Reinforcement Learning for Reasoning in Large Language Models with *One* Training Example

Yiping Wang[1] † * **Qing Yang**[2] **Zhiyuan Zeng**[1] **Liliang Ren**[3] **Lucas Liu**[3]

Baolin Peng[3] **Hao Cheng**[3] **Xuehai He**[4] **Kuan Wang**[5] **Jianfeng Gao**[3]

Weizhu Chen[3] **Shuohang Wang**[3] † **Simon Shaolei Du**[1] † **Yelong Shen**[3] †

[1]University of Washington [2]University of Southern California [3]Microsoft
[4]University of California, Santa Cruz [5]Georgia Institute of Technology

Abstract

We show that reinforcement learning with verifiable reward using one training example (*1-shot RLVR*) is effective in incentivizing the mathematical reasoning capabilities of large language models (LLMs). Applying RLVR to the base model Qwen2.5-Math-1.5B, we identify a single example that elevates model performance on MATH500 from 36.0% to 73.6%, and improves the average performance across six common mathematical reasoning benchmarks from 17.6% to 35.7%. This result matches the performance obtained using the 1.2k DeepScaleR subset (MATH500: 73.6%, average: 35.9%), which includes the aforementioned example. Furthermore, RLVR with only two examples even slightly exceeds these results (MATH500: 74.8%, average: 36.6%). Similar substantial improvements are observed across various models (Qwen2.5-Math-7B, Llama3.2-3B-Instruct, DeepSeek-R1-Distill-Qwen-1.5B), RL algorithms (GRPO and PPO), and different math examples (many of which yield approximately 30% or greater improvement on MATH500 when employed as a single training example). In addition, we identify some interesting phenomena during 1-shot RLVR, including cross-domain generalization, increased frequency of self-reflection, and sustained test performance improvement even after the training accuracy has saturated, a phenomenon we term *post-saturation generalization*. Moreover, we verify that the effectiveness of 1-shot RLVR primarily arises from the policy gradient loss, distinguishing it from the "grokking" phenomenon. We also show the critical role of promoting exploration (e.g., by incorporating entropy loss with an appropriate coefficient) in 1-shot RLVR training. As a bonus, we observe that applying entropy loss alone, without any outcome reward, significantly enhances Qwen2.5-Math-1.5B's performance on MATH500 by 27.4%. These findings can inspire future work on RLVR data efficiency and encourage a re-examination of both recent progress and the underlying mechanisms in RLVR. Our code, model, and data are open source at https://github.com/ypwang61/One-Shot-RLVR.

1 Introduction

Recently, significant progress has been achieved in enhancing the reasoning capabilities of large language models (LLMs), including OpenAI-o1 [1], DeepSeek-R1 [2], and Kimi-1.5 [3], particularly

[*] : This work was done during Yiping's internship at Microsoft. †: Corresponding authors. Correspondence email: {ypwang61, ssdu}@cs.washington.edu, {shuohang, yeshe}@microsoft.com

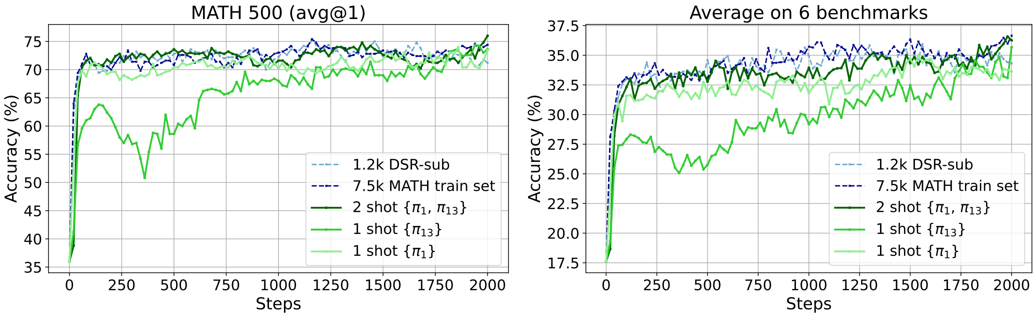


Figure 1: **RLVR with 1 example (green) can perform as well as using datasets with thousands of examples (blue).** Left/Right corresponds to MATH500/Average performance on 6 mathematical reasoning benchmarks. Base model is Qwen2.5-Math-1.5B. π_1 and π_{13} are examples defined by Eqn. 2 and detailed in Tab. 2, and they are all from the 1.2k DeepScalerR subset (DSR-sub). Setup details are in Sec. 3.1. We find that RLVR with 1 example $\{\pi_{13}\}$ (35.7%) performs close to that with 1.2k DSR-sub (35.9%), and RLVR with 2 examples $\{\pi_1, \pi_{13}\}$ (36.6%) even performs better than RLVR with DSR-sub and as well as using 7.5k MATH train dataset (36.7%). Detailed results are in Fig. 2. Additional results for non-mathematical reasoning tasks are in Tab. 1.

for complex mathematical tasks. A key method contributing to these advancements is *Reinforcement Learning with Verifiable Reward* (RLVR) [4, 5, 2, 3], which commonly employs reinforcement learning on an LLM with a rule-based outcome reward, such as a binary reward indicating the correctness of the model's final answer to a math problem. Several intriguing empirical phenomena have been observed in RLVR, such as the stimulation or enhancement of specific cognitive behaviors [6] (e.g., self-reflection) and improved generalization across various downstream tasks [5, 2, 3].

Currently, substantial efforts are directed toward refining RL algorithms (e.g., PPO [7] and GRPO [8]) to further enhance RLVR's performance and stability [9–16]. Conversely, data-centric aspects of RLVR remain relatively underexplored. Although several studies attempt to curate high-quality mathematical reasoning datasets [17, 18, 11], there is relatively limited exploration into the specific role of data in RLVR. Thus, critical questions remain open: How much data is truly necessary? What data is most effective? How do the quality and quantity of the training data relate to observed empirical phenomena (e.g., self-reflection and robust generalization)? The most relevant study to these problems is LIMR [19], which proposed a metric called *learning impact measurement* (LIM) to evaluate the effectiveness of training examples. Using the LIM score, they maintain model performance while reducing the number of training examples by sixfold. However, this study does not explore how aggressively the RLVR training dataset can be reduced. Motivated by these considerations, in this paper, we specifically investigate the following research question:

"To what extent can we reduce the training dataset for RLVR while maintaining comparable performance compared to using the full dataset?"

We empirically demonstrate that, surprisingly, **the training dataset for RLVR can be reduced to as little as *ONE* example!** This finding supports recent claims that base models already possess significant reasoning capabilities [13, 20, 6, 21], and further shows that a single example is sufficient to substantially enhance the base model's mathematical performance. We refer to this setup as *1-shot RLVR*. We summarize our contributions and findings below:

- We find that selecting one specific example as the training dataset can achieve similar downstream performance to that of the 1.2k DeepScaleR subset (DSR-sub) containing that example. Specifically, this improves the Qwen2.5-Math-1.5B model from 36.0% to 73.6% on MATH500, and from 17.6% to 35.7% on average across 6 mathematical reasoning benchmarks (Fig. 1, 2). Additionally, 2-shot RLVR slightly surpasses DSR-sub and matches performance with the 7.5k MATH training set. Notably, these two examples are relatively easy for the base model, which can solve them with high probability without any training. Additionally, 1-shot RLVR on math examples can improve model performance on non-mathematical reasoning tasks, even outperforming full-set RLVR (Tab. 1).

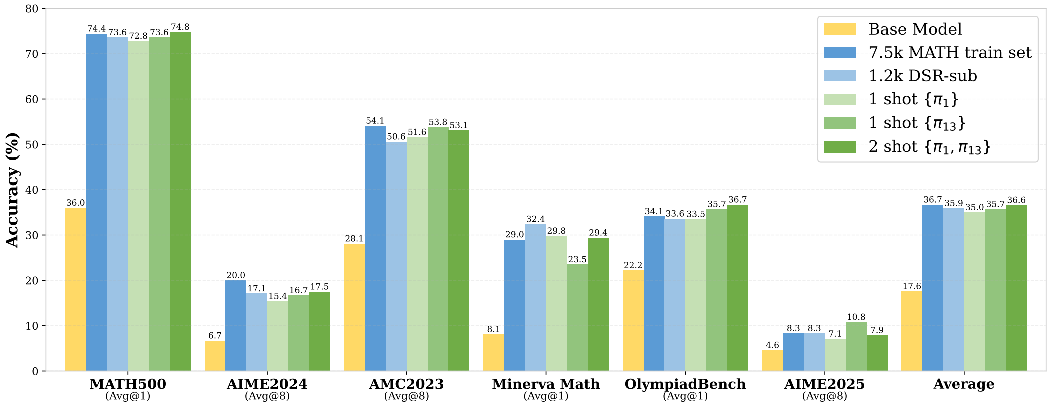


Figure 2: **Detailed performance of 1/2-shot RLVR for Qwen2.5-Math-1.5B.** Results are reported for the checkpoint achieving the best average across 6 math benchmarks (Fig. 1). Yellow, blue, and green correspond to the performance of the base model, RLVR with full dataset, and 1/2-shot RLVR, respectively. All RLVR setups significantly outperform the base model. Further evaluation details are provided in Sec. 3.1. Models' best individual benchmark results are listed in Tab. 8 in the Appendix. Additional results for non-mathematical reasoning tasks are in Tab. 1.

- We confirm the effectiveness of 1(few)-shot RLVR across different base models (Qwen2.5-Math-1.5/7B, Llama3.2-3B-Instruct), models distilled from long Chain-of-Thought (CoT) data (DeepSeek-R1-Distill-Qwen-1.5B), and different RL algorithms (GRPO, PPO).
- We highlight an intriguing phenomenon in 1-shot RLVR: **post-saturation generalization**. Specifically, the training accuracy on the single example rapidly approaches 100%, yet the model's test accuracy continues to improve. Moreover, despite using only one training example, overfitting does not occur until after approximately 1.4k training steps. Even post-overfitting, while the model's reasoning outputs for the training example become incomprehensible multilingual gibberish mixed with correct solutions, its test performance remains strong, and the reasoning outputs for the test examples remain human-interpretable.
- In addition, we demonstrate the following phenomena: (1) 1-shot RLVR is viable for nearly all math examples in the full dataset when each example is individually used for training. (2) 1-shot RLVR enables cross-domain generalization: training on a single example from one domain (e.g., Geometry) often enhances performance in other domains (e.g., Algebra, Number Theory). (3) As 1-shot RLVR training progresses, both the response length for the training example and the frequency of self-reflective terms in downstream tasks increase.
- Through ablation studies, we show that policy gradient loss primarily drives the improvements observed in 1-shot RLVR, distinguishing it from "grokking", which heavily depends on regularization methods like weight decay. Additionally, we emphasize the importance of promoting diverse exploration in model outputs, showing that adding an entropy loss with an appropriate coefficient further enhances performance.
- Lastly, we find that employing entropy loss alone, even without any outcome reward, achieves a 27% performance boost on MATH500 for Qwen2.5-Math-1.5B. Similar improvements are observed for Qwen2.5-Math-7B and Llama-3.2-3B-Instruct. We also discuss the relation of this observation to label robustness.

2 Preliminary

RL Loss Function. In this paper, we adopt GRPO [8, 2] as the reinforcement learning algorithm for large language models unless stated otherwise. The GRPO loss function comprises three main components: *policy gradient loss, KL divergence loss, and entropy loss*. We briefly describe each component here, with further details provided in Appendix A.1.

(1) The policy gradient loss encourages the model to produce response sequences with higher rewards, assigning weights to sampled outputs according to their group-normalized advantages. Thus,

better-than-average solutions are reinforced, whereas inferior ones are penalized. Since we focus on mathematical problems, the reward is defined as binary (0-1), where a reward of 1 is granted only when the outcome of the model's response correctly matches the ground-truth answer.

(2) The KL divergence loss measures the divergence between the current model's responses and those from a reference model, serving as a regularization term to maintain general language quality.

(3) The entropy loss, applied with a negative coefficient, incentivizes higher per-token entropy to encourage exploration and generate more diverse reasoning paths. We note that entropy loss is not strictly necessary for GRPO training, but it is included by default in the verl [22] pipeline used in our experiments. Its effect on 1-shot RLVR is further discussed in Sec. 4.2.

Data Selection: Historical Variance Score. To explore how extensively we can reduce the RLVR training dataset, we propose a simple data selection approach for ranking training examples. We first train the model for E epochs on the full dataset using RLVR. Then for each example $i \in [N] = \{1, \ldots, N\}$, we can obtain a list of historical training accuracy $L_i = [s_{i,1}, \ldots, s_{i,E}]$, which records its average training accuracy for every epoch. Note that some previous work has shown that the variance of the reward signal [23] is critical for RL training, we simply rank the data by their historical variance of training accuracy, which is directly related to the reward:

$$v_i := \mathrm{var}(s_{i,1}, \ldots, s_{i,E}) \tag{1}$$

Next, we define a permutation $\pi : [N] \to [N]$ such that $v_{\pi(1)} \geq \cdots \geq v_{\pi(N)}$. Under this ordering, $\pi(j)$ (denoted as π_j for convenience) corresponds to the example with the j-th largest variance v_i:

$$\pi_j := \pi(j) = \underset{j}{\arg\,\mathrm{sort}}\{v_i : i \in [N]\} \tag{2}$$

We then select examples according to this straightforward ranking criterion. For instance, π_1, identified by the historical variance score on Qwen2.5-Math-1.5B, performs well in 1-shot RLVR (Sec. 3.2.3, 3.3). We also choose additional examples from diverse domains among $\{\pi_1, \ldots, \pi_{17}\}$ and evaluate them under 1-shot RLVR (Tab. 3), finding that π_{13} likewise achieves strong performance. Importantly, we emphasize that **this criterion is not necessarily optimal for selecting single examples for 1-shot RLVR**[2]. In fact, Tab. 3 indicates that many examples, including those with moderate or low historical variance, can also individually yield approximately 30% or greater improvement on MATH500 when used as the single training example in RLVR. This suggests a potentially general phenomenon independent of the specific data selection method.

3 Experiments

3.1 Setup

Models. We by default run our experiments on Qwen2.5-Math-1.5B [24, 25], and also verify the effectiveness of Qwen2.5-Math-7B [25], Llama-3.2-3B-Instruct [26], and DeepSeek-R1-Distill-Qwen-1.5B [2] for 1-shot RLVR in Sec. 3.3.

