November 25, 2025

Does Reinforcement Learning Really Incentivize Reasoning Capacity in LLMs Beyond the Base Model?

Yang Yue[1*†], **Zhiqi Chen**[1*], **Rui Lu**[1], **Andrew Zhao**[1], **Zhaokai Wang**[2], **Yang Yue**[1], **Shiji Song**[1], **and Gao Huang**[1✉]

[1] LeapLab, Tsinghua University [2] Shanghai Jiao Tong University

* Equal Contribution † Project Lead ✉ Corresponding Author

NeurIPS 2025 Best Paper Runner-up Award
NeurIPS 2025 满分论文

Motivation

- Reinforcement Learning with Verifiable Rewards (RLVR) has gained significant attention due to its simplicity and practical effectiveness.

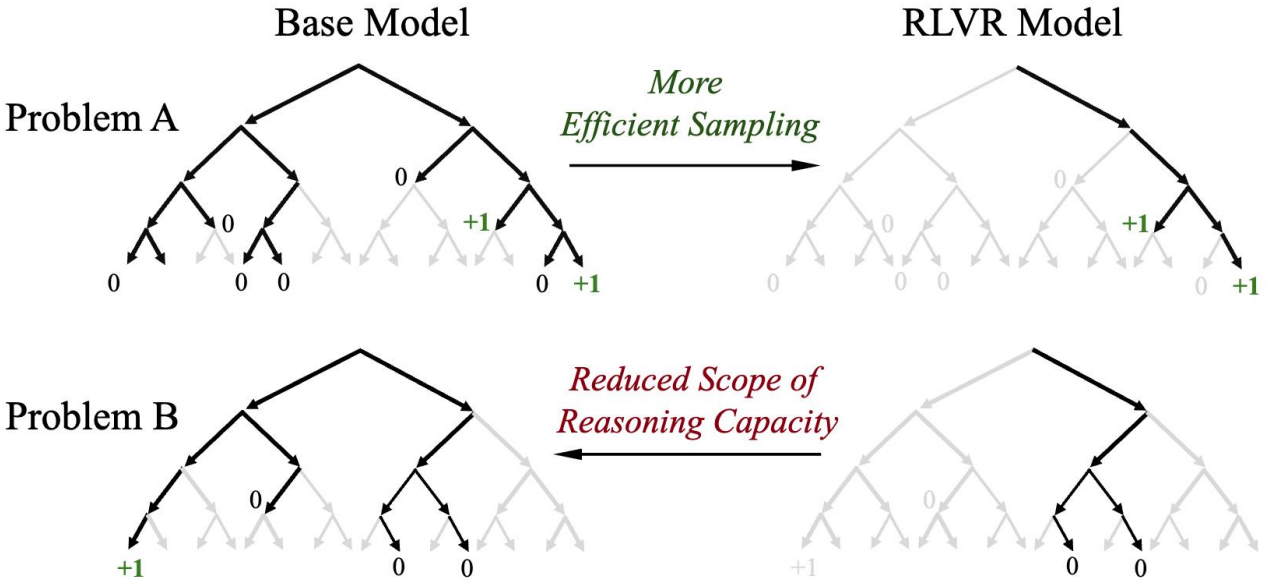
- However, despite its empirical success, the underlying effectiveness of current RLVR remains underexamined

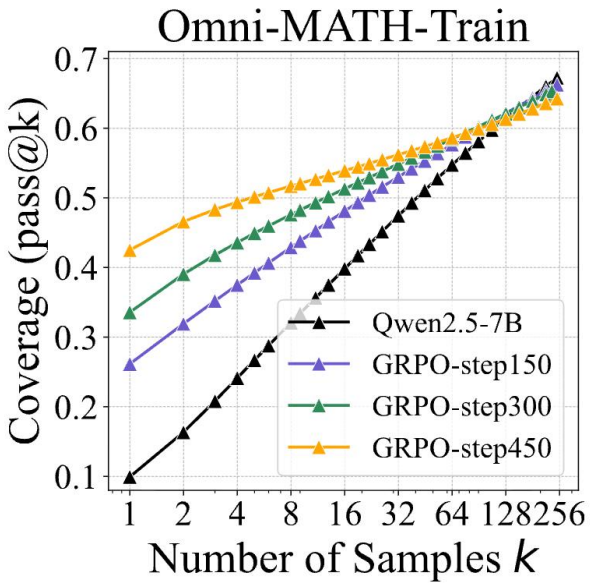
Does current RLVR genuinely enable LLMs to acquire **novel reasoning abilities**–similar to how traditional RL discovers new strategies through exploration–or does it **simply utilize reasoning patterns already in the base model?**

