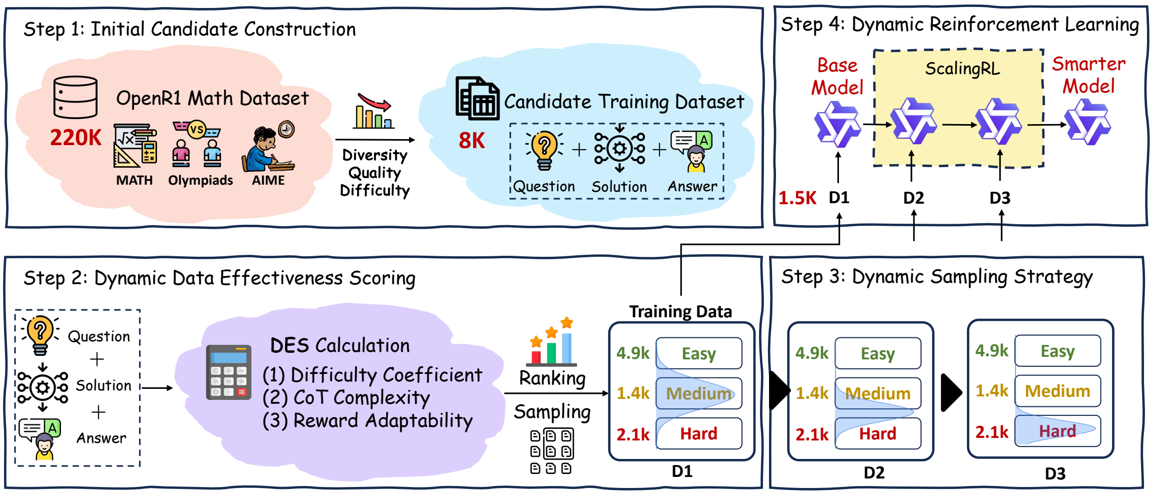


Figure 2: Overview of our data sampling and training pipeline. The process starts with a diverse 220K math dataset from multiple sources (MATH, Olympiads, AIME). This is filtered to an 8K candidate set based on diversity and quality metrics. The training then proceeds through multiple epochs, where each epoch contains the same distribution of 8.4K samples (4.9K easy, 1.4K medium, 2.1K hard) but with dynamic sampling within each difficulty level based on our effectiveness scoring mechanism. This maintains consistent difficulty ratios while optimizing sample selection across epochs.

eliminate samples containing errors or missing fields in their reasoning traces. Then, we remove questions with ambiguous problem statements or incorrect mathematical expressions. Finally, we retain only those examples that demonstrate clear problem formulation, rigorous mathematical notation, and complete solution procedures, thereby establishing a clean and reliable training dataset.

- **Diversity**: We categorize problems into distinct mathematical domains (such as geometry, algebra, number theory, etc.), allowing us to maintain a balanced distribution across different problem types and ensure comprehensive coverage of mathematical concepts.
- **Difficulty**: We assess problem difficulty through an inference-based approach, adapting the s1K (Muennighoff et al., 2025) methodology to better suit our computational constraints. Unlike s1K, while employs large-scale models for difficulty evaluation, we use *Qwen2.5-Math-1.5B* as our assessment baseline. Problems correctly solved by this baseline are deemed trivial and excluded from the candidate set. In addition, we filter out problems whose reasoning traces exceed 8,192 tokens, as such excessively long solutions are computationally impractical for our targeted efficient training regime.

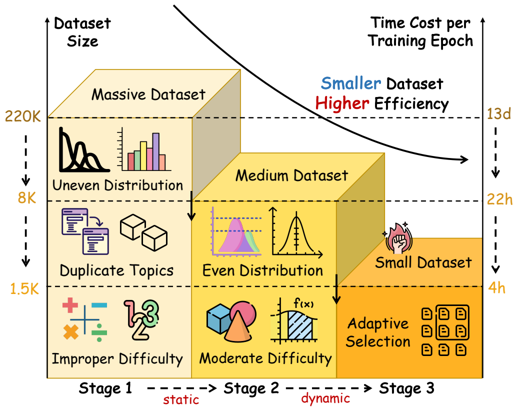


Figure 3: Overview of our three-stage data scaling methodology. Stage 1 processes the initial massive dataset (220K samples) characterized by uneven distribution, duplicate topics, and predominantly easy questions. Stage 2 refines this to a medium-sized dataset (8K samples) with more balanced distribution and moderate difficulty. Stage 3 implements dynamic sampling with a small, highly effective dataset (1.5K samples). The time cost per training epoch decreases significantly from 13 days to 4 hours through this progressive scaling.