Dataset. Due to resource limitations, we randomly select a subset consisting of 1209 examples from DeepScaleR-Preview-Dataset [18] as our instance pool for data selection (Sec. 2), and we abbreviate it as "DSR-sub" for convenience. Therefore, after ranking the data based on the historical variance score (Eqn. 2), we would denote the examples in DSR-sub as $\pi_1, \ldots, \pi_{1209}$. We also use the MATH [27] training set (consisting of 7500 instances) as another dataset used in full RLVR to provide a comparison. For data selection, as in Sec. 2, we first train Qwen2.5-Math-1.5B for 500 steps, and then obtain its historical variance score (Eqn. 1) and the corresponding ranking (Eqn. 2). To avoid ambiguity, we do not change the correspondence between $\{\pi_i\}$ and examples for all the experiments in this paper, i.e., they are always ranked by the historical variance score of the Qwen2.5-Math-1.5B model. Furthermore, to enable RLVR with one or very few examples, we duplicate the chosen data until they reach the training batch size (e.g., 128) and store them as a new dataset.

[2] Nevertheless, as shown in Tab. 4 (Sec. 3.3), selection based on historical variance scores outperforms random selection in RLVR on Qwen2.5-Math-7B.

Training. As described in Sec. 2, we follow the verl [22] pipeline, and by default, the coefficients for KL divergence and entropy loss are $\beta = 0.001$ and $\alpha = -0.001$, respectively. The training rollout temperature is set to 0.6 for vLLM [28]. The coefficient of weight decay is set to 0.01 by default. The training batch size and mini-batch size are 128, and we sample 8 responses for each prompt. Therefore, we have 8 gradient updates for each rollout step. The learning rate is set to 1e-6. By default, the maximum prompt length is set to 1024, and the maximum response length is 3072, considering that the Qwen2.5-Math-1.5B/7B model has a 4096 context length. But for DeepSeek-R1-Distill-Qwen-1.5B, we let the maximum response length be 8192, following the setup of stage 1 in DeepScaleR [18]. We store the model checkpoint every 20 steps for evaluation, and use 8 A100 GPUs for each experiment. For Qwen2.5-Math-1.5B, Qwen2.5-Math-7B, Llama-3.2-3B-Instruct, and DeepSeek-R1-Distill-Qwen-1.5B, we train for 2000, 1000, 1000, and 1200 steps, respectively, unless the model has already shown a significant drop in performance. We use the same approach as DeepScaleR [18] (whose repository is also derived from the verl) to save the model in safetensor format to facilitate evaluation.

Evaluation. We use the official Qwen2.5-Math evaluation pipeline [25] for our evaluation. Six widely used complex mathematical reasoning benchmarks are used in our paper: MATH500 [27, 29], AIME 2024 [30], AMC 2023 [31], Minerva Math [32], OlympiadBench [33], and AIME 2025 [30]. More details about benchmarks are illustrated in Appendix A.3. We also consider non-mathematical reasoning tasks ARC-Easy and ARC-Challenge [34]. The maximum number of generated tokens is set to be 3072. For Qwen-based models, we use the "`qwen25-math-cot`" prompt template in evaluation. For Llama and distilled models, we use their original chat templates. We set the evaluation seed to be 0 and `top_p` to be 1. For MATH500, we let temperature be 0. For AIME 2024, AIME 2025, and AMC 2023, which contain only 30 or 40 questions, we repeat the test set 8 times for evaluation stability and evaluate the model with temperature = 0.6, and finally report the average `pass@1` (`avg@8`) performance. We use 4 A100 GPUs for the evaluation. We use the same evaluation setting for Llama-3.2-3B-Instruct [26] except using its own chat template, and for DeepSeek-R1-Distill-Qwen-1.5B [2], we also use its own chat template, and set the maximum number of generated tokens to be 8192. By default, we report the performance of the checkpoint that obtains the best average performance on 6 benchmarks. But in Sec. 3.2.3 and Sec. 4.2, since we only evaluate MATH500 and AIME2024, we report the best model performance on each benchmark separately, i.e., the best MATH500 checkpoint and best AIME2024 checkpoint can be different. Finally we mention that there are slightly performance difference on initial model caused by numerical precision, but it does not influence our conclusions (Appendix A.4).

Table 1: **1-shot RLVR with math examples π_1/π_{13} improves model performance on ARC, even better than full-set RLVR.** Base model is Qwen2.5-Math-1.5B, and we evaluate on ARC-Easy (ARC-E) and ARC-Challenge (ARC-C). We select the checkpoint achieving the best average across 6 math benchmarks as shown in Fig. 2.

Dataset	**Size**	**ARC-E**	**ARC-C**
Base	NA	48.0	30.2
MATH	7500	51.6	32.8
DSR-sub	1209	42.2	29.9
$\{\pi_1\}$	1	52.0	32.2
$\{\pi_{13}\}$	1	**55.8**	**33.4**
$\{\pi_1, \pi_{13}\}$	2	52.1	32.4

3.2 Observation of 1/Few-Shot RLVR

In Fig. 1 and Fig. 2, we have found that RLVR with 1 or 2 examples can perform as well as RLVR with datasets containing thousands of examples, and Tab. 1 further shows that 1(few)-shot RLVR with these math examples enable better generalization on non-mathematical reasoning tasks (More details are in Appendix B.1). To better understand this phenomenon, we provide a detailed analysis of 1-shot RLVR in this section.

3.2.1 Dissection of π_1 and π_{13}: Not-So-Difficult Problems

First, we inspect the examples that produce such strong results. Tab. 2 lists the instances of π_1 and π_{13}, which are defined by Eqn. 2. Taking π_1 as an example, we can see that it's actually a simple algebra problem with a physics background. The key steps for it are obtaining $k = 1/256$ for formula $P = kAV^3$, and calculating $V = (2048)^{1/3}$. Interestingly, we have the following

Table 2: **Example π_1 and π_{13}.** They are from a 1.2k subset of DeepScaleR-Preview-Dataset [18] (DSR-sub). A more precise answer for π_1 should be "12.7".

Prompt of example π_1:
`The pressure \\(P \\) exerted by wind on a sail varies jointly as the area \\(A \\) of the sail and the cube of the wind's velocity \\(V \\). When the velocity is \\(8 \\) miles per hour, the pressure on a sail of \\(2 \\) square feet is \\(4 \\) pounds. Find the wind velocity when the pressure on \\(4 \\) square feet of sail is \\(32 \\) pounds. Let's think step by step and output the final answer within \\boxed{}.`
Ground truth (label in DSR-sub): 12.8.
Prompt of example π_{13}:
`Given that circle C passes through points $P(0,-4)$, $Q(2,0)$, and $R(3,-1)$. \n$(1)$ Find the equation of circle C. \n$(2)$ If the line $l: mx+y-1=0$ intersects circle C at points A and B, and $\|AB\|=4$, find the value of m. Let's think step by step and output the final answer within \\boxed{}.`
Ground truth (label in DSR-sub): $\frac{4}{3}$.

observations: (1) **The answer is not precise.** A more precise answer would be 12.7 rather than 12.8 ($\sqrt[3]{2048} \approx 12.699$). (2) **Base model already almost solves π_1.** In Fig. 4, the base model without any training already solves all the key steps before calculating $(2048)^{1/3}$ with high probability. Just for the last step to calculate the cube root, the model has some diverse outputs, including 4, 10.95, 12.6992, $8\sqrt[3]{4}$, 12.70, 12.8, 13, etc. Specifically, for 128 samplings from the base model, we find that 57.8% of outputs are "12.7" or "12.70", 6.3% of outputs are "12.8", and 6.3% are "13".

For π_{13}, it is a geometry problem and its answer is precise. And similarly, the initial base model still has 21.9% of outputs successfully obtaining $\frac{4}{3}$ in 128 samplings. The details of other examples used in this paper are shown in Appendix C.

3.2.2 Post-saturation Generalization: Continual Generalization beyond Training Accuracy Saturation

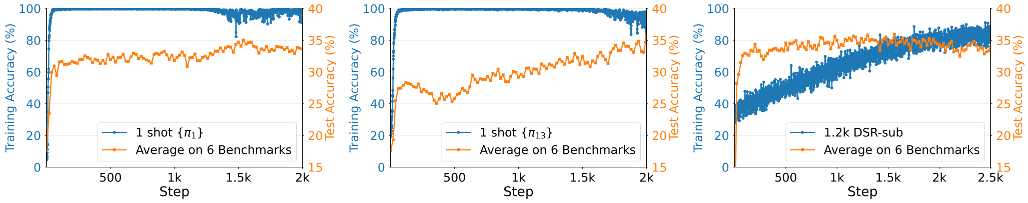
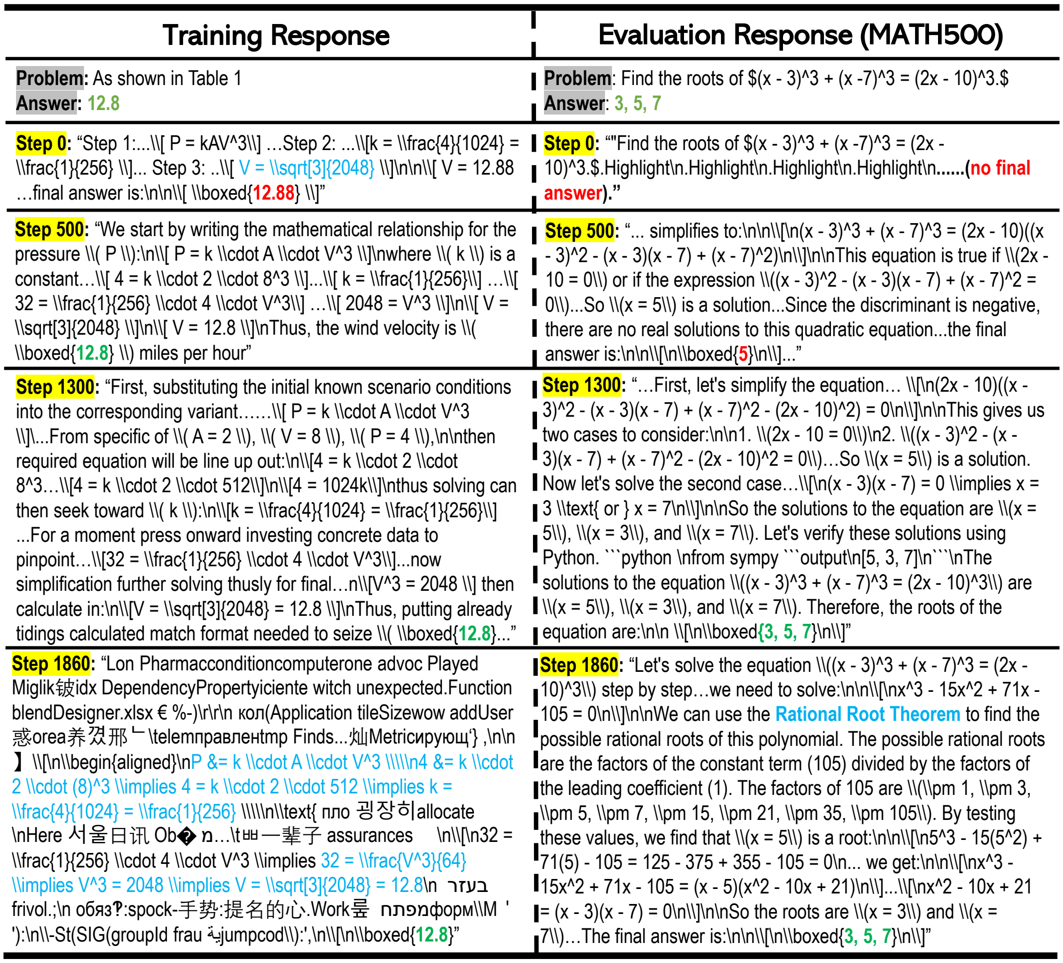


Figure 3: **Post-saturation generalization in 1-shot RLVR.** The training accuracy of π_1(Left) and π_{13}(Middle) in 1-shot RLVR saturates before step 100, but their test performance continues improving. On the other hand, the training accuracy for RLVR with 1.2k DSR-sub dataset (Right) still has not saturated after 2000 steps, but there is no significant improvement on test tasks after step 1000.

Then, we show an interesting phenomenon in the 1-shot RLVR process. As shown in Fig. 3, since we only have one training example, it's foreseeable that the training accuracy for π_1 and π_{13} quickly saturates before the 100th step and remains around 100% throughout the training process. However, the performance on the test set still continue improving: 1-shot RLVR with π_1 gets 3.4% average improvement from step 100 to step 1540, and that with π_{13} dramatically obtains 9.9% average improvement from step 500 to step 2000. We call this phenomenon "**post-saturation generalization**". In particular, this phenomenon cannot be observed when using multi-example datasets like DSR-sub currently, as the test performance has started to drop before training accuracy converges.