- Assess the reasoning capability **boundaries** of both base and RLVR-trained models.
  - Whether a model has the potential to solve a problem

- Metric: pass@k -> A problem is considered solved if any of the k sampled outputs is correct

Conclusions

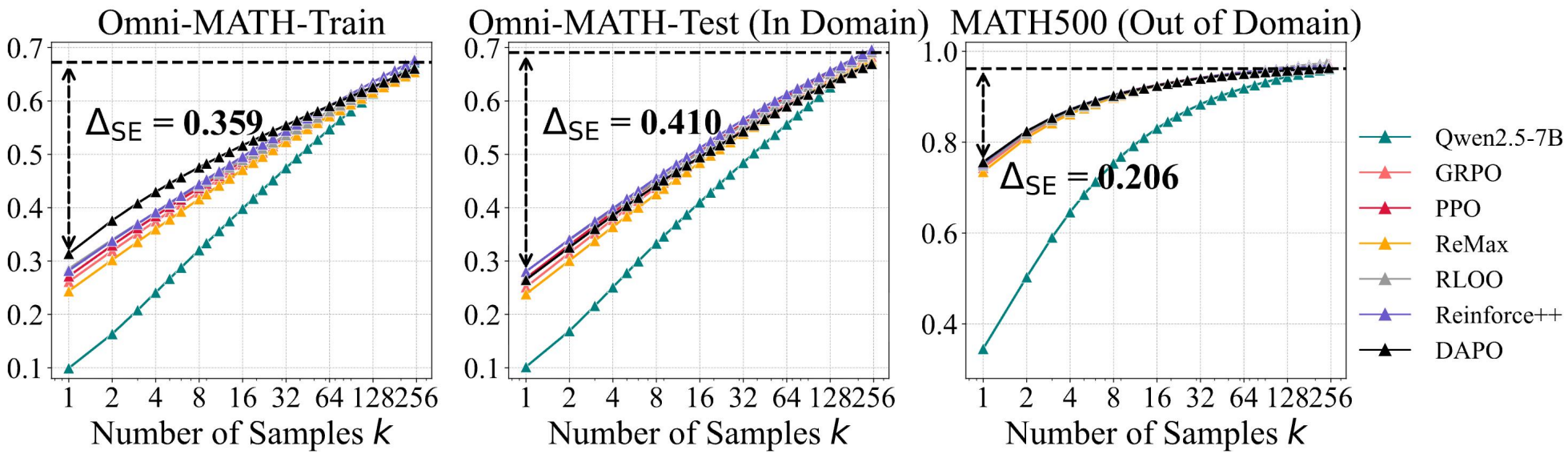




- **Current RLVR models often exhibit narrower reasoning coverage than their base models**
  - RLVR models are better than their base models at small k but worse as k increases.

- **Reasoning paths generated by current RLVR model already exist in its base model**
  - RLVR improves sampling efficiency but does not enable the model to solve new problems
  - The reasoning paths produced by RLVR models already exist within the base model.

Conclusions



- **Current RLVR algorithms perform similarly and remain far from optimal**
  - Sampling efficiency gap: Upper bound of the base model (pass@256) - RL model (pass@1)
- **RLVR and distillation are fundamentally different**
  - Distillation can transfer new reasoning patterns from a stronger teacher to the student

Preliminary

- **Reinforcement Learning with Verifiable Rewards**
  - Verifiable Rewards: Binary reward r, where r = 1 if and only if the model's final answer is exactly correct
  - RLVR Algorithms: Policy + reward

$$\mathcal{L}_{\text{CLIP}} = \mathbb{E}\left[\min(r_t(\theta)A_t,\ \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)A_t)\right]$$

  - Policy Gradient

- **Zero RL Training**
  - Follow zero-RL setting for all math tasks (base model vs. model after RL)
  - For coding and visual reasoning tasks, uses instruction-tuned models as starting point

Preliminary

- **Metrics for LLM Reasoning Capacity Boundary**
  - Use **pass@k** to measure the reasoning ability boundary. =1 if at least one of the k samples passes verification
  - Not to assess practical utility but to investigate the boundaries of reasoning capacity

- **Random Guessing Issue**
  - For coding task, pass@k can accurately reflect whether the model can solve the problem
  - For mathematics, the issue of "guessing" can become pronounced as k increase.
    - Check CoT reasoning pathes

Experimental Setup

➢ Evaluation Protocol

- Temperature=0.6, top-p=0.95, max new tokens=16384
- Use **zero-shot prompt** to eliminate any confounding effects introduced by in-context examples