The outcome of the data filtering process is illustrated in Figure 3 (from stage 1 to stage 2), which elucidates our progressive data refinement strategy. This multi-stage filtering approach encompasses three key criteria: quality, difficulty, and diversity. We distill the initial dataset into a carefully curated set of 8K samples, which forms the basis for our subsequent dynamic training process.

3.2 Dynamic Data Effectiveness Scoring

During training process, not all data samples contribute equally to model updating. Some problems may be too simple to provide meaningful learning signals, while others might be so complex that they hinder efficient training. Moreover, the effectiveness of a training sample may vary during different stages of the learning process.

To effectively manage and utilize training dataset, we introduce a novel metric called *Data Effectiveness Score* (DES). DES synthesizes three complementary components: difficulty coefficient $D(x)$, Chain-of-Thought (CoT) complexity $C(x)$, and reward adaptability $R(x)$. Formally, DES is defined as a weighted combination of these components:

$$\text{DES}(x) = \lambda_d \, D(x) + \lambda_c \, C(x) + \lambda_r \, R(x), \quad (1)$$

where the weighting coefficients $\{\lambda_d, \lambda_c, \lambda_r\}$ control the relative importance of each component. These coefficients are dynamically adjusted during training to optimize the overall learning effectiveness. To maintain valid probability distribution properties ($\sum_i \lambda_i = 1, \lambda_i > 0$), we parameterize the weights via a softmax transformation:

$$\lambda_i = \frac{e^{\alpha_i}}{\sum_{j \in \{d,c,r\}} e^{\alpha_j}}, \quad i \in \{d, c, r\} \quad (2)$$

where $(\alpha_d, \alpha_c, \alpha_r)$ are learnable real-valued parameter.

3.2.1 Difficulty Coefficient $D(x)$

The *Difficulty Coefficient* $D(x)$ quantitatively measures the complexity of a math problem relative to the dataset's overall difficulty, and adaptively shifts the target difficulty level throughout training. It is defined as:

$$D(x) = \exp\left(-\frac{\left[d_x - d_{\text{avg}}\left(1 + \alpha(1 - w_t)\right)\right]^2}{2\sigma_d^2}\right), \quad (3)$$

where d_x represents the difficulty score of current problem (we utilize DeepSeek-R1 (Guo et al., 2025) to answer the question multiple times and define the difficulty score based on the accuracy rate. For a detailed explanation of the scoring mechanism, please refer to Appendix A), d_{avg} denotes the average difficulty score across the dataset, σ_d represents the standard deviation and $w_t \in [0, 1]$ is a time-dependent weight that evolves during training process. The weight w_t is linearly decayed from 1 to 0 as training progresses. At the start of training, w_t is set to 1, favoring scores where the value of d_x is closer to d_{avg}. Over time, we aim to select questions that are slightly more challenging than d_{avg}, in alignment with normal cognitive patterns, where learning begins with simpler problems and gradually progresses towards overcoming more difficult ones. The coefficient $\alpha > 0$ controls the adaptive difficulty adjustment rate, determining how much the target difficulty can deviate from the average difficulty level during training.

3.2.2 Chain-of-Thought Complexity $C(x)$

Instead of relying solely on problem difficulty, we evaluate the complexity of solution reasoning paths through multiple sampling iterations from DeepSeek-R1 (Guo et al., 2025). The *Chain-of-Thought Complexity* $C(x)$ measures the intricacy of the reasoning process through a Gaussian-based formulation:

$$C(x) = \exp\left(-\frac{\left[l_x - l_{\text{avg}}\left(1 + \beta(1 - w_t)\right)\right]^2}{2\sigma_c^2}\right), \quad (4)$$

where l_x and l_{avg} respectively represents the current and average solution length, and σ^2 controls the sensitivity to length deviations. Similar to $D(x)$, $C(x)$ adapts during training through the time-dependent weight w_t and the adaptation rate β, progressively favoring more sophisticated reasoning chains. Gaussian-based formulation ensures that problems with appropriate reasoning complexity receive higher scores.

3.2.3 Reward Adaptability $R(x)$

The *Reward Adaptability* $R(x)$ captures how well the rewards associated with a data sample have adapted over training epochs. Inspired by the gradient-decreasing problem observed in existing RL algorithms (as noted in DAPO (Yu et al., 2025)), we specifically design our reward mechanism to maintain effective learning signals. When samples consistently achieve perfect accuracy (reward = 1) or complete failure (reward = -1), they provide minimal gradient information for model improvement. Therefore, we modify our reward formulation as:

$$R(x) = \begin{cases} 0, & \bar{r}_x \in \{-1, 1\}, \\ \exp\left(-\frac{\left[\bar{r}_x - r_{\text{avg}}(1 + \gamma(1 - w_t))\right]^2}{2\sigma_r^2}\right), & \text{otherwise}, \end{cases} \quad (5)$$

where $\bar{r}_x$ represents the average reward over recent training epochs for x, r_{avg} denotes the mean reward across all samples, and γ controls the adaptive rate of target reward. By explicitly filtering out samples with extreme rewards (-1 or 1), we ensure that training focuses on problems that provide meaningful gradient signals, thereby promoting more efficient and stable learning dynamics.