Moreover, we compare the training and evaluation responses in Fig. 4. Surprisingly, we find that at the final stage of 1-shot RLVR, the model overfits the single training example by mixing the correct calculation process into long unintelligible multilingual outputs in its outputted reasoning. However, the test responses still behave normally and achieve high accuracy, which implies that **post-saturation generalization still holds even after overfitting the training example**. In particular, we emphasize that although RLVR overfits the single training example, this happens really late, like π_1 is overfitted after 1400 steps and π_{13} after 1800 steps. Considering that each example is sampled 1024 times per



Training Response	Evaluation Response (MATH500)
Problem: As shown in Table 1 **Answer**: 12.8	**Problem**: Find the roots of $(x - 3)^3 + (x -7)^3 = (2x - 10)^3.$ **Answer**: 3, 5, 7
Step 0: "Step 1:...\\[P = kAV^3\\] …Step 2: ...\\[k = \\frac{4}{1024} = \\frac{1}{256} \\]... Step 3: ..\\[V = \\sqrt[3]{2048} \\]\n\n\\[V = 12.88 …final answer is:\n\n\\[\\boxed{12.88} \\]"	**Step 0**: ""Find the roots of $(x - 3)^3 + (x -7)^3 = (2x - 10)^3.$.Highlight\n.Highlight\n.Highlight\n.Highlight\n**......(no final answer).**"
Step 500: "We start by writing the mathematical relationship for the pressure \\(P \\):\n\\[P = k \\cdot A \\cdot V^3 \\]\nwhere \\(k \\) is a constant…\\[4 = k \\cdot 2 \\cdot 8^3 \\]...\\[k = \\frac{1}{256}\\] …\\[32 = \\frac{1}{256} \\cdot 4 \\cdot V^3\\] …\\[2048 = V^3 \\]\n\\[V = \\sqrt[3]{2048} \\]\n\\[V = 12.8 \\]\nThus, the wind velocity is \\(\\boxed{12.8} \\) miles per hour"	**Step 500**: "... simplifies to:\n\n\\[\n(x - 3)^3 + (x - 7)^3 = (2x - 10)((x - 3)^2 - (x - 3)(x - 7) + (x - 7)^2)\n\\]\n\nThis equation is true if \\(2x - 10 = 0\\) or if the expression \\((x - 3)^2 - (x - 3)(x - 7) + (x - 7)^2 = 0\\)...So \\(x = 5\\) is a solution...Since the discriminant is negative, there are no real solutions to this quadratic equation...the final answer is:\n\n\\[\n\\boxed{5}\n\\]..."
Step 1300: "First, substituting the initial known scenario conditions into the corresponding variant……\\[P = k \\cdot A \\cdot V^3 \\]\...From specific of \\(A = 2 \\), \\(V = 8 \\), \\(P = 4 \\),\n\nthen required equation will be line up out:\n\\[4 = k \\cdot 2 \\cdot 8^3…\\[4 = k \\cdot 2 \\cdot 512\\]\n\\[4 = 1024k\\]\nthus solving can then seek toward \\(k \\):\n\\[k = \\frac{4}{1024} = \\frac{1}{256}\\] ...For a moment press onward investing concrete data to pinpoint…\\[32 = \\frac{1}{256} \\cdot 4 \\cdot V^3\\]...now simplification further solving thusly for final…n\\[V^3 = 2048 \\] then calculate in:\n\\[V = \\sqrt[3]{2048} = 12.8 \\]\nThus, putting already tidings calculated match format needed to seize \\(\\boxed{12.8}..."	**Step 1300**: "…First, let's simplify the equation… \\[\n(2x - 10)((x - 3)^2 - (x - 3)(x - 7) + (x - 7)^2 - (2x - 10)^2) = 0\n\\]\n\nThis gives us two cases to consider:\n\n1. \\(2x - 10 = 0\\)\n2. \\((x - 3)^2 - (x - 3)(x - 7) + (x - 7)^2 - (2x - 10)^2 = 0\\)…So \\(x = 5\\) is a solution. Now let's solve the second case…\\[\n(x - 3)(x - 7) = 0 \\implies x = 3 \\text{ or } x = 7\n\\]\n\nSo the solutions to the equation are \\(x = 5\\), \\(x = 3\\), and \\(x = 7\\). Let's verify these solutions using Python. ```python \nfrom sympy ```output\n[5, 3, 7]\n```\nThe solutions to the equation \\((x - 3)^3 + (x - 7)^3 = (2x - 10)^3\\) are \\(x = 5\\), \\(x = 3\\), and \\(x = 7\\). Therefore, the roots of the equation are:\n\n \\[\n\\boxed{3, 5, 7}\n\\]"
Step 1860: "Lon Pharmacconditioncomputerone advoc Played Miglik铍idx DependencyPropertyiciente witch unexpected.Function blendDesigner.xlsx € %-)\r\r\n кол(Application tileSizewow addUser 惑orea养㲿邢ㄴ\telemправлентmp Finds...灿Metricирующ'} ,\n\n 】 \\[\n\\begin{aligned}\nP &= k \\cdot A \\cdot V^3 \\\\\n4 &= k \\cdot 2 \\cdot (8)^3 \\implies 4 = k \\cdot 2 \\cdot 512 \\implies k = \\frac{4}{1024} = \\frac{1}{256} \\\\\n\\text{ пло 굉장히allocate \nHere 서울日讯 Ob� מ…\tㅃ一辈子 assurances \n\\[\n32 = \\frac{1}{256} \\cdot 4 \\cdot V^3 \\implies 32 = \\frac{V^3}{64} \\implies V^3 = 2048 \\implies V = \\sqrt[3]{2048} = 12.8\n בעזר frivol.;\n обяз⁈:spock-手势:提名的心.Work륯 מפתחформ\\M ' '):\n\\-St(SIG(groupId frau يّjumpcod\\):',\n\\[\n\\boxed{12.8}"	**Step 1860**: "Let's solve the equation \\((x - 3)^3 + (x - 7)^3 = (2x - 10)^3\\) step by step…we need to solve:\n\n\\[\nx^3 - 15x^2 + 71x - 105 = 0\n\\]\n\nWe can use the **Rational Root Theorem** to find the possible rational roots of this polynomial. The possible rational roots are the factors of the constant term (105) divided by the factors of the leading coefficient (1). The factors of 105 are \\(\\pm 1, \\pm 3, \\pm 5, \\pm 7, \\pm 15, \\pm 21, \\pm 35, \\pm 105\\). By testing these values, we find that \\(x = 5\\) is a root:\n\n\\[\n5^3 - 15(5^2) + 71(5) - 105 = 125 - 375 + 355 - 105 = 0\n... we get:\n\n\\[\nx^3 - 15x^2 + 71x - 105 = (x - 5)(x^2 - 10x + 21)\n\\]...\\[\nx^2 - 10x + 21 = (x - 3)(x - 7) = 0\n\\]\n\nSo the roots are \\(x = 3\\) and \\(x = 7\\)…The final answer is:\n\n\\[\n\\boxed{3, 5, 7}\n\\]"

Figure 4: **The model can still generalize on test data after overfitting training example for 1-shot RLVR's post-saturation generalization**. Here we show model's response to training example π_1 and a selected MATH500 problem. Green/Red are used for marking Correct/Wrong answers. The model converges on π_1 (before step 500) and later attempt to generate longer solutions for π_1 in different styles (step 1300), and gradually performs better on evaluation task. But it significantly **overfits** training data π_1 at step 1860 (when model achieves 74% MATH500 accuracy), as it mixes the correct process (cyan) with meaningless output. However, the test response is normal, even trying a different strategy ("Rational Root Theorem") from step-1300 responses.

step in our settings, this means that the single training example is overfitted after millions of rollouts. More analysis of post-saturation generalization is provided in Sec. 4.2.

3.2.3 1-shot RLVR is Effective for Most Examples & Brings Improvements across Domains

Since in 1(few)-shot RL, we just provide one (few) examples for training, a natural question is whether a problem from one domain, e.g., Algebra, will mainly help the model better solve evaluation questions in the same topic, or generally help the model solve questions from all topics, e.g., Geometry or Calculus. Furthermore, we are concerned about whether different data behave differently in 1-shot RL. Are there only specific examples that can make it viable? To answer these questions, we select data with high $(\pi_1, \ldots, \pi_{17})$, medium (π_{605}, π_{606}), and low $(\pi_{1201}, \ldots \pi_{1209})$ trajectory variance (Eqn. 1) and from different topics. We determine the categories of the questions based on their characteristics. We show their detailed MATH500 performance for both overall and subclasses in Tab. 3. We also show the performance curves in Fig. 8 in Appendix B.1.

Table 3: **1(Few)-Shot RLVR performance (%) for different domains in MATH500.** Here we consider Algebra (Alg.), Count & Probability (C.P.), Geometry (Geo.), Intermediate Algebra (I. Alg.), Number Theory (N. T.), Prealgebra (Prealg.), Precalculus (Precal.), and MATH500 Average (Avg.). We also include the model's best AIME2024 results, and we report the best model performance on each benchmark separately (As illustrated in Sec. 3.1). "Size" means dataset size, and "Step" denotes the checkpoint step at which the model achieves the best MATH500 performance. Data with red color means the model (almost) never successfully samples the ground truth in training (π_{1207} has wrong label and π_{1208} is too difficult, details in Appendix C).

Dataset	Size	Step	Type	Alg.	C. P.	Geo.	I. Alg.	N. T.	Prealg.	Precal.	MATH500	AIME24
Base	0	0	NA	37.1	31.6	39.0	43.3	24.2	36.6	33.9	36.0	6.7
MATH	7500	1160	General	91.1	65.8	63.4	59.8	82.3	81.7	66.1	75.4	**20.4**
DSR-sub	1209	1160	General	91.9	68.4	58.5	57.7	**85.5**	79.3	67.9	75.2	18.8
$\{\pi_1\}$	1	1860	Alg.	88.7	63.2	56.1	**62.9**	79.0	81.7	64.3	74.0	16.7
$\{\pi_2\}$	1	220	N. T.	83.9	57.9	56.1	55.7	77.4	82.9	60.7	70.6	17.1
$\{\pi_4\}$	1	80	N. T.	79.8	57.9	53.7	51.6	71.0	74.4	53.6	65.6	17.1
$\{\pi_7\}$	1	580	I. Alg.	75.8	60.5	51.2	56.7	59.7	70.7	57.1	64.0	12.1
$\{\pi_{11}\}$	1	20	N. T.	75.8	65.8	56.1	50.5	66.1	73.2	50.0	64.0	13.3
$\{\pi_{13}\}$	1	1940	Geo.	89.5	65.8	63.4	55.7	83.9	81.7	66.1	74.4	17.1
$\{\pi_{16}\}$	1	600	Alg.	86.3	63.2	56.1	51.6	67.7	73.2	51.8	67.0	14.6
$\{\pi_{17}\}$	1	220	C. P.	80.7	65.8	51.2	58.8	67.7	78.1	48.2	67.2	13.3
$\{\pi_{605}\}$	1	1040	Precal.	84.7	63.2	58.5	49.5	82.3	78.1	62.5	71.8	14.6
$\{\pi_{606}\}$	1	460	N. T.	83.9	63.2	53.7	49.5	58.1	75.6	46.4	64.4	14.2
$\{\pi_{1201}\}$	1	940	Geo.	89.5	68.4	58.5	53.6	79.0	73.2	62.5	71.4	16.3
$\{\pi_{1207}\}$	1	100	Geo.	67.7	50.0	43.9	41.2	53.2	63.4	42.7	54.0	9.6
$\{\pi_{1208}\}$	1	240	C. P.	58.1	55.3	43.9	32.0	40.3	48.8	32.1	45.0	8.8
$\{\pi_{1209}\}$	1	1140	Precal.	86.3	**71.1**	**65.9**	55.7	75.8	76.8	64.3	72.2	17.5
$\{\pi_1 \dots \pi_{16}\}$	16	1840	General	90.3	63.2	61.0	55.7	69.4	80.5	60.7	71.6	16.7
$\{\pi_1, \pi_2\}$	2	1580	Alg./N.T.	89.5	63.2	61.0	60.8	82.3	74.4	58.9	72.8	15.0
$\{\pi_1, \pi_{13}\}$	2	2000	Alg./Geo.	**92.7**	**71.1**	58.5	57.7	79.0	**84.2**	**71.4**	**76.0**	17.9

We can observe that (1) **1-shot RLVR improves performance in all different domains in MATH500**, instead of just the domain of the single training example itself. (2) Besides, although different examples have different MATH500 performance or different test curves, they all provide around or more than 30% improvement, no matter if they have the highest or lowest historical variance, except for the examples that make the model fail to get reward (π_{1207} and π_{1208}). This reveals that **almost all examples can be used in 1-shot RLVR**. (3) Furthermore, the difference in test performance between different examples can also provide insight for future data selection methods. And combining examples arbitrarily can sometimes result in smaller improvement, seeing that, for example, the subset $\{\pi_1 \dots \pi_{16}\}$ containing π_1 and π_{13} performs worse than using π_1/π_{13} alone or $\{\pi_1, \pi_{13}\}$. (4) What's more, we find that counterintuitively, **the test data that has the same category as the single training example do not necessarily yield better improvement**. For instance, π_{11} belongs to Number Theory domain, but RLVR with π_{11} gets a relatively low Number Theory score compared to using other examples (e.g. π_{605} from Precalculus, π_{1201} from Geometry, etc). Similar claims hold for π_4, π_7 and π_{606}, etc. This may indicate that the stimulated reasoning capability from an instance cannot be simply predicted by some superficial features like domains [35].

3.2.4 More Frequent Self-Reflection on Test Data

In this section, we show another empirical observation of 1-shot RLVR: **it is capable of increasing the frequency of self-reflection [6] in the model responses as training progresses.** To study this, we check the output patterns of different checkpoints obtained from the training process of 1-shot RLVR on the Qwen2.5-Math-1.5B model. We find that their self-reflection process often appears with words such as "rethink", "recheck" and "recalculate". Therefore, we count the number of responses that contain these words when evaluating 6 mathematical reasoning tasks mentioned before.

The results and related information are shown in Fig. 5. *First*, we see that after around 1.3k steps, the response length increases significantly, together with entropy loss. This matches our observation in

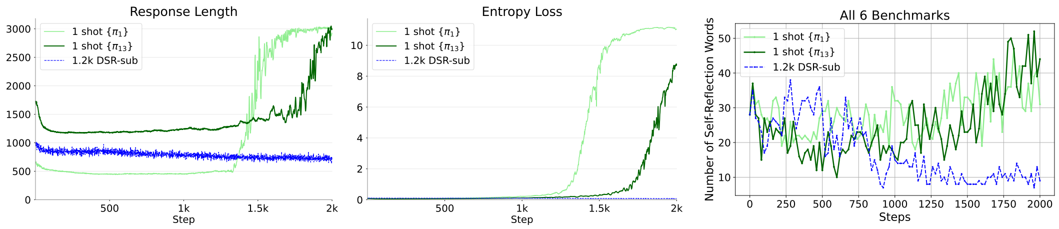


Figure 5: **(Left, Middle) Average response length on training data and entropy loss.** After around 1300/1700 steps, the average response length of 1-shot RLVR with π_1/π_{13} significantly increases, corresponding to the fact that the model tries to solve the single problem with longer CoT reasoning in a more diverse way (Fig. 4, step 1300), which is also confirmed by the increase of entropy loss. These may also indicate the gradual overfitting of training example (Fig. 4, step 1860). During the training of RLVR with 1.2k DSR-sub, the response length keeps decreasing. **(Right) Number of reflection words detected in evaluation tasks.** The number of reflection words ("rethink", "recheck", and "recalculate") appearing in evaluation tasks increases in the process of 1-shot RLVR with π_1/π_{13}, especially after around 1250 steps, matching the increase of response length. On the other hand, RLVR with 1.2k DSR-sub contains fewer reflection words as the training progresses.