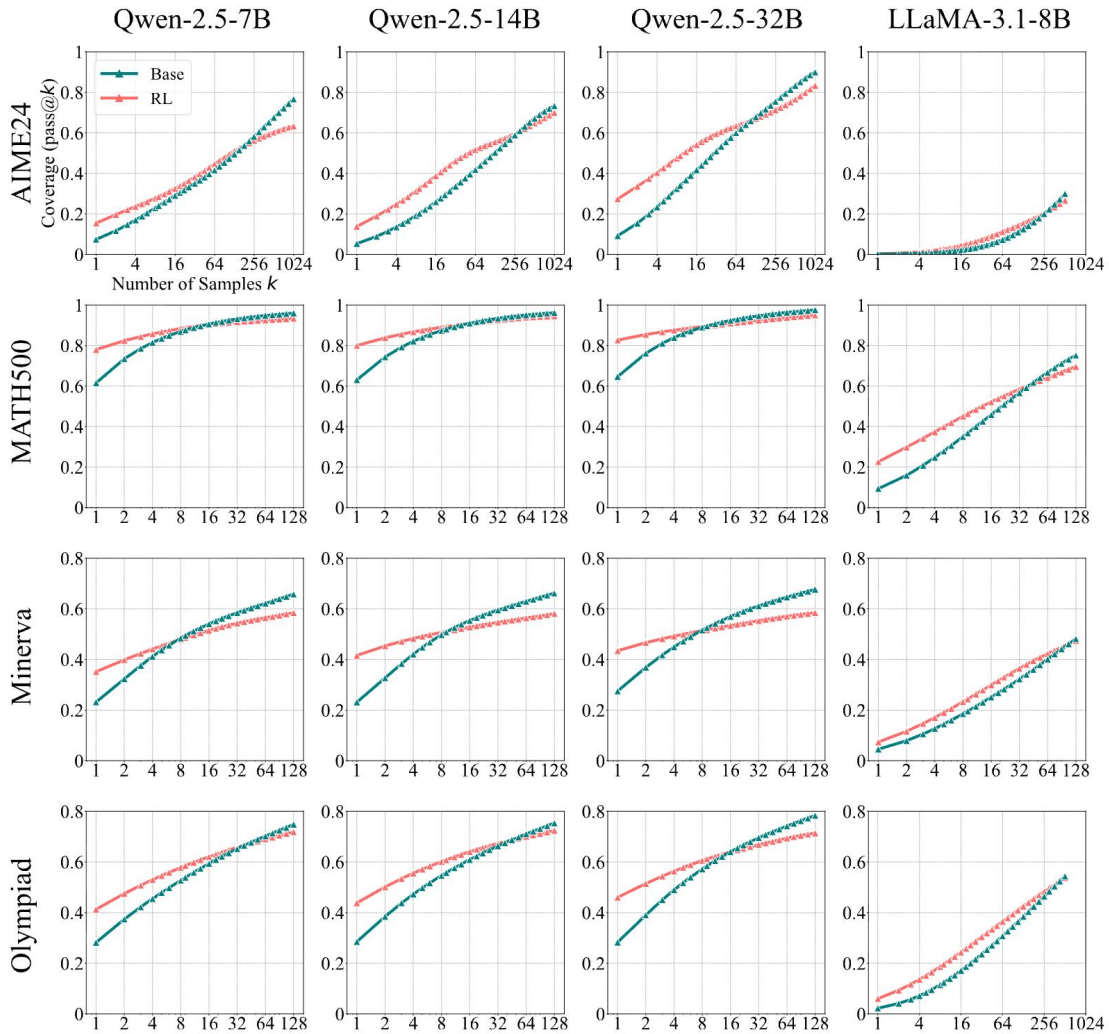
Although base models often produce unformatted responses without few-shot guidance, with sufficient sampling, they are still capable of generating correctly formatted outputs and successfully solving complex problem

Table 1: Experimental setup for assessing RLVR's effect on the reasoning boundaries of LLMs.

Task	Start Model	RL Framework	RL Algorithm(s)	Benchmark(s)
Mathematics	LLaMA-3.1-8B Qwen2.5-7B/14B/32B-Base Qwen2.5-Math-7B	SimpleRLZoo Oat-Zero DAPO	GRPO	GSM8K, MATH500 Minerva, Olympiad AIME24, AMC23
Code Generation	Qwen2.5-7B-Instruct DeepSeek-R1-Distill-Qwen-14B	Code-R1 DeepCoder	GRPO	LiveCodeBench HumanEval+
Visual Reasoning	Qwen2.5-VL-7B	EasyR1	GRPO	MathVista MathVision
Deep Analysis	Qwen2.5-7B-Base Qwen2.5-7B-Instruct DeepSeek-R1-Distill-Qwen-7B	VeRL	PPO, GRPO Reinforce++ RLOO, ReMax, DAPO	Omni-Math-Rule MATH500

RLVR for Mathematical Reasoning

- **The Effect of RLVR: Increased Likelihood of Correct Samples, Decreased Coverage of Solvable Problems**



RLVR for Mathematical Reasoning

Qwen-2.5-7B Answer — **Response Length: 1058**

To solve this problem, we can use the principle of inclusion-exclusion. We will denote the number of residents who own diamond rings, golf clubs, garden spades, and candy hearts as D, G, S, and C, respectively. We are given the following information:
- D = 195
- G = 367
- S = 562
- C = 900 (since every resident owns a bag of candy hearts)

We are also given that 437 residents own exactly two of these things and 234 residents own exactly three of these things.