3.3 Dynamic Sampling Strategy with Temperature Scheduling

Based on the DES scores, we implement implement a hybrid sampling strategy that combines temperature scheduling with threshold-based filtering. For each training epoch, we first compute the temperature-controlled probability distribution:

$$P_t(x) = \frac{e^{\mathrm{DES}(x)/T(t)}}{\sum_{x_i \in \mathcal{B}} e^{\mathrm{DES}(x_i)/T(t)}} \tag{6}$$

where the temperature follows a decay schedule:

$$T(t) = T_0 \cdot (1 - \frac{t}{t_{\max}}) \tag{7}$$

Here, T_0 denotes the initial temperature and $t_{\max}$ represents the maximum number of training epochs. The final training samples are selected based on a probability threshold τ:

$$\mathcal{S} = \{x \in \mathcal{B} : P_t(x) \geq \tau\} \tag{8}$$

where $\mathcal{B}$ denotes the batch size. This hybrid approach enables broad exploration in early stages (high $T(t)$) and gradually shifts towards selecting high-DES samples (through threshold τ), providing flexible control over both sample quality and quantity throughout the training process.

3.4 Reinforcement Learning Algorithm

To train our model efficiently, we adopt the Proximal Policy Optimization (PPO) (Schulman et al., 2017) algorithm. For each problem q, PPO samples $|o|$ responses from the old policy $\pi_{\theta_{\text{old}}}$ and optimizes the trained policy π_θ by maximizing the following objective:

$$\begin{aligned}\mathcal{J}_{\text{PPO}}(\theta) &= \mathbb{E}_{[q\sim P(Q), o\sim\pi_{\theta_{\text{old}}}(O|q)]} \\ &\frac{1}{|o|}\sum_{t=1}^{|o|}\min\Big(\frac{\pi_\theta(o_t|q,o_{<t})}{\pi_{\theta_{\text{old}}}(o_t|q,o_{<t})}A_t, \\ &\text{clip}(\frac{\pi_\theta(o_t|q,o_{<t})}{\pi_{\theta_{\text{old}}}(o_t|q,o_{<t})}, 1-\varepsilon, 1+\varepsilon)A_t\Big)\end{aligned} \tag{9}$$

For the reward function design, we follow a similar approach to DeepSeek-R1 (Guo et al., 2025), implementing a rule-based reward system that evaluates both answer correctness and format compliance. Let a denote the model's output answer for a given mathematical problem, the reward function is defined as:

$$R(a) = \begin{cases} 1 & \text{if } a \text{ is correct,} \\ -0.5 & \text{if } a \text{ is incorrect but well-formatted,} \\ -1 & \text{if } a \text{ has formatting errors.} \end{cases} \tag{10}$$

To ensure consistent evaluation and facilitate automated assessment, we require the model's output to follow a structured format. Specifically, the final answer must be enclosed within 'boxed{}' notation, and the complete reasoning process should be wrapped in appropriate tags. This structured approach not only enables precise reward calculation but also promotes clear and systematic problem-solving strategies.

4 Experiments

4.1 Experimental Setup

We conducted reinforcement learning training using the Proximal Policy Optimization (PPO) (Schulman et al., 2017) algorithm within the OpenRLHF (Hu et al., 2024) framework. Our experiments employed the **Qwen2.5-MATH-7B-Instruct** (Yang et al., 2024) model as the initial base model. During the exploration phase, a rollout batch size of 128 was configured, and 8 samples per prompt were generated with a temperature setting of 1.2. The training process utilized a batch size of 64, with learning rates set to 5×10^{-7} for the actor model and 9×10^{-6} for the critic model. Additionally, a KL coefficient of 0.01 was applied to regulate policy updates. Further experimental details and hyperparameters are provided in Appendix B.

Notably, all experiments are conducted on a single machine equipped with 8×A800 GPUs, aiming to address the requirements of low-resource scenarios. Our approach does not utilize a system-wide prompt. Instead, we append the instruction "Please reason step by step, and put your final answer within boxed{}." at the end of each problem to guide the model's reasoning process.