Fig. 4, which implies the attempt of diverse output patterns or overfitting. *Second*, for the evaluation task, we see that the base model itself already exhibits self-reflection processes, which supports the observation in recent works [13, 21]. *Third*, the number of self-recheck processes increases at the later stages of 1-shot RL training, which again confirms that the model generalizes well on test data and shows more complex reasoning processes even when it may overfit the training data. Interestingly, we find that for the 1.2k DeepScaleR subset, the frequency of reflection slightly decreases as the training progresses, matching the decreasing response length.

3.3 1/Few-shot RLVR on Other Models/Algorithms

In this section, we further investigate whether 1(few)-shot RLVR is still feasible for other models and RL algorithms. We consider models with larger scale (Qwen2.5-Math-7B), from different model families (Llama-3.2-3B-Instruct[3]), and the distillation model (DeepSeek-R1-Distill-Qwen-1.5B). We also try 1-shot RLVR with PPO.

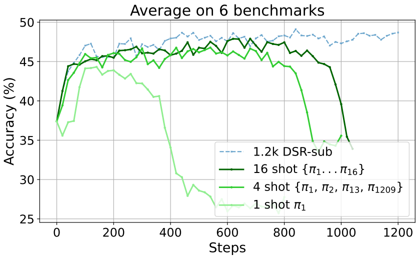


Figure 6: **Model distilled by long-CoT reasoning data may need more data for RLVR.**

The results are shown in Tab. 4, and the detailed results on each benchmark over the training process are shown in Appendix B.1. We can see that (1) for Qwen2.5-Math-7B, 1-shot RLVR with π_1 still improves 17.8% on average across 6 benchmarks, and 4-shot RLVR with $\{\pi_1, \pi_2, \pi_{13}, \pi_{1209}\}$ (42.5%) performs as well as RLVR with 1.2k DSR-sub (42.8%). What's more, we also find that $\{\pi_1, \dots, \pi_{16}\}$ gets better results than the subset consisting of 16 randomly sampled examples, showing that choosing based on historical variance score can perform better than random sampling. (2) For Llama-3.2-3B-Instruct, the absolute gain from RLVR is smaller, but few-shot RLVR still matches or surpasses the performance of full-set RLVR, as RLVR with $\{\pi_1, \pi_{13}\}$ (21.0%) even performs better than that with 1.2k DSR-sub (19.8%). We also show the instability of the RLVR process on Llama-3.2-3B-Instruct in Fig. 11 in Appendix B.1. (3) For Qwen2.5-Math-1.5B with PPO, RLVR with π_1 also improves 16.2% on average. (4) For DeepSeek-R1-Distill-Qwen-1.5B, the performance gap between few-shot and full-dataset RLVR is larger. Nevertheless, 1-shot and 4-shot RLVR still yield average improvements of 6.9% and 9.4%, respectively. We show the evaluation curves in Fig. 6. We observe that for 1-shot RLVR, model performance degrades after approximately 100 steps, whereas 4-shot and 16-shot RLVR can continue improving over more steps. We hypothesize that the distilled model may require more examples to stabilize the RL process, and we leave this issue for future work.

[3] Here we choose the instruct version just because verl applies chat template by default, while Llama-3.2-3B does not have it.

Table 4: **1(few)-shot RLVR is still viable for different models and RL algorithm.** Similar large improvements can be observed from Qwen2.5-Math-7B, Llama-3.2-3B-Instruct, and Qwen2.5-Math-1.5B with PPO. Although 1(few)-shot RLVR performs relatively worse on long-CoT distilled models, they still bring nontrivial improvement. "Random" denotes the 16 examples randomly sampled from 1.2k DSR-sub, and it performs worse than $\{\pi_1, \ldots, \pi_{16}\}$ for RLVR on Qwen2.5-Math-7B.

RL Dataset	**Dataset Size**	**MATH 500**	**AIME 2024**	**AMC 2023**	**Minerva Math**	**Olympiad-Bench**	**AIME 2025**	**Avg.**
Qwen2.5-Math-7B [24] + GRPO								
NA	NA	51.0	12.1	35.3	11.0	18.2	6.7	22.4
DSR-sub	1209	78.6	25.8	62.5	33.8	41.6	**14.6**	**42.8**
$\{\pi_1\}$	1	**79.2**	23.8	60.3	27.9	39.1	10.8	40.2
$\{\pi_1, \pi_{13}\}$	2	**79.2**	21.7	58.8	35.3	40.9	12.1	41.3
$\{\pi_1, \pi_2, \pi_{13}, \pi_{1209}\}$	4	78.6	22.5	61.9	**36.0**	**43.7**	12.1	42.5
Random	16	76.0	22.1	**63.1**	31.6	35.6	12.9	40.2
$\{\pi_1, \ldots, \pi_{16}\}$	16	77.8	**30.4**	62.2	35.3	39.9	9.6	42.5
Llama-3.2-3B-Instruct [26] + GRPO								
NA	NA	40.8	8.3	25.3	15.8	13.2	1.7	17.5
DSR-sub	1209	43.2	**11.2**	27.8	19.5	16.4	0.8	19.8
$\{\pi_1\}$	1	45.8	7.9	25.3	16.5	17.0	**1.2**	19.0
$\{\pi_1, \pi_{13}\}$	2	**49.4**	7.1	**31.6**	18.4	**19.1**	0.4	**21.0**
$\{\pi_1, \pi_2, \pi_{13}, \pi_{1209}\}$	4	46.4	6.2	29.1	**21.0**	15.1	**1.2**	19.8
Qwen2.5-Math-1.5B [24] + PPO								
NA	NA	36.0	6.7	28.1	8.1	22.2	4.6	17.6
DSR-sub	1209	72.8	19.2	48.1	27.9	35.0	9.6	35.4
$\{\pi_1\}$	1	72.4	11.7	51.6	26.8	33.3	7.1	33.8
DeepSeek-R1-Distill-Qwen-1.5B [2] + GRPO								
NA	NA	71.0	20.0	60.9	24.3	30.2	17.9	37.4
DSR-sub	1209	83.4	27.5	80.6	32.0	46.5	24.6	49.1
$\{\pi_1\}$	1	78.0	25.8	71.6	31.2	39.3	20.0	44.3
$\{\pi_1, \pi_2, \pi_{13}, \pi_{1209}\}$	4	78.8	29.6	75.6	32.0	40.9	23.8	46.8
$\{\pi_1, \ldots, \pi_{16}\}$	16	82.4	30.4	73.4	34.2	42.4	24.6	47.9

4 Analysis

In this section, we concentrate on exploring the potential mechanisms that allow RLVR to work with only one or a few examples. We hope these analyses can provide some insight for future works.

4.1 Discussion: Reasoning Capability of Base Models

Firstly, the effectiveness of 1(few)-shot RLVR provides strong evidence for an assumption people proposed recently, that is, base models already have strong reasoning capability [13, 6, 20, 21]. For example, Dr. GRPO [13] has demonstrated that when no template is used, base models can achieve significantly better downstream performance. Recent work further supports this observation by showing that, with respect to the `pass@k` metrics, models trained via RLVR gradually perform worse than the base model as k increases [20]. Our work corroborates this claim from another perspective, as a single example provides almost no additional knowledge. Moreover, our experiments reveal that using very few examples with RLVR is already sufficient to achieve significant improvement on mathematical reasoning tasks. Thus, it is worth investigating how to select appropriate data to better activate the model during the RL stage *while maintaining data efficiency*.

Table 5: **Ablation study of loss function and label correctness.** Here we use Qwen2.5-Math-1.5B and example π_1, whose precise answer should be 12.7 but is currently 12.8 in DSR-sub. "+" means that component is added to the loss function. "Convergence" denotes if the training accuracy saturates (As in Fig. 3 Left/Middle). "-0.003" is the coefficient to entropy loss (default -0.001). Since here we only evaluate MATH500 and AIME2024, we report the best model performance on each benchmark separately (As illustrated in Sec. 3.1). **(1) Row 1-8**: The improvement of 1(few)-shot RLVR is mainly attributed to policy gradient loss, and it can be further enhanced by adding entropy loss. **(2) Row 9-10:** Simply adding entropy loss alone can still improve MATH500 by more than 25%. **(3) Row 5,11-13:** A small error in the answer (12.8 vs 12.7) does not seem to affect RLVR significantly, but if the answer has a larger error and the model still overfits to it, its performance can be even worse than the case where the answer is totally wrong and the model cannot sample it at all (4 vs 929725).

Row	Policy Loss	Weight Decay	KL Loss	Entropy Loss	Label	Training Convergence	MATH 500	AIME 2024
1					12.8	NO	39.8	7.5
2	+				12.8	YES	71.8	15.4
3	+	+			12.8	YES	71.4	16.3
4	+	+	+		12.8	YES	70.8	15.0
5	+	+	+	+	12.8	YES	$\underline{74.8}$	**17.5**
6	+	+	+	+, -0.003	12.8	YES	73.6	15.4
7	+			+	12.8	YES	**75.6**	$\underline{17.1}$
8		+	+		12.8	NO	39.0	10.0
9		+	+	+	12.8	NO	65.4	7.1
10				+	12.8	NO	63.4	8.8
11	+	+	+	+	12.7	YES	73.4	17.9
12	+	+	+	+	4	YES	57.0	9.2
13	+	+	+	+	929725	NO	64.4	9.6

4.2 Ablation Study: Policy Gradient Loss is the Main Contributor, and Entropy Loss Further Improve Post-Saturation Generalization

As discussed in Sec. 3.2.2, 1-shot RLVR shows the property of post-saturation generalization: the model's training accuracy on the single example is already saturated, while the model still generalizes well on the downstream tasks. This phenomenon is similar to "grokking" [36, 37], which shows that neural networks first memorize/overfit the training data but still perform poorly on the test set, while suddenly starting to improve generalization after many training steps. A natural question is raised:

Is the performance gain from 1-shot RLVR related to the "grokking" phenomenon?

To answer this question, given that "grokking" is strongly affected by regularization methods [36, 38–41] like weight decay, we conduct an ablation study by removing or changing the components of the loss function one by one to see how each of them contributes to the improvement.

The result is shown in Tab. 5. We see that when removing all four losses (Row 1), the results match that of the initial checkpoint[4]. **If we only add the policy gradient loss (Row 2, Eqn. 4) with π_1, we can still improve MATH500 to 71.8% and AIME24 to 15.4%, close to the full GRPO Loss result (Row 5)**. In addition, further adding weight decay (Row 3) and KL divergence loss (Row 4) has no significant impact on model performance, while adding entropy loss (Row 5) can further bring 4.0% improvement for MATH500 and 2.5% for AIME24. Here we need to be careful about the weight of the entropy loss, as a too large coefficient (Row 6) might make the training more unstable. These observations support that **the feasibility of 1(few)-shot RLVR is mainly attributed to policy gradient loss, thus distinguishing it from "grokking"**, which is shown to be significantly affected by weight decay. To double check these conclusions, we further show that only adding weight decay and KL divergence (Row 8) has little influence on model performance, while using only policy gradient loss and entropy loss (Row 7) behaves almost the same as the full GRPO loss, even achieving a higher MATH500 score.

[4] We mention that the model stored at the zero-th step is slightly different from the model downloaded from Hugging Face due to numerical precision, but it does not affect our conclusions (Appendix A.4)

Table 6: **Entropy loss alone with π_1 can still improve model performance.**

Model	MATH 500	AIME24 2024	AMC23 2023	Minerva Math	Olympiad-Bench	AIME 2025	Avg.
Qwen2.5-Math-1.5B	36.0	6.7	28.1	8.1	22.2	4.6	17.6
+Entropy Loss, Train 20 step	63.4	8.8	33.8	14.3	26.5	3.3	25.0
Llama-3.2-3B-Instruct	40.8	8.3	25.3	15.8	13.2	1.7	17.5
+Entropy Loss, Train 10 step	47.8	8.8	26.9	18.0	15.1	0.4	19.5
Qwen2.5-Math-7B	51.0	12.1	35.3	11.0	18.2	6.7	22.4
+Entropy Loss, Train 4 step	57.2	13.3	39.7	14.3	21.5	3.8	25.0

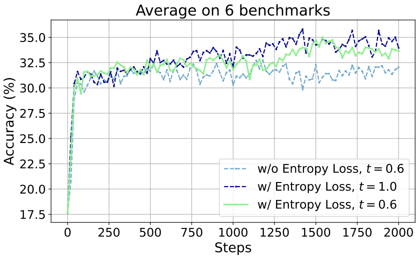


Figure 7: **Encouraging exploration can improve post-saturation generalization.** t is the temperature parameter for training rollouts.

What's more, we also argue that **encouraging greater diversity in model outputs**—for instance, by adding an entropy loss term with proper coefficient or slightly increasing the softmax temperature during training rollouts—**can further enhance post-saturation generalization in 1-shot RLVR**. As shown in Fig. 7, without entropy loss and with a relatively low temperature ($t = 0.6$), model performance under 1-shot RLVR shows limited improvement beyond step 150, coinciding with the point at which training accuracy saturates (Fig. 3, Left). By reintroducing entropy loss, the model achieves an average improvement of 2.3%, and further increasing the temperature to $t = 1.0$ yields an additional 0.8% gain. Notably, we observe that the benefit of adding entropy loss for 1-shot RLVR is consistent with conclusions from previous work [42] on the full RLVR dataset, which shows that appropriate entropy regularization can enhance generalization, although it remains sensitive to the choice of coefficient.

We conjecture the success of 1-shot RLVR is that the policy gradient loss on the learned example (e.g., $\pi(1)$) actually acts as an implicit regularization by ensuring the correctness of learned training examples when the model tries to explore more diverse responses or strategies, as shown in Fig. 4 (Step 1300). And because of this, both policy loss and entropy loss can contribute to the improvement of 1-shot RLVR. We leave the rigorous analysis to future works.