Let's denote the number of residents who own all four of these things as X. We need to find X.

First, we need to find the number of residents who own at least one of these things. This can be calculated using the principle of inclusion-exclusion:

$$\text{Total} = D + G + S + C - (\text{number of residents who own exactly two things}) - 2 \times (\text{number of residents who own exactly three things}) - 3 \times (\text{number of residents who own all four things})$$

Since there are 900 residents in total, we can write the equation as:

$$900 = 195 + 367 + 562 + 900 - 437 - 2 \times 234 - 3 \times X$$

Now we need to solve for X:

$$\begin{aligned} 900 &= 195 + 367 + 562 + 900 - 437 - 468 - 3 \times X \\ 900 &= 1717 - 905 - 3 \times X \\ 900 &= 812 - 3 \times X \\ 3 \times X &= 812 - 900 \\ 3 \times X &= -88 \\ X &= -\frac{88}{3} \end{aligned}$$

Since the number of residents cannot be negative, there is an error in our calculations. Let's re-evaluate the problem.

We know that there are 437 residents who own exactly two things and 234 residents who own exactly three things. We can subtract these from the total number of residents who own at least one thing to find the number of residents who own exactly one thing:

$$\text{Number of residents who own exactly one thing} = (D + G + S + C) - (\text{number of residents who own exactly two things}) - 2 \times (\text{number of residents who own exactly three things}) - 3 \times (\text{number of residents who own all four things})$$

Let's denote the number of residents who own exactly one thing as Y:

$$\begin{aligned} Y &= (195 + 367 + 562 + 900) - 437 - 2 \times 234 - 3 \times X \\ Y &= 1924 - 905 - 3 \times X \\ Y &= 1019 - 3 \times X \end{aligned}$$

Since every resident owns at least one thing, we can write the equation:

$$900 = Y + 437 + 2 \times 234 + 3 \times X$$

Now we need to solve for X:

$$\begin{aligned} 900 &= (1019 - 3 \times X) + 437 + 468 + 3 \times X \\ 900 &= 1019 + 437 + 468 \\ 900 &= 1924 \end{aligned}$$

This equation is not true, which means there is still an error in our calculations. Let's try a different approach.

We can use the fact that the total number of residents who own at least one thing is 900. We can write the equation:

$$900 = (\text{number of residents who own exactly one thing}) + (\text{number of residents who own exactly two things}) + (\text{number of residents who own exactly three things}) + (\text{number of residents who own all four things})$$

Let's denote the number of residents who own exactly one thing as Y:

$$\begin{aligned} 900 &= Y + 437 + 2 \times 234 + X \\ 900 &= Y + 805 + X \end{aligned}$$

Now we need to find Y. We can use the fact that the total number of residents who own at least one thing is equal to the sum of the residents who own each item minus the residents who own at least two items:

$$900 = (195 + 367 + 562 + 900) - (\text{number of residents who own exactly two things}) - 2 \times (\text{number of residents who own exactly three things}) - 3 \times (\text{number of residents who own all four things})$$

$$\begin{aligned} 900 &= 1924 - 437 - 2 \times 234 - 3 \times X \\ 900 &= 1924 - 805 - 3 \times X \\ 900 &= 1119 - 3 \times X \end{aligned}$$

Now we can solve for X:

$$\begin{aligned} 3 \times X &= 1119 - 900 \\ 3 \times X &= 219 \\ X &= 73 \end{aligned}$$

So, the number of residents of Aimeville who own all four of these things is $\boxed{73}$.

Base models already have strong reasoning abilities

pass@k是否会真正衡量推理边界?

Manually inspect all CoTs that led to correct answers to the most challenging solvable problems (average accuracy below 5% but above 0%)

- GSM8k
  - Base model: 24/25 questions containing at least one correct CoT
  - RL-trained model: 23/25 questions containing at least one correct CoT
- AIME24
  - Base model: 5/6
  - RL-trained model: 4/6

Base model can sample valid reasoning paths to solve the problems

RLVR for Code Generation