4.2 Benchmarks

To comprehensively evaluate the performance of the trained models, we conduct experiments on **5** diverse mathematical benchmarks, as detailed in Table 2. These benchmarks include MATH 500 (Hendrycks et al., 2021), a comprehensive collection of 500 problems, AIME (2024, 2025) featuring challenging competition problems, AMC 2023

Table 1: PASS@1 Performance of Baseline Models on Various Mathematical Benchmarks (best and second-best results highlighted)

Model	MATH 500 ↑	AIME 2024 ↑	AMC 2023 ↑	avg. ↑	Labeled Data ↓
Qwen2.5-MATH-7B-Instruct	72.4	16.7	56.4	48.5	/
DeepSeek-R1-Distill-7B@3k	60.1	10.0	26.2	32.1	800K
SimpleRL-Zero-7B	78.2	**26.7**	60.2	55.0	8.9K
PRIME-Zero-7B	83.8	16.7	62.7	54.4	230K
OpenReasoner-Zero-7B@3k	79.2	13.3	47.0	46.5	129K
Qwen2.5-MATH-7B-LIMR	78.0	22.5	63.8	52.8	**1.4K**
Qwen2.5-MATH-7B-MATH8K	81.4	22.5	63.3	55.7	8K
Qwen2.5-MATH-7B-Random	81.4	20.3	58.6	53.4	1.5K
ScalingRL	**84.2**	22.5	**67.5**	**58.1**	1.5K

representing standard high school mathematics, and OlympiadBench (He et al., 2024a) containing complex Olympic-level questions. For AIME2025 and OlympiadBench, we analyze the response evolution trends as shown in Figure 4.

4.3 Baselines

We compare our approach with several state-of-the-art 7B-parameter mathematical reasoning models, to fairly validate whether ScalingRL can achieve effective improvements through self-evolution. Our primary comparison is with Qwen2.5-MATH-7B-Instruct, which undergoes large-scale instruction-following training based on Qwen2.5-MATH-7B architecture. Additionally, we include several leading "R1-Zero-Like" models that employ reinforcement learning with similar backbone architectures: DeepSeek-R1-Distill-7B (Guo et al., 2025), SimpleRL-Zero-7B (Zeng et al., 2025), PRIME-Zero-7B (Cui et al., 2025), OpenReasoner-Zero-7B (Liu et al., 2025b), and LIMR (Li et al., 2025). For controlled comparison, we also implement two variants: Qwen2.5-MATH-7B-MATH8K using the filtered 8K dataset without dynamic sampling, and Qwen2.5-MATH-7B-Random using random sampling from 1.5K examples.

4.4 Evaluation Metrics

We limit the models to generate a maximum of 8,192 tokens and adopt PASS@1 as our primary evaluation metric. Specifically, we set the sampling temperature to 0.6 to produce k responses for each question, where k is typically 16. The PASS@1 metric is subsequently calculated as:

$$\text{PASS@1} = \frac{1}{k}\sum_{i=1}^{k} p_i, \tag{11}$$

where p_i represents the accuracy of the i-th response.

Table 2: Evaluation Benchmarks Overview: Sample Size, Average and Maximum Token Lengths, and Difficulty Levels

Benchmark	#Size	Avg. L	Max. L	Difficulty
MATH 500	500	312	1730	Medium
AIME 2024	30	356	938	Hard
AIME 2025	30	418	986	Hard
AMC 2023	50	286	688	Medium
OlympiadBench	675	486	4420	Hard

4.5 Main Results

We evaluate our ScalingRL framework from performance, data efficiency, and model scalability, with results presented in Table 1.

Performance Analysis On MATH 500, ScalingRL achieves the best performance (84.2%), outperforming the strong baseline PRIME-Zero-7B (83.8%) while using only 1.5K labeled samples compared to PRIME's 230K samples. For AMC 2023, our method reaches 67.5%, establishing a new state-of-the-art result with a significant margin over the second-best performer Qwen2.5-MATH-7B-LIMR (63.8%). In terms of average performance across benchmarks, ScalingRL achieves 58.1%, surpassing all baseline models while maintaining data efficiency.

Data Efficiency Notably, ScalingRL achieves these results with only 1.5K labeled samples, which is substantially less than most baselines (e.g., Qwen2.5-MATH-7B-Instruct uses 3.1M samples, DeepSeek uses 800K samples). While Qwen2.5-MATH-7B-LIMR demonstrates competitive performance with 1.4K samples, our experiments reveal

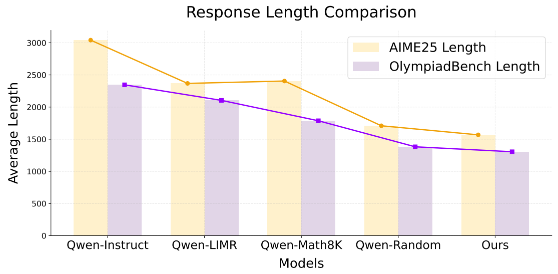


Figure 4: Comparison of response lengths across different models on AIME 2025 and OlympiadBench datasets.