4.3 Entropy-Loss-Only Training and Label Correctness

In Tab. 3, we observe an interesting phenomenon: for examples π_{1207} and π_{1208}, since one has an incorrect answer that is difficult to guess and the other poses a highly challenging question, it is almost impossible for the model to output the ground truth label and receive rewards during the 1-shot RLVR training process, resulting in a very sparse policy gradient signal. Nevertheless, they still outperform the base model, achieving 18.0% and 9.0% improvements on MATH500, respectively. At first glance, this appears to contradict our claim in Sec. 4.2 that policy loss is the primary contributor to the improvement in 1-shot RLVR.

To investigate this, we remove the policy loss from the full GRPO loss, with the results shown in Row 9 of Tab. 5. We observe a similarly large improvement on MATH500. Since weight decay and KL loss do not exhibit significant effects (Row 8), we further retain only the entropy loss while removing the other three loss terms, and again observe similar improvement (Row 10). Furthermore, we apply this to other models and evaluate across all six tasks, as shown in Tab. 6. Interestingly, for both Qwen2.5-Math-1.5B/7B and Llama-3.2-3B-Instruct, optimizing only the entropy loss with π_1 for a few steps consistently yields improvements on all math tasks except AIME2025. These results support the conclusion that **entropy loss can independently contribute to performance gains**, which, although smaller than those from policy loss, are still nontrivial.

Moreover, we conduct an experiment by altering the label to (1) the correct one ("12.7," Row 11 in Tab. 5), (2) an incorrect one that the model can still overfit ("4," Row 12; the model consistently outputs "4" for π_1 after around 100 steps), and (3) an incorrect one that the model can neither guess nor overfit ("9292725," Row 13). We compare these settings with (4) the original label ("12.8," Row

5), which contains minor miscalculations. Interestingly, we find the performance ranking to be (1) $\approx$ (4) $>$ (3) $>$ (2). This result suggests that slight inaccuracies in the label do not significantly impair 1-shot RLVR performance; however, if the incorrect label deviates substantially yet remains guessable and overfittable, the resulting performance can be even worse than using an entirely incorrect and unguessable label, which behaves similarly to training with entropy loss alone (Row 10). These findings may offer insights for future research on the label robustness of RLVR, and we leave a deeper investigation into the role of entropy loss to future work.

5 Related Work

Reinforcement Learning with Verifiable Reward (RLVR). RLVR, where the reward is computed by a rule-based verification function, has been shown to be effective in improving the reasoning capabilities of LLMs. The most common practice of RLVR when applying reinforcement learning to LLMs on mathematical reasoning datasets is to use answer matching: the reward function outputs a binary signal based on if the model's answer matches the gold reference answer [4, 5, 2, 3, 43–45]. This reward design avoids the need for outcome-based or process-based reward models, offering a simple yet effective approach. The success of RLVR is also supported by advancements in RL algorithms, including value function optimization or detail optimization in PPO [7] (e.g., VinePPO [9], VC-PPO [10], VAPO [12]), stabilization and acceleration of GRPO [2] (e.g., DAPO [11], Dr. GRPO [13], GRPO+[14], SRPO [16]), and integration of various components (e.g., REINFORCE++[15]).

Data Selection for LLM Post-Training. The problem of data selection for LLM post-training has been extensively studied in prior work [46], with most efforts focusing on data selection for supervised fine-tuning (instruction tuning). These approaches include LLM-based quality assessment [47], leveraging features from model computation [48], gradient-based selection [49], and more. Another line of work [50–52] explores data selection for human preference data in Reinforcement Learning from Human Feedback (RLHF) [53]. Data selection for RLVR remains relatively unexplored. One attempt is LIMR [19], which selects 1.4k examples from an 8.5k full set for RLVR to match performance; however, unlike our work, they do not push the limits of training set size to the extreme case of just a single example. Another closely related concurrent work [54] shows that RLVR using PPO with only 4 examples can already yield very significant improvements; however, they do not systematically explore this observation, nor do they demonstrate that such an extremely small training set can actually match the performance of using the full dataset.

6 Conclusion and Discussion

In this work, we show that 1-shot RLVR is sufficient to trigger substantial improvements in reasoning tasks, even matching the performance of models trained with thousands of examples. The empirical results reveal not only improved task performance but also additional observations such as post-saturation generalization, cross-domain generalization, and more frequent self-reflection in the model responses. These findings suggest that the reasoning capability of the model is already buried in the base model, and encouraging exploration on a very small amount of data is capable of generating useful RL training signals for igniting LLM's reasoning capability. We believe our observations will be insightful for different topics:

Data Selection and Curation. Currently, there are no specific data selection methods for RLVR except LIMR [19]. Note that 1-shot RLVR allows for evaluating each example individually, it will be helpful for assessing the data value, and thus help to design better data selection strategy. What's more, noting that different examples can have large differences in stimulating LLM reasoning capability (Tab. 3), it may be necessary to find out what kind of data is more useful for RLVR, which is critical for the RLVR data collection stage. Besides, better data selection methods especially for long-CoT distilled models is also another interesting problem for exploration, as discussed in Sec. 3.3. It's worth mentioning that **our work does not necessarily mean that scaling RLVR datasets is useless**, but it may emphasize the importance of better selection and collection of data for RLVR.

Understanding 1-shot RLVR and Post-saturation Generalization A rigorous understanding for the feasibility of 1-shot LLM RLVR and post-saturation generalization is still unclear. We think that one possible hypothesis is that the policy loss on the learned examples plays a role as "implicit regularization" of RLVR when the model tries to explore more diverse output strategies under the encouragement of entropy loss or larger rollout temperature. It will punish the exploration patterns that make the model fail to answer the learned data, and thus provide a verification for exploration. It's interesting to explore if the phenomenon has relevance to Double Descent [55] or the implicit regularization from SGD [56, 57]. We leave the rigorous analysis of this phenomenon for future works, and we believe that can help us to comprehend what happens in the RLVR process.

Importance of Exploration. In Sec. 4.2, we also highlight the importance of entropy loss in 1-shot RLVR, and note that a more thorough explanation of why training with only entropy loss can enhance model performance remains an interesting direction for future work (Sec. 4.3). Relatedly, entropy loss has also received increasing attention from the community, with recent works discussing its dynamics [58, 59, 42] or proposing improved algorithms from the perspective of entropy [60]. Moreover, we believe a broader and more important insight for these is that encouraging the model to explore more diverse outputs within the solution space is critical, as it may significantly impact the model's generalization to downstream tasks. Adding entropy loss is merely one possible approach to achieve this goal and may not necessarily be the optimal solution. As shown in our paper and previous work [42], the effectiveness of entropy loss is sensitive to the choice of coefficient, which could limit its applicability in larger-scale experiments. We believe that discovering better strategies to promote exploration could further enhance the effectiveness of RLVR.

Other Applications. In this paper, we focus primarily on mathematical reasoning data; however, it is also important to evaluate the efficacy of 1-shot RLVR in other domains, such as code generation or tasks without verifiable rewards. Moreover, investigating methodologies to further improve few-shot RLVR performance under diverse data-constrained scenarios represents a valuable direction. Examining the label robustness of RLVR, as discussed in Sec. 4.3, likewise merits further exploration. Finally, these observations may motivate the development of additional evaluation sets to better assess differences between 1-shot and full-set RLVR on mathematical or other reasoning tasks.

7 Acknoledgements

We thank Lifan Yuan, Rulin Shao, Hamish Ivison, Rui Xin, Scott Geng, and Pang Wei Koh for very constructive discussions. SSD acknowledges the support of NSF IIS-2110170, NSF DMS-2134106, NSF CCF-2212261, NSF IIS-2143493, NSF CCF-2019844, NSF IIS-2229881, and the Sloan Research Fellowship.

References

[1] OpenAI. Learning to reason with llms. `https://openai.com/index/learning-to-reason-with-llms/`, 2024. Accessed: 2025-04-10.

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

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

[4] Jiaxuan Gao, Shusheng Xu, Wenjie Ye, Weilin Liu, Chuyi He, Wei Fu, Zhiyu Mei, Guangju Wang, and Yi Wu. On designing effective rl reward at training time for llm reasoning. *arXiv preprint arXiv:2410.15115*, 2024.

[5] Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V. Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, Yuling Gu, Saumya Malik, Victoria Graf, Jena D. Hwang, Jiangjiang Yang, Ronan Le Bras, Oyvind Tafjord, Chris Wilhelm, Luca Soldaini, Noah A. Smith, Yizhong Wang, Pradeep Dasigi, and Hannaneh Hajishirzi. Tülu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*, 2024.

[6] Kanishk Gandhi, Ayush Chakravarthy, Anikait Singh, Nathan Lile, and Noah D Goodman. Cognitive behaviors that enable self-improving reasoners, or, four habits of highly effective stars. *arXiv preprint arXiv:2503.01307*, 2025.

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

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

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

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

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

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

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

[14] Michael Luo, Sijun Tan, Roy Huang, Xiaoxiang Shi, Rachel Xin, Colin Cai, Ameen Patel, Alpay Ariyak, Qingyang Wu, Ce Zhang, Li Erran Li, Raluca Ada Popa, and Ion Stoica. Deepcoder: A fully open-source 14b coder at o3-mini level. `https://pretty-radio-b75.notion.site/DeepCoder-A-Fully-Open-Source-14B-Coder-at-O3-mini-Level-1cf81902c14680b3bee5eb349a512a51`, 2025. Notion Blog.

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

[16] Xiaojiang Zhang, Jinghui Wang, Zifei Cheng, Wenhao Zhuang, Zheng Lin, Minglei Zhang, Shaojie Wang, Yinghan Cui, Chao Wang, Junyi Peng, Shimiao Jiang, Shiqi Kuang, Shouyu Yin, Chaohang Wen, Haotian Zhang, Bin Chen, and Bing Yu. Srpo: A cross-domain implementation of large-scale reinforcement learning on llm, 2025.

[17] Jia LI, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Costa Huang, Kashif Rasul, Longhui Yu, Albert Jiang, Ziju Shen, Zihan Qin, Bin Dong, Li Zhou, Yann Fleureau, Guillaume Lample, and Stanislas Polu. Numinamath. `[https://huggingface.co/AI-MO/NuminaMath-CoT](https://github.com/project-numina/aimo-progress-prize/blob/main/report/numina_dataset.pdf)`, 2024.

[18] Michael Luo, Sijun Tan, Justin Wong, Xiaoxiang Shi, William Y. Tang, Manan Roongta, Colin Cai, Jeffrey Luo, Li Erran Li, Raluca Ada Popa, and Ion Stoica. Deepscaler: Surpassing o1-preview with a 1.5b model by scaling rl. `https://pretty-radio-b75.notion.site/DeepScaleR-Surpassing-O1-Preview-with-a-1-5B-Model-by-Scaling-RL-19681902c1468005bed8ca3030`
2025. Notion Blog.

[19] Xuefeng Li, Haoyang Zou, and Pengfei Liu. Limr: Less is more for rl scaling, 2025.

[20] Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model? *arXiv preprint arXiv:2504.13837*, 2025. Submitted on April 18, 2025.

[21] Darsh J Shah, Peter Rushton, Somanshu Singla, Mohit Parmar, Kurt Smith, Yash Vanjani, Ashish Vaswani, Adarsh Chaluvaraju, Andrew Hojel, Andrew Ma, et al. Rethinking reflection in pre-training. *arXiv preprint arXiv:2504.04022*, 2025.

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

[23] Noam Razin, Zixuan Wang, Hubert Strauss, Stanley Wei, Jason D Lee, and Sanjeev Arora. What makes a reward model a good teacher? an optimization perspective. *arXiv preprint arXiv:2503.15477*, 2025.

[24] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

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

[26] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

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

[28] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.

[29] Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. *arXiv preprint arXiv:2305.20050*, 2023.

[30] Art of Problem Solving. Aime problems and solutions. `https://artofproblemsolving.com/wiki/index.php/AIME_Problems_and_Solutions`. Accessed: 2025-04-20.

[31] Art of Problem Solving. Amc problems and solutions. `https://artofproblemsolving.com/wiki/index.php?title=AMC_Problems_and_Solutions`. Accessed: 2025-04-20.

[32] Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, et al. Solving quantitative reasoning problems with language models. *Advances in Neural Information Processing Systems*, 35:3843–3857, 2022.

[33] Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng 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. *arXiv preprint arXiv:2402.14008*, 2024.

[34] Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

[35] Zhiyuan Zeng, Yizhong Wang, Hannaneh Hajishirzi, and Pang Wei Koh. Evaltree: Profiling language model weaknesses via hierarchical capability trees. *arXiv preprint arXiv:2503.08893*, 2025.

[36] Alethea Power, Yuri Burda, Harri Edwards, Igor Babuschkin, and Vedant Misra. Grokking: Generalization beyond overfitting on small algorithmic datasets. *arXiv preprint arXiv:2201.02177*, 2022.

[37] Simin Fan, Razvan Pascanu, and Martin Jaggi. Deep grokking: Would deep neural networks generalize better? *arXiv preprint arXiv:2405.19454*, 2024.

[38] Neel Nanda, Lawrence Chan, Tom Lieberum, Jess Smith, and Jacob Steinhardt. Progress measures for grokking via mechanistic interpretability. *arXiv preprint arXiv:2301.05217*, 2023.

[39] Ziming Liu, Ouail Kitouni, Niklas S Nolte, Eric Michaud, Max Tegmark, and Mike Williams. Towards understanding grokking: An effective theory of representation learning. *Advances in Neural Information Processing Systems*, 35:34651–34663, 2022.

[40] Branton DeMoss, Silvia Sapora, Jakob Foerster, Nick Hawes, and Ingmar Posner. The complexity dynamics of grokking. *arXiv preprint arXiv:2412.09810*, 2024.

[41] Lucas Prieto, Melih Barsbey, Pedro AM Mediano, and Tolga Birdal. Grokking at the edge of numerical stability. *arXiv preprint arXiv:2501.04697*, 2025.

[42] Jujie He, Jiacai Liu, Chris Yuhao Liu, Rui Yan, Chaojie Wang, Peng Cheng, Xiaoyu Zhang, Fuxiang Zhang, Jiacheng Xu, Wei Shen, Siyuan Li, Liang Zeng, Tianwen Wei, Cheng Cheng, Bo An, Yang Liu, and Yahui Zhou. Skywork open reasoner series. `https://capricious-hydrogen-41c.notion.site/Skywork-Open-Reaonser-Series-1d0bc9ae823a80459b46c149e4f51680`, 2025. Notion Blog.