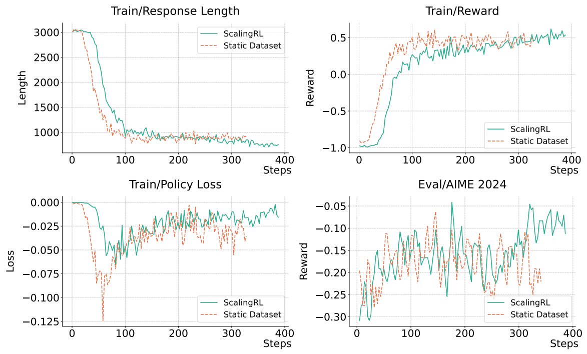


Figure 5: Training dynamics comparison between two different data selection strategies. The green solid line represents ScalingRL with dynamic data selection, while the orange dashed line shows the static dataset approach where training data remains fixed throughout the training process.

that its reported performance on challenging benchmarks like AIME 2024 (claimed >30%) could only reach 22.5% in our reproduction. Additionally, LIMR's data selection strategy requires multiple complete training epochs to obtain reward signals for filtering, significantly increasing computational overhead. In contrast, our method maintains reliable performance across all benchmarks while enabling more efficient training through dynamic data selection.

Training Dynamics and Generalization As illustrated in Figure 5, models trained on static datasets demonstrate faster convergence in both reward metrics and response lengths. However, this rapid convergence may indicate overfitting rather than genuine learning progress. Our analysis reveals that while static training quickly achieves high performance on its fixed dataset, it fails to generalize well to challenging evaluation tasks like AIME. In contrast, ScalingRL's dynamic data selection strategy shows a more gradual learning curve. The strategy resembles a curriculum learning process where the model progressively builds up its mathematical reasoning capabilities. This explains why, despite showing slower training metrics, our method achieves superior performance on complex mathematical problems during evaluation.

Table 3: Ablation study on DES components. Check marks indicate which components are included in each variant configuration, with performance metrics showing the impact of different combinations.

Configuration	Components			PASS@1
	$D(x)$	$C(x)$	$R(x)$	
Single-D	✓	–	–	80.7
Single-C	–	✓	–	79.8
Single-R	–	–	✓	81.6
Dual-DC	✓	✓	–	83.3
Dual-DR	✓	–	✓	82.9
Dual-CR	–	✓	✓	82.4
ScalingRL	✓	✓	✓	**84.2**

4.6 Ablation Study

To evaluate the effectiveness of each component in our Dynamic Effectiveness Score (DES), we conduct comprehensive ablation studies. Table 3 presents the performance comparison of different DES component combinations.

Using single components ($D(x)$, $C(x)$, or $R(x)$ alone) shows limited effectiveness, indicating that each aspect captures different but essential characteristics of training samples. Pairs of components ($D(x) + C(x)$, $D(x) + R(x)$, or $C(x) + R(x)$) demonstrate improved performance, suggesting synergistic effects between different aspects. The full DES combining all three components achieves the best performance, validating our design choice of integrating difficulty, reasoning complexity, and reward adaptability.

5 Conclusion

In this paper, we present an efficient reinforcement learning framework for mathematical reasoning that focuses on maximizing training efficiency through dynamic data selection which contains a three-dimensional filtering process based on quality, diversity, and difficulty. We first refine initial dataset from 220K to 1.5K samples. Then, we introduce a novel Dynamic Effectiveness Score (DES) for intelligent sample evaluation and selection during training. At last, we achieve substantial improvement in training efficiency. We expect our approach will benefit researchers in scenarios where computational resources are limited.

Limitations

While ScalingRL demonstrates significant improvements in data-efficient RL for LLMs, our framework has two limitations that warrant further investigation. First, the initial data curation phase relies on heuristic-based filtering to reduce the prompt pool from 220K to 8K examples. This process, while effective in establishing a robust starting point, lacks full automation and depends on manually defined criteria (e.g., clustering thresholds or diversity metrics). Such heuristic choices may introduce bias or inefficiency, potentially limiting the framework's adaptability to diverse datasets or tasks. Future work could explore integrating model-driven scoring mechanisms (e.g., uncertainty estimation or reward prediction) into this stage to enable fully automated, dynamic data pruning. Second, the current evaluation of **ScalingRL** is confined to mathematical reasoning tasks, such as theorem proving or problem-solving. While this domain provides a structured environment for rigorous testing, it limits the generalizability of the framework to other applications, such as natural language understanding, code generation, or multimodal tasks. Validating **ScalingRL** on broader datasets (e.g., dialogue systems, scientific text generation, or cross-modal reasoning) would strengthen its practical relevance and uncover potential domain-specific challenges.

Ethics Statement

The datasets used in this study are derived from publicly available mathematical reasoning benchmarks (e.g., MATH, AIME, AMC, Olympiadbench), which do not contain personally identifiable information (PII) or sensitive content. All data processing steps (e.g., clustering for initial curation) are performed on anonymized, pre-tokenized prompts, ensuring no direct exposure to private or confidential information. All experimental workflows and outcomes are disclosed in detail to ensure replicability, alongside strict compliance with model usage policies and regulatory frameworks.

Acknowledgements

We would like to express gratitude to the anonymous reviewers for their comments. This work is supported by DAMO Academy through DAMO Academy Research Intern Program. We also wish to thank the individuals who assisted with the data annotation, particularly the 50 professional employees from the State Grid Shandong Electric Power Company, for their valuable time and effort in manually labeling the data.

References

Lichang Chen, Shiyang Li, Jun Yan, Hai Wang, Kalpa Gunaratna, Vikas Yadav, Zheng Tang, Vijay Srinivasan, Tianyi Zhou, Heng Huang, and 1 others. 2023. Alpagasus: Training a better alpaca with fewer data. *arXiv preprint arXiv:2307.08701.*

Qiguang Chen, Libo Qin, Jinhao Liu, Dengyun Peng, Jiannan Guan, Peng Wang, Mengkang Hu, Yuhang Zhou, Te Gao, and Wanxiang Che. 2025a. Towards reasoning era: A survey of long chain-of-thought for reasoning large language models. *arXiv preprint arXiv:2503.09567.*

Xiaoyin Chen, Jiarui Lu, Minsu Kim, Dinghuai Zhang, Jian Tang, Alexandre Piché, Nicolas Gontier, Yoshua Bengio, and Ehsan Kamalloo. 2025b. Self-evolving curriculum for llm reasoning. *arXiv preprint arXiv:2505.14970.*

Xiangxiang Chu, Hailang Huang, Xiao Zhang, Fei Wei, and Yong Wang. 2025. Gpg: A simple and strong reinforcement learning baseline for model reasoning. *arXiv preprint arXiv:2504.02546.*

Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Wendi Li, Bingxiang He, Yuchen Fan, Tianyu Yu, Qixin Xu, Weize Chen, and 1 others. 2025. Process reinforcement through implicit rewards. *arXiv preprint arXiv:2502.01456.*

Nirjhar Das, Souradip Chakraborty, Aldo Pacchiano, and Sayak Ray Chowdhury. 2024. Active preference optimization for sample efficient rlhf. *arXiv preprint arXiv:2402.10500.*

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948.*

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, and 1 others. 2024a. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. *arXiv preprint arXiv:2402.14008.*

Yuhang He, Jihai Zhang, Jianzhu Bao, Fangquan Lin, Cheng Yang, Bing Qin, Ruifeng Xu, and Wotao Yin. 2024b. Bc-prover: Backward chaining prover for formal theorem proving. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 3059–3077.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*.

Jian Hu. 2025. Reinforce++: A simple and efficient approach for aligning large language models. *arXiv preprint arXiv:2501.03262*.

Jian Hu, Xibin Wu, Zilin Zhu, Weixun Wang, Dehao Zhang, Yu Cao, and 1 others. 2024. Openrlhf: An easy-to-use, scalable and high-performance rlhf framework. *arXiv preprint arXiv:2405.11143*.

Hamish Ivison, Noah A Smith, Hannaneh Hajishirzi, and Pradeep Dasigi. 2022. Data-efficient finetuning using cross-task nearest neighbors. *arXiv preprint arXiv:2212.00196*.

Hamish Ivison, Muru Zhang, Faeze Brahman, Pang Wei Koh, and Pradeep Dasigi. 2025. Large-scale data selection for instruction tuning. *arXiv preprint arXiv:2503.01807*.

Leslie Pack Kaelbling, Michael L Littman, and Andrew W Moore. 1996. Reinforcement learning: A survey. *Journal of artificial intelligence research*, 4:237–285.

Amirhossein Kazemnejad, Milad Aghajohari, Eva Portelance, Alessandro Sordoni, Siva Reddy, Aaron Courville, and Nicolas Le Roux. 2024. Vineppo: Unlocking rl potential for llm reasoning through refined credit assignment. *arXiv preprint arXiv:2410.01679*.