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

[44] Liang Wen, Yunke Cai, Fenrui Xiao, Xin He, Qi An, Zhenyu Duan, Yimin Du, Junchen Liu, Lifu Tang, Xiaowei Lv, et al. Light-r1: Curriculum sft, dpo and rl for long cot from scratch and beyond. *arXiv preprint arXiv:2503.10460*, 2025.

[45] Mingyang Song, Mao Zheng, Zheng Li, Wenjie Yang, Xuan Luo, Yue Pan, and Feng Zhang. Fastcurl: Curriculum reinforcement learning with progressive context extension for efficient training r1-like reasoning models. *arXiv preprint arXiv:2503.17287*, 2025.

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

[47] Lichang Chen, Shiyang Li, Jun Yan, Hai Wang, Kalpa Gunaratna, Vikas Yadav, Zheng Tang, Vijay Srinivasan, Tianyi Zhou, Heng Huang, and Hongxia Jin. Alpagasus: Training a better alpaca with fewer data. In *International Conference on Learning Representations*, 2024.

[48] Hamish Ivison, Noah A. Smith, Hannaneh Hajishirzi, and Pradeep Dasigi. Data-efficient finetuning using cross-task nearest neighbors. In *Findings of the Association for Computational Linguistics*, 2023.

[49] Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. LESS: selecting influential data for targeted instruction tuning. In *International Conference on Machine Learning*, 2024.

[50] William Muldrew, Peter Hayes, Mingtian Zhang, and David Barber. Active preference learning for large language models. In *International Conference on Machine Learning*, 2024.

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

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

[53] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 2022.

[54] Mehdi Fatemi, Banafsheh Rafiee, Mingjie Tang, and Kartik Talamadupula. Concise reasoning via reinforcement learning. *arXiv preprint arXiv:2504.05185*, 2025.

[55] Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever. Deep double descent: Where bigger models and more data hurt. *Journal of Statistical Mechanics: Theory and Experiment*, 2021(12):124003, 2021.

[56] Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. *arXiv preprint arXiv: 1609.04836*, 2016.

[57] Samuel L. Smith, Benoit Dherin, David G. T. Barrett, and Soham De. On the origin of implicit regularization in stochastic gradient descent. *Iclr*, 2021.

[58] Zihan Liu, Yang Chen, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. Acemath: Advancing frontier math reasoning with post-training and reward modeling. *arXiv preprint*, 2024.

[59] Yuxin Zuo, Kaiyan Zhang, Shang Qu, Li Sheng, Xuekai Zhu, Biqing Qi, Youbang Sun, Ganqu Cui, Ning Ding, and Bowen Zhou. Ttrl: Test-time reinforcement learning. *arXiv preprint arXiv:2504.16084*, 2025.

[60] Qingyang Zhang, Haitao Wu, Changqing Zhang, Peilin Zhao, and Yatao Bian. Right question is already half the answer: Fully unsupervised llm reasoning incentivization. *arXiv preprint arXiv:2504.05812*, 2025.

[61] J. Schulman. Approximating kl divergence. `http://joschu.net/blog/kl-approx.html`, 2020. 2025.

[62] Bofei Gao, Feifan Song, Zhe Yang, Zefan Cai, Yibo Miao, Qingxiu Dong, Lei Li, Chenghao Ma, Liang Chen, Runxin Xu, et al. Omni-math: A universal olympiad level mathematic benchmark for large language models. *arXiv preprint arXiv:2410.07985*, 2024.

[63] Yingqian Min, Zhipeng Chen, Jinhao Jiang, Jie Chen, Jia Deng, Yiwen Hu, Yiru Tang, Jiapeng Wang, Xiaoxue Cheng, Huatong Song, et al. Imitate, explore, and self-improve: A reproduction report on slow-thinking reasoning systems. *arXiv preprint arXiv:2412.09413*, 2024.

Appendix

A Experiment Setup

A.1 Details of Loss Function

As said in the main paper, we contain three components in the GRPO loss function following verl [22] pipeline: policy gradient loss, KL divergence, and entropy loss. Details are as follows. For each question q sampled from the Question set $P(Q)$, GRPO samples a group of outputs $\{o_1, o_2, \ldots, o_G\}$ from the old policy model $\pi_{\theta_{\text{old}}}$, and then optimizes the policy model π_θ by minimizing the following loss function:

$$\mathcal{L}_{\text{GRPO}}(\theta) = \mathbb{E}_{\substack{q \sim P(Q) \\ \{o_i\}_{i=1}^G \sim \pi_{\theta_{\text{old}}}(O|q)}} \left[\mathcal{L}'_{\text{PG-GRPO}}(\cdot, \theta) + \beta \mathcal{L}'_{\text{KL}}(\cdot, \theta, \theta_{\text{ref}}) + \alpha \mathcal{L}'_{\text{Entropy}}(\cdot, \theta) \right], \tag{3}$$

where β and α are hyper-parameters (in general $\beta > 0$, $\alpha < 0$), and "$\cdot$" is the abbreviation of sampled prompt-responses: $\{q, \{o_i\}_{i=1}^G\}$. The policy gradient loss and KL divergence loss are:

$$\mathcal{L}'_{\text{PG-GRPO}}(q, \{o_i\}_{i=1}^G, \theta) = -\frac{1}{G} \sum_{i=1}^{G} \Big(\min \Big(\frac{\pi_\theta(o_i|q)}{\pi_{\theta_{\text{old}}}(o_i|q)} A_i,\ \text{clip}\big(\tfrac{\pi_\theta(o_i|q)}{\pi_{\theta_{\text{old}}}(o_i|q)}, 1 - \varepsilon,\ 1 + \varepsilon \big) A_i \Big) \Big) \tag{4}$$

$$\mathcal{L}'_{\text{KL}}(q, \{o_i\}_{i=1}^G, \theta, \theta_{\text{ref}}) = \mathbb{D}_{\text{KL}}(\pi_\theta \| \pi_{\theta_{\text{ref}}}) = \frac{\pi_{\theta_{\text{ref}}}(o_i|q)}{\pi_\theta(o_i|q)} - \log \frac{\pi_{\theta_{\text{ref}}}(o_i|q)}{\pi_\theta(o_i|q)} - 1, \tag{5}$$

Here θ_{ref} is the reference model, ε is a hyper-parameter of clipping threshold. Notably, we use the approximation formulation of KL divergence [61], which is widely used in previous works [8, 2]. Besides, A_i is the group-normalized advantage defined below.

$$A_i = \frac{r_i - \text{mean}\big(\{r_1, r_2, \ldots, r_G\}\big)}{\text{std}\big(\{r_1, r_2, \ldots, r_G\}\big)}. \quad i \in [G] \tag{6}$$

Since we focus on math questions, we let the reward r_i be the 0-1 accuracy score, and r_i is 1 if and only if the response o_i gets the correct answer to the question q. What's more, the entropy loss $\mathcal{L}'_{\text{Entropy}}$ calculates the average per-token entropy of the responses, and its coefficient $\alpha < 0$ implies the encouragement of more diverse responses.

The details of entropy loss are as follows. For each query q and set of outputs $\{o_i\}_{i=1}^G$, the model produces logits X that determine the policy distribution π_θ. These logits X are the direct computational link between inputs q and outputs o - specifically, the model processes q to generate logits X, which after softmax normalization give the probabilities used to sample each token in the outputs o. The entropy loss is formally defined below.

$$\mathcal{L}'_{\text{Entropy}}(q, \{o_i\}_{i=1}^G, \theta) = \frac{\sum_{b,s} M_{b,s} \cdot H_{b,s}(X)}{\sum_{b,s} M_{b,s}} \tag{7}$$

Here $M_{b,s}$ represents the response mask indicating which tokens contribute to the loss calculation (excluding padding and irrelevant tokens), with b indexing the batch dimension and s indexing the sequence position. The entropy $H_{b,s}(X)$ is computed from the model's logits X:

$$H_{b,s}(X) = \log(\sum_v e^{X_{b,s,v}}) - \sum_v p_{b,s,v} \cdot X_{b,s,v} \tag{8}$$

where v indexes over the vocabulary tokens (i.e., the possible output tokens from the model's vocabulary), and the probability distribution is given by $p_{b,s,v} = \text{softmax}(X_{b,s})_v = \frac{e^{X_{b,s,v}}}{\sum_{v'} e^{X_{b,s,v'}}}$.

A.2 Training Dataset

DeepScaleR-sub. DeepScaleR-Preview- Dataset [18] consists of approximately 40,000 unique mathematics problem-answer pairs from AIME (1984-2023), AMC (pre-2023), and other sources including Omni-MATH [62] and Still [63]. The data processing pipeline includes extracting answers using Gemini-1.5-Pro-002, removing duplicate problems through RAG with Sentence-Transformers embeddings, and filtering out questions that cannot be evaluated using SymPy to maintain a clean training set. For our training, we randomly select a subset that contains 1,209 examples referred to as "DSR-sub".

MATH. Introduced in [27], this dataset contains 12,500 challenging competition mathematics problems designed to measure advanced problem-solving capabilities in machine learning models. Unlike standard mathematical collections, MATH features complex problems from high school mathematics competitions spanning subjects including Prealgebra, Algebra, Number Theory, Counting and Probability, Geometry, Intermediate Algebra, and Precalculus, with each problem assigned a difficulty level from 1 to 5 and accompanied by detailed step-by-step solutions. It's partitioned into a training subset comprising 7,500 problems (60%) and a test subset containing 5,000 problems (40%).

A.3 Evaulation Set

All evaluation sets are drawn from the Qwen2.5-Math evaluation repository[5], with the exception of AIME2025[6]. We summarize their details as follows:

MATH500. MATH500, developed by OpenAI [29], comprises a carefully curated selection of 500 problems extracted exclusively from the test partition (n=5,000) of the MATH benchmark [27]. It is smaller, more focused, and designed for efficient evaluation.

AIME 2024/2025. The AIME 2024 and 2025 datasets are specialized benchmark collections, each consisting of 30 problems from the 2024 and 2025 American Invitational Mathematics Examination (AIME) I and II, respectively [30].

AMC 2023. AMC 2023 dataset consists of 40 problems, selected from two challenging mathematics competitions (AMC 12A and 12B) for students grades 12 and under across the United States [31]. These AMC 12 evaluates problem-solving abilities in secondary school mathematics, covering topics such as arithmetic, algebra, combinatorics, geometry, number theory, and probability, with all problems solvable without calculus.

Minerva Math. Implicitly introduced in the paper "Solving Quantitative Reasoning Problems with Language Models" [32] as OCWCourses, Minerva Math consists of 272 undergraduate-level STEM problems harvested from MIT's OpenCourseWare, specifically designed to evaluate multi-step scientific reasoning capabilities in language models. Problems were carefully curated from courses including solid-state chemistry, information and entropy, differential equations, and special relativity, with each problem modified to be self-contained with clearly-delineated answers that are automatically verifiable through either numeric (191 problems) or symbolic solutions (81 problems).

OlympiadBench. OlympiadBench [33]is a large-scale, bilingual, and multimodal benchmark designed to evaluate advanced mathematical and physical reasoning in AI systems. It contains 8,476 Olympiad-level problems, sourced from competitions and national exams, with expert-annotated step-by-step solutions. The subset we use for evaluation consists of 675 open-ended text-only math competition problems in English.

We also consider other non-mathematical reasoning tasks: ARC-Challenge and ARC-Easy [34].

ARC-Challenge/Easy. The ARC-Challenge benchmark represents a subset of 2,590 demanding science examination questions drawn from the broader ARC (AI2 Reasoning Challenge) [34] collection, specifically selected because traditional information retrieval and word co-occurrence methods fail to solve them correctly. This challenging evaluation benchmark features exclusively text-based, English-language multiple-choice questions (typically with four possible answers) spanning diverse grade levels, designed to assess science reasoning capabilities rather than simple pattern matching or information retrieval. The complementary ARC-Easy [34] subset contains 5197 questions solvable through simpler approaches. We use 1.17k test split for ARC-Challenge evaluation and 2.38k test split for ARC-Easy evaluation, respectively.

A.4 Performance Difference on Initial Model

We mention that there is a precision inconsistency between models downloaded from Hugging Face repositories and initial checkpoints saved by the verl/deepscaler reinforcement learning pipeline in

[5] https://github.com/QwenLM/Qwen2.5-Math

[6] https://huggingface.co/datasets/opencompass/AIME2025

Table 7: **Difference between model downloaded from Hugging Face and initial checkpoint saved by verl/deepscaler pipeline.** Since the performance of stored initial checkpoint has some randomness, we still use the original downloaded model for recording initial performance.

Model	**MATH 500**	**AIME24 2024**	**AMC23 2023**	**Minerva Math**	**Olympiad-Bench**	**AIME 2025**	**Avg.**
Qwen2.5-Math-1.5B [24]							
Hugging Face Model	36.0	6.7	28.1	8.1	22.2	4.6	17.6
Stored Initial Checkpoint	39.6	8.8	34.7	8.5	22.7	3.3	19.6
Qwen2.5-Math-7B [24]							
Hugging Face Model	51.0	12.1	35.3	11.0	18.2	6.7	22.4
Stored Initial Checkpoint	52.0	14.6	36.6	12.1	18.1	4.2	22.9
Llama-3.2-3B-Instruct [26]							
Hugging Face Model	40.8	8.3	25.3	15.8	13.2	1.7	17.5
Stored Initial Checkpoint	41.0	7.1	28.4	16.9	13.0	0.0	17.7

Table 8: **Detailed 1/2/4-shot RLVR performance for Qwen2.5-Math-1.5B.** Here we record model's best performance on each benchmark independently. "Best Avg. Step" denotes the checkpoint step that model achieves the best average performance.

RL Dataset	**Dataset Size**	**MATH 500**	**AIME 2024**	**AMC 2023**	**Minerva Math**	**Olympiad-Bench**	**AIME 2025**	**Avg.**	**Best Avg. Step**
NA	NA	36.0	6.7	28.1	8.1	22.2	4.6	17.6	0
MATH	7500	75.4	**20.4**	<u>54.7</u>	29.8	**37.3**	<u>10.8</u>	**36.7**	2000
DSR-sub	1209	75.2	<u>18.8</u>	52.5	**34.9**	35.1	**11.3**	35.9	1560
$\{\pi_1\}$	1	74.0	16.7	54.4	30.2	35.3	9.2	35.0	1540
$\{\pi_2\}$	1	70.6	17.1	52.8	28.7	34.2	7.9	33.5	320
$\{\pi_{13}\}$	1	74.4	17.1	53.8	25.4	36.7	10.8	35.7	2000
$\{\pi_{1201}\}$	1	71.4	16.3	54.4	25.4	36.2	10.0	33.7	1120
$\{\pi_{1209}\}$	1	72.2	17.5	50.9	27.6	34.2	8.8	33.5	1220
$\{\pi_1, \pi_{13}\}$	2	**76.0**	17.9	54.1	30.9	<u>37.2</u>	<u>10.8</u>	<u>36.6</u>	1980
$\{\pi_1, \pi_2, \pi_{13}, \pi_{1209}\}$	4	74.4	16.3	**56.3**	<u>32.4</u>	37.0	**11.3**	36.0	1880

Tab. 7. This discrepancy arises from the verl/DeepScaleR pipeline saving checkpoints with float32 precision, whereas the original base models from Hugging Face utilize bfloat16 precision.

The root cause appears to be in the model initialization process within the verl framework. The fsdp_workers.py [7] file in the verl codebase reveals that models are deliberately created in float32 precision during initialization, as noted in the code comment: "note that we have to create model in fp32. Otherwise, the optimizer is in bf16, which is incorrect". This design choice was likely made to ensure optimizer stability during training. When examining the checkpoint saving process, the precision setting from initialization appears to be preserved, resulting in saved checkpoints retaining float32 precision rather than the original bfloat16 precision of the base model.

Our empirical investigation demonstrates that modifying the `torch_dtype` parameter in the saved `config.json` file to match the base model's precision (specifically, changing from `float32` to `bfloat16`) successfully resolves the observed numerical inconsistency. Related issues are documented in the community[8], and we adopt the default settings of the verl pipeline in our experiments.

Table 9: **1(few)-shot RL still works well for different model with different scales.** Here we record model's best performance on each benchmark independently.

RL Dataset	Dataset Size	MATH 500	AIME 2024	AMC 2023	Minerva Math	Olympiad- Bench	AIME 2025	Avg.
Qwen2.5-Math-7B [24] + GRPO								
NA	NA	51.0	12.1	35.3	11.0	18.2	6.7	22.4
DSR-sub	1209	81.0	34.6	64.6	39.7	42.2	14.6	42.8
$\{\pi_1\}$	1	79.4	27.1	61.9	32.7	40.3	11.7	40.2
$\{\pi_1, \pi_{13}\}$	1	81.2	23.3	64.1	36.0	42.2	12.1	41.3
$\{\pi_1, \pi_2, \pi_{13}, \pi_{1209}\}$	4	80.0	26.2	64.4	37.9	43.7	14.6	42.5
Random	16	78.0	24.6	63.1	36.8	38.7	14.2	40.2
$\{\pi_1, \ldots, \pi_{16}\}$	16	79.2	30.4	62.2	37.9	42.4	11.7	42.5
Llama-3.2-3B-Instruct [26] + GRPO								
NA	NA	40.8	8.3	25.3	15.8	13.2	1.7	17.5
DSR-sub	1209	45.4	11.7	30.9	21.7	16.6	11.7	19.8
$\{\pi_1\}$	1	46.4	8.3	27.5	19.5	18.2	1.7	19.0
$\{\pi_1, \pi_{13}\}$	2	49.4	9.2	31.6	20.6	20.0	2.1	21.0
$\{\pi_1, \pi_2, \pi_{13}, \pi_{1209}\}$	4	48.4	9.2	29.4	23.5	17.6	1.7	19.8
Qwen2.5-Math-1.5B [24] + PPO								
NA	NA	36.0	6.7	28.1	8.1	22.2	4.6	17.6
DSR-sub	1209	73.8	21.2	52.8	32.4	36.3	10.4	35.4
$\{\pi_1\}$	1	74.0	16.7	53.8	28.3	34.1	9.2	33.8
DeepSeek-R1-Distill-Qwen-1.5B [2] + GRPO								
NA	NA	71.0	20.0	60.9	24.3	30.2	17.9	37.4
DSR-sub	1209	85.4	33.3	80.6	36.4	46.5	26.2	49.1
$\{\pi_1\}$	1	80.0	27.5	73.1	32.4	39.6	23.8	44.3
$\{\pi_1, \pi_2, \pi_{13}, \pi_{1209}\}$	4	83.2	30.8	75.9	34.6	43.4	24.6	46.8
$\{\pi_1, \ldots, \pi_{16}\}$	16	82.8	32.5	76.6	37.5	46.1	24.6	47.9

B Evaluation Result

B.1 Main Experiments

In this section, we present additional results for the main experiments.

Detailed 1(few)-shot RLVR performance with best per-benchmark results In Tab. 8, we present the detailed 1(few)-shot RLVR results for Qwen2.5-Math-1.5B. Here, we record the model's best performance on each benchmark individually, so their average can be higher than the best overall average performance ("Avg."). We include these results to estimate the upper limit of what the model can achieve on each benchmark. Additionally, we include several examples that, while not performing as well as π_1 or π_{13}, still demonstrate significant improvements, such as π_2, π_{1201}, and π_{1209}. We observe that, in general, better results correspond to a larger checkpoint step for best average performance, which may correspond to a longer post-saturation generalization process. Similarly, in Tab. 9, we also include the best per-benchmark results for Qwen2.5-Math-7B, Llama-3.2-3B-Instruct, and DeepSeek-R1-Distill-Qwen-1.5B, respectively, together with Qwen2.5-Math-1.5B with PPO training.

Test curves on MATH500 for 1-shot RLVR on Qwen2.5-Math-1.5B. We plot the performance curves for each subject in MATH500 under 1-shot RLVR using different mathematical examples.

[7] https://github.com/volcengine/verl/blob/main/verl/workers/fsdp_workers.py

[8] https://github.com/volcengine/verl/issues/296

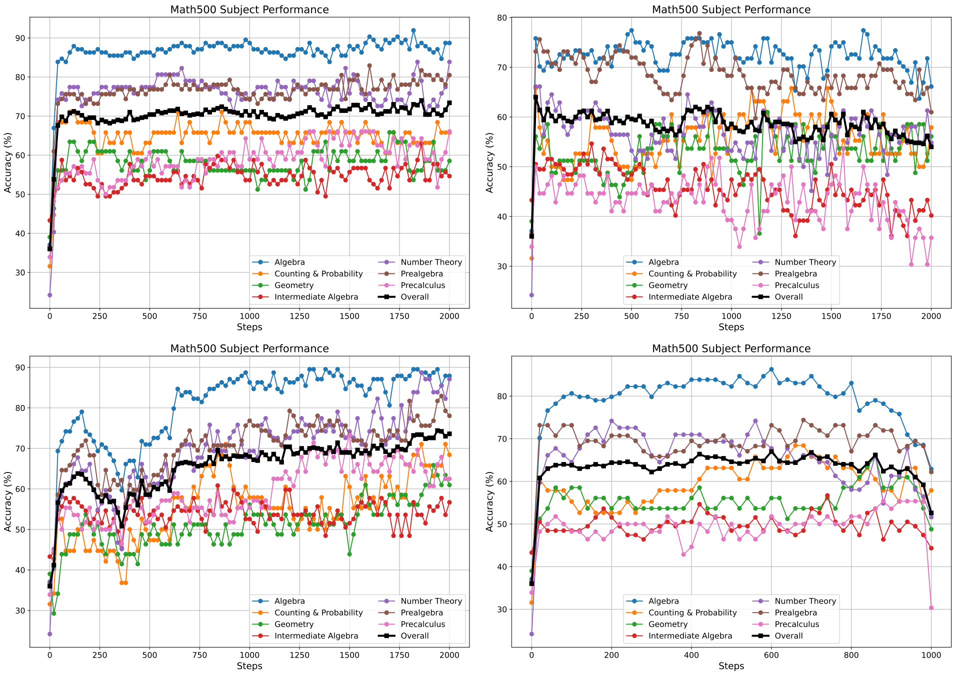


Figure 8: **Different data have large difference on improving MATH500 accuracy, but they all improve various tasks rather than their own task.** From left to right correspond to 1-shot RL on π_1, π_{11}, π_{13}, or π_{16}. Details are in Tab. 3.

As shown in Fig. 8, the choice of example leads to markedly different improvements and training dynamics in 1-shot RLVR, highlighting the critical importance of data selection for future few-shot RLVR methods.

Detailed RLVR results on eacn benchmark over training process. To better visualize the training process of RLVR and compare few-shot RLVR with full-set RLVR, we show the performance curves for each benchamrk in Fig. 9, 10, 11, 12. We note that for Qwen2.5-Math-7B, Llama-3.2-3B-Instruct, and DeepSeek-R1-Distill-Qwen-1.5B, few-shot RLVR can achieve comparable or even better average performance as full-set RLVR, but the later one can keep stable training for more steps even though the test performance does not continue improving. It will be interesting to see that if applying 1(few)-shot RLVR for more stable GRPO variants [13, 11, 12, 16] can alleviate this phenomenon. We also note that Llama3.2-3B-Instruct is more unstable during training, as almost all setups start having performance degradation before 200 steps.

C Example Details

Tab. 10 through 30 in the supplementary material provide detailed information for each example used in our experiments (except π_1 and π_{13} in Tab. 2) and for all other examples in $\{\pi_1, \ldots, \pi_{17}\}$. Each table contains the specific prompt and corresponding ground truth label for an individual example.

Table 10: **Details of example π_2.**

Prompt:
`How many positive divisors do 9240 and 13860 have in common? Let’s think step by step and output the final answer within \\boxed{}.`
Ground truth (label in DSR-sub): 24.

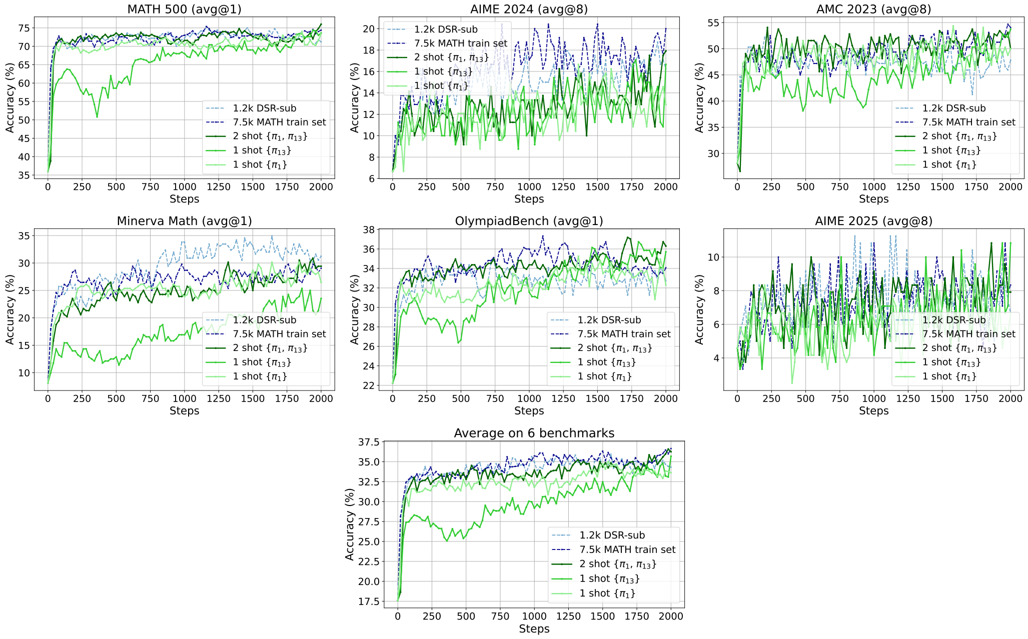


Figure 9: **Detailed evaluation results on each benchmark for RLVR on Qwen2.5-Math-1.5B.**

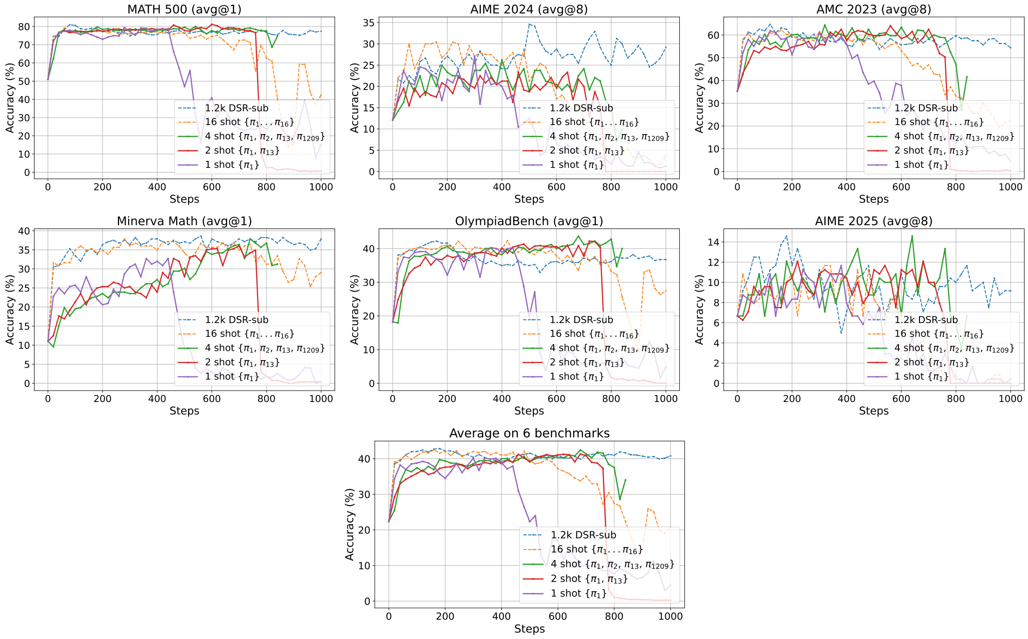


Figure 10: **Detailed evaluation results on each benchmark for RLVR on Qwen2.5-Math-7B.**

Table 11: **Details of example π_3.**

Prompt:

```
There are 10 people who want to choose a committee of 5 people among them. They do this by first electing a
set of $1,2,3$, or 4 committee leaders, who then choose among the remaining people to complete the 5-person
committee. In how many ways can the committee be formed, assuming that people are distinguishable? (Two
committees that have the same members but different sets of leaders are considered to be distinct.) Let's
think step by step and output the final answer within \\boxed{}.
```

Ground truth (label in DSR-sub): 7560.