Komal Kumar, Tajamul Ashraf, Omkar Thawakar, Rao Muhammad Anwer, Hisham Cholakkal, Mubarak Shah, Ming-Hsuan Yang, Phillip HS Torr, Fahad Shahbaz Khan, and Salman Khan. 2025. Llm post-training: A deep dive into reasoning large language models. *arXiv preprint arXiv:2502.21321*.

Xuefeng Li, Haoyang Zou, and Pengfei Liu. 2025. Limr: Less is more for rl scaling. *arXiv preprint arXiv:2502.11886*.

Ben Liu, Jihai Zhang, Fangquan Lin, Cheng Yang, and Min Peng. 2024a. Filter-then-generate: Large language models with structure-text adapter for knowledge graph completion. *arXiv preprint arXiv:2412.09094*.

Ben Liu, Jihai Zhang, Fangquan Lin, Cheng Yang, Min Peng, and Wotao Yin. 2025a. Symagent: A neural-symbolic self-learning agent framework for complex reasoning over knowledge graphs. In *Proceedings of the ACM on Web Conference 2025*, pages 98–108.

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

Zijun Liu, Boqun Kou, Peng Li, Ming Yan, Ji Zhang, Fei Huang, and Yang Liu. 2024b. Enabling weak llms to judge response reliability via meta ranking. *arXiv preprint arXiv:2402.12146*.

Zijun Liu, Peiyi Wang, Runxin Xu, Shirong Ma, Chong Ruan, Peng Li, Yang Liu, and Yu Wu. 2025c. Inference-time scaling for generalist reward modeling. *arXiv preprint arXiv:2504.02495*.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. 2025. s1: Simple test-time scaling. *Preprint*, arXiv:2501.19393.

William Muldrew, Peter Hayes, Mingtian Zhang, and David Barber. 2024. Active preference learning for large language models. *arXiv preprint arXiv:2402.08114*.

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

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

Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, and 1 others. 2025. Kimi k1. 5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599*.

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

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. 2024. Less: Selecting influential data for targeted instruction tuning. *arXiv preprint arXiv:2402.04333*.

Wei Xiong, Jiarui Yao, Yuhui Xu, Bo Pang, Lei Wang, Doyen Sahoo, Junnan Li, Nan Jiang, Tong Zhang, Caiming Xiong, and 1 others. 2025. A minimalist approach to llm reasoning: from rejection sampling to reinforce. *arXiv preprint arXiv:2504.11343*.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, and 1 others. 2024. Qwen2. 5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*.

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

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

Yufeng Yuan, Qiying Yu, Xiaochen Zuo, Ruofei Zhu, Wenyuan Xu, Jiaze Chen, Chengyi Wang, TianTian Fan, Zhengyin Du, Xiangpeng Wei, and 1 others. 2025a. Vapo: Efficient and reliable reinforcement learning for advanced reasoning tasks. *arXiv preprint arXiv:2504.05118*.

Yufeng Yuan, Yu Yue, Ruofei Zhu, Tiantian Fan, and Lin Yan. 2025b. What's behind ppo's collapse in long-cot? value optimization holds the secret. *arXiv preprint arXiv:2503.01491*.

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

Jihai Zhang, Wei Wang, Siyan Guo, Li Wang, Fangquan Lin, Cheng Yang, and Wotao Yin. 2024. Solving general natural-language-description optimization problems with large language models. *arXiv preprint arXiv:2407.07924*.

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

Kaichen Zhang, Yuzhong Hong, Junwei Bao, Hongfei Jiang, Yang Song, Dingqian Hong, and Hui Xiong. 2025a. Gvpo: Group variance policy optimization for large language model post-training. *arXiv preprint arXiv:2504.19599*.

Xiaojiang Zhang, Jinghui Wang, Zifei Cheng, Wenhao Zhuang, Zheng Lin, Minglei Zhang, Shaojie Wang, Yinghan Cui, Chao Wang, Junyi Peng, and 1 others. 2025b. Srpo: A cross-domain implementation of large-scale reinforcement learning on llm. *arXiv preprint arXiv:2504.14286*.

A Difficulty Level Classification

The difficulty levels are classified based on DeepSeek-R1's performance over n sampling iterations:

- Level 1: 100% correct answers
- Level 2: Accuracy rate $\in (50\%, 100\%)$
- Level 3: Accuracy rate $\in (0\%, 50\%]$
- Level 4: 0% accuracy but all answers in correct format
- Level 5: 0% accuracy with format errors

B More Experimental Setup

Table 4: Detailed experimental configuration for ScalingRL training.