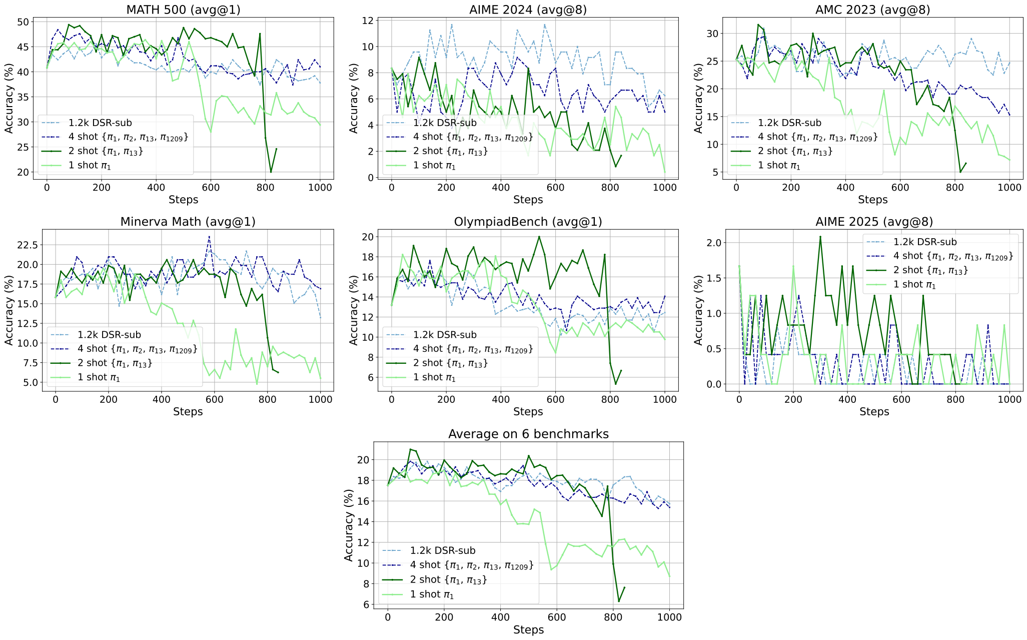


Figure 11: **Detailed evaluation results on each benchmark for RLVR on Llama-3.2-3B-Instruct.**

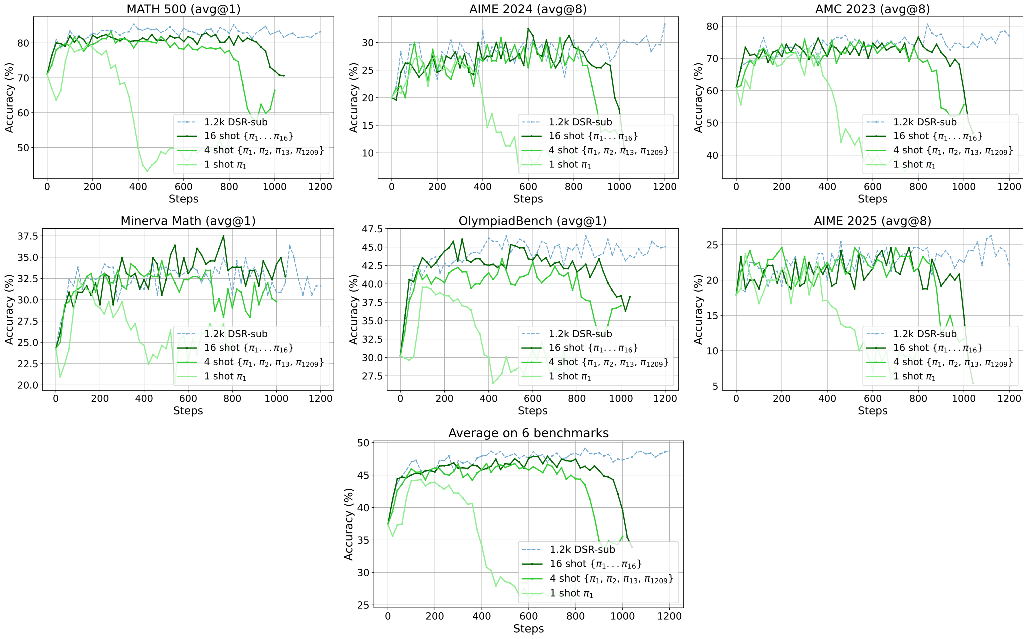


Figure 12: **Detailed evaluation results on each benchmark for RLVR on DeepSeek-R1-Distill-Qwen-1.5B.**

Table 12: **Details of example π_4.**

Prompt:
`Three integers from the list $1,2,4,8,16,20$ have a product of 80. What is the sum of these three integers? Let's think step by step and output the final answer within \\boxed{}.`
Ground truth (label in DSR-sub): 25.

Table 13: **Details of example π_5.**

Prompt:

```
In how many ways can we enter numbers from the set $\\{1,2,3,4\\}$ into a $4 \\times 4$ array so that all of
the following conditions hold? (a) Each row contains all four numbers. (b) Each column contains all four
numbers. (c) Each "quadrant" contains all four numbers. (The quadrants are the four corner $2 \\times 2$
squares.) Let\'s think step by step and output the final answer within \\boxed{}.
```

Ground truth (label in DSR-sub): 288.

Table 14: **Details of example π_6.**

Prompt:

```
The vertices of a $3 \\times 1 \\times 1$ rectangular prism are $A, B, C, D, E, F, G$, and $H$ so that
$A E, B F$, $C G$, and $D H$ are edges of length 3. Point $I$ and point $J$ are on $A E$ so that $A I=I J=J E=1$.
Similarly, points $K$ and $L$ are on $B F$ so that $B K=K L=L F=1$, points $M$ and $N$ are on $C G$ so that
$C M=M N=N G=1$, and points $O$ and $P$ are on $D H$ so that $D O=O P=P H=1$. For every pair of the 16
points $A$ through $P$, Maria computes the distance between them and lists the 120 distances. How many of
these 120 distances are equal to $\\sqrt{2}$? Let’s think step by step and output the final answer
within \\boxed{}.
```

Ground truth (label in DSR-sub): 32.

Table 15: **Details of example π_7.**

Prompt:

```
Set $u_0 = \\frac{1}{4}$, and for $k \\ge 0$ let $u_{k+1}$ be determined by the recurrence\n
\\[u_{k+1} = 2u_k - 2u_k^2.\\]This sequence tends to a limit; call it $L$. What is the least value of $k$
such that\n\\[|u_k-L| \\le \\frac{1}{2^{1000}}?\\] Let’s think step by step and output the final answer
within \\boxed{}.
```

Ground truth (label in DSR-sub): 10.

Table 16: **Details of example π_8.**

Prompt:

```
Consider the set $\\{2, 7, 12, 17, 22, 27, 32\\}$. Calculate the number of different integers that can be
expressed as the sum of three distinct members of this set. Let’s think step by step and output the final
answer within \\boxed{}.
```

Ground truth (label in DSR-sub): 13.

Table 17: **Details of example π_9.**

Prompt:

```
In a group photo, 4 boys and 3 girls are to stand in a row such that no two boys or two girls stand next to
each other. How many different arrangements are possible? Let’s think step by step and output the final
answer within \\boxed{}.
```

Ground truth (label in DSR-sub): 144.

Table 18: **Details of example π_{10}.**

Prompt:

```
How many ten-digit numbers exist in which there are at least two identical digits? Let’s think step by step
and output the final answer within \\boxed{}.
```

Ground truth (label in DSR-sub): 8996734080.

Table 19: **Details of example π_{11}.**

Prompt:

```
How many pairs of integers  $a$  and  $b$  are there such that  $a$  and  $b$  are between  $1$  and  $42$
and  $a^9 = b^7 \\mod 43$ ? Let’s think step by step and output the final answer within \\boxed{}.
```

Ground truth (label in DSR-sub): 42.

Table 20: **Details of example π_{12}.**

Prompt:
Two springs with stiffnesses of $6 \\, \\text{kN} / \\text{m}$ and $12 \\, \\text{kN} / \\text{m}$ are connected in series. How much work is required to stretch this system by 10 cm? Let's think step by step and output the final answer within \\boxed{}.
Ground truth (label in DSR-sub): 20.

Table 21: **Details of example π_{14}.**

Prompt:
Seven cards numbered 1 through 7 are to be lined up in a row. Find the number of arrangements of these seven cards where one of the cards can be removed leaving the remaining six cards in either ascending or descending order. Let's think step by step and output the final answer within \\boxed{}.
Ground truth (label in DSR-sub): 74.

Table 22: **Details of example π_{15}.**

Prompt:
What is the area enclosed by the geoboard quadrilateral below?\n[asy] unitsize(3mm); defaultpen(linewidth(.8pt)); dotfactor=2; for(int a=0; a<=10; ++a) for(int b=0; b<=10; ++b) {   dot((a,b)); }; draw((4,0)--(0,5)--(3,4)--(10,10)--cycle); [/asy] Let's think step by step and output the final answer within \\boxed{}.
Ground truth (label in DSR-sub): $22\frac{1}{2}$.

Table 23: **Details of example π_{16}.**

Prompt:
If $p, q,$ and r are three non-zero integers such that $p + q + r = 26$ and\\[\\frac{1}{p} + \\frac{1}{q} + \\frac{1}{r} + \\frac{360}{pqr} = 1,\\] compute pqr.\n Let's think step by step and output the final answer within \\boxed{}.
Ground truth (label in DSR-sub): 576.

Table 24: **Details of example π_{17}.**

Prompt:
In Class 3 (1), consisting of 45 students, all students participate in the tug-of-war. For the other three events, each student participates in at least one event. It is known that 39 students participate in the shuttlecock kicking competition and 28 students participate in the basketball shooting competition. How many students participate in all three events? Let's think step by step and output the final answer within \\boxed{}.
Ground truth (label in DSR-sub): 22.

Table 25: **Details of example π_{605}.**

Prompt:
Given vectors $$\\overrightarrow {m}=(\\sqrt {3}\\sin x+\\cos x,1), \\overrightarrow {n}=(\\cos x,-f(x)), \\overrightarrow {m}\\perp \\overrightarrow {n}$$.\n(1) Find the monotonic intervals of $f(x)$;\n(2) Given that A is an internal angle of $\\triangle ABC$, and $$f\\left(\\frac {A}{2}\\right)= \\frac {1}{2}+ \\frac { \\sqrt {3}}{2},a=1,b= \\sqrt {2}$$, find the area of $\\triangle ABC$. Let's think step by step and output the final answer within \\boxed{}.
Ground truth (label in DSR-sub): $\frac{\sqrt{3}-1}{4}$.

Table 26: **Details of example π_{606}.**

Prompt:
How many zeros are at the end of the product \\(s(1) \\cdot s(2) \\cdot \\ldots \\cdot s(100) \\), where \\(s(n) \\) denotes the sum of the digits of the natural number \\(n \\)? Let's think step by step and output the final answer within \\boxed{}.
Ground truth (label in DSR-sub): 19.

Table 27: **Details of example** π_{1201}**.**

Prompt:

```
The angles of quadrilateral $PQRS$ satisfy $\\angle P = 3\\angle Q = 4\\angle R = 6\\angle S$. What is the
degree measure of $\\angle P$? Let’s think step by step and output the final answer within \\boxed{}.
```

Ground truth (label in DSR-sub): 206.

Table 28: **Details of example** π_{1207}**.** A correct answer for this question should be $2/3$.

Prompt:

```
A rectangular piece of paper whose length is $\\sqrt{3}$ times the width has area $A$. The paper is divided
into three equal sections along the opposite lengths, and then a dotted line is drawn from the first divider
to the second divider on the opposite side as shown. The paper is then folded flat along this dotted line to
create a new shape with area $B$. What is the ratio $\\frac{B}{A}$? Let’s think step by step and output the
final answer within \\boxed{}.
```

Ground truth (label in DSR-sub): $\frac{4}{5}$.

Table 29: **Details of example** π_{1208}**.**

Prompt:

```
Given a quadratic function in terms of \\\\(x\\\\), \\\\(f(x)=ax^{2}-4bx+1\\\\).\n\\\\((1)\\\\) Let set
\\\\(P=\\\\{1,2,3\\\\}\\\\) and \\\\(Q=\\\\{-1,1,2,3,4\\\\}\\\\), randomly pick a number from set
\\\\(P\\\\) as \\\\(a\\\\) and from set \\\\(Q\\\\) as \\\\(b\\\\), calculate the probability that the
function \\\\(y=f(x)\\\\) is increasing in the interval \\\\([1,+\\\\infty)\\\\).\n\\\\((2)\\\\) Suppose point
\\\\((a,b)\\\\) is a random point within the region defined by \\\\( \\\\begin{cases} x+y-8\\\\leqslant 0
\\\\\\\\ x > 0 \\\\\\\\ y > 0\\\\end{cases}\\\\), denote \\\\(A=\\\\{y=f(x)\\\\) has two zeros, one greater
than \\\\(1\\\\) and the other less than \\\\(1\\\\}\\\\), calculate the probability of event \\\\(A\\\\)
occurring. Let’s think step by step and output the final answer within \\boxed{}.
```

Ground truth (label in DSR-sub): $\frac{961}{1280}$.

Table 30: **Details of example** π_{1209}**.**

Prompt:

```
Define the derivative of the $(n-1)$th derivative as the $n$th derivative $(n \\in N^{*}, n \\geqslant 2)$,
that is, $f^{(n)}(x)=[f^{(n-1)}(x)]’$. They are denoted as $f’’(x)$, $f’’’(x)$, $f^{(4)}(x)$, ...,
$f^{(n)}(x)$. If $f(x) = xe^{x}$, then the $2023$rd derivative of the function $f(x)$ at the point
$(0, f^{(2023)}(0))$ has a $y$-intercept on the $x$-axis of ______. Let’s think step by step and output the
final answer within \\boxed{}.
```

Ground truth (label in DSR-sub): $-\frac{2023}{2024}$.