Model Configuration	
Base Model	Qwen2.5-Math-7B-Instruct
Training Strategy	PPO with Rule-based Reward Model
Training Device	8×H100 GPUs with VLLM
Precision	BF16 mixed precision
PPO Hyperparameters	
Learning Rate	5e-7 (actor), 9e-6 (critic)
Batch Size	64 (train), 128 (rollout)
Sequence Length	1024 (prompt), 3072 (generation)
Episodes	20
KL Coefficient	0.01
Epsilon/Value Clip	0.2/0.2
GAE Lambda	0.95
Discount Factor	1.0
Temperature	1.2 (train), 0.7 (eval)
Training Infrastructure	
Distributed Setup	Ray framework
Model Parallelism	VLLM with 4 engines
Tensor Parallel Size	2
GPU Memory Usage	50%
DeepSpeed Zero Stage	3
Optimization Details	
Gradient Checkpointing	Enabled
Flash Attention	Enabled
Optimizer	Adam with Offloading
KL Estimator	K3
Advantage Estimation	GAE

C Additional Experimental Results and Clarifications

This section consolidates the additional experiments, clarifications, and implementation details that were provided during the rebuttal process in response to reviewers' comments. We include new baseline comparisons, hyperparameter sensitivity studies, reproducibility details, cost analysis, and dataset filtering pipelines.

C.1 Comparison with Data Selection and Curriculum Learning Methods

To contextualize our contributions, we evaluated ScalingRL against state-of-the-art data selection methods (**s1** (Muennighoff et al., 2025), **LIMO** (Ye et al., 2025), and LIMR (Li et al., 2025)) and the recent curriculum learning method **SEC** (Chen et al., 2025b). Results for Qwen2.5-MATH-7B are shown in Table 5 and Table 6.

Table 5: Comparison with data selection methods (Qwen2.5-MATH-7B). Labeled Data = math reasoning samples used in RL stage.

Method	MATH500	AIME24	AMC23	Avg.
Random	81.4	20.3	58.6	53.4
LIMR	78.0	22.5	63.8	52.8
s1	55.8	15.8	42.5	38.0
LIMO	65.0	15.8	56.3	45.7
ScalingRL	**84.2**	**22.5**	**67.5**	**58.1**

Table 6: Comparison with curriculum learning method SEC.

Method	MATH500	AIME24	AMC22-23
SEC-3B	67.2	10.0	35.1
SEC-7B	76.1	17.5	51.1
ScalingRL-7B	**84.2**	**22.5**	**65.4**

C.2 Hyperparameter Sensitivity Analysis

We performed sensitivity analysis for the DES weighting coefficients $(\lambda_d, \lambda_c, \lambda_r)$ and the temperature threshold τ. Results are provided in Table 7 and Table 8.

Table 7: Performance vs. selection threshold τ.

τ	Labeled Data	MATH500	AMC23
0.50	4382	**84.2**	66.8
0.55	2459	83.4	67.2
0.60	**1537**	**84.2**	**67.5**
0.70	815	81.2	63.4

C.3 Sampling Strategy Ablation

We examined the effect of different sampling strategies: random selection, fixed Gaussian, and our dynamic adaptive Gaussian. See Table 9.

Table 8: Performance vs. DES weights.

λ_d	λ_c	λ_r	MATH500	AMC23
0.4	0.3	0.3	**84.2**	67.5
0.4	0.4	0.2	84.0	**67.9**

Table 9: Sampling strategy comparison.

Strategy	MATH500	AIME24
Random	72.5	16.3
Fixed Gaussian	78.9	18.8
Dynamic Gaussian	**84.2**	**22.5**

C.4 Cost of Initial Candidate Construction

The "Initial Candidate Construction" stage reduced 220K raw problems to 8K curated candidates through automated filtering:

- Structural/mathematical syntax checks: negligible wall time.
- Clarity check via LLM judge: ~4h.
- Easy problem removal (Qwen-1.5B inference): ~2h.
- Hard problem reasoning (Qwen-7B/32B): ~3h.
- Validation by Claude 3.5 Sonnet API: ~4h.

Total one-time cost: $\approx \mathbf{14-15}$ hours.

C.5 Automated Filtering Implementation

The filtering pipeline (fully automated) is as follows:

1. **Quality**: structural field check (Python), math expression parsing (Sympy), LLM-based clarity checking.
2. **Diversity**: stratified sampling over `problem_type`.
3. **Difficulty**: removal of trivial problems solved by Qwen-1.5B, validation of upper-bound difficulty with Qwen-7B/32B + Claude API, enforce 4K token solution limit.

C.6 ScalingRL on Different Models and Datasets

We verified ScalingRL's generality on:

- **DeepSeek-R1-Distill-1.5B**: consistent improvements over static selection.
- **DeepScaleR-Preview dataset** (4K samples): ScalingRL outperforms RL with static data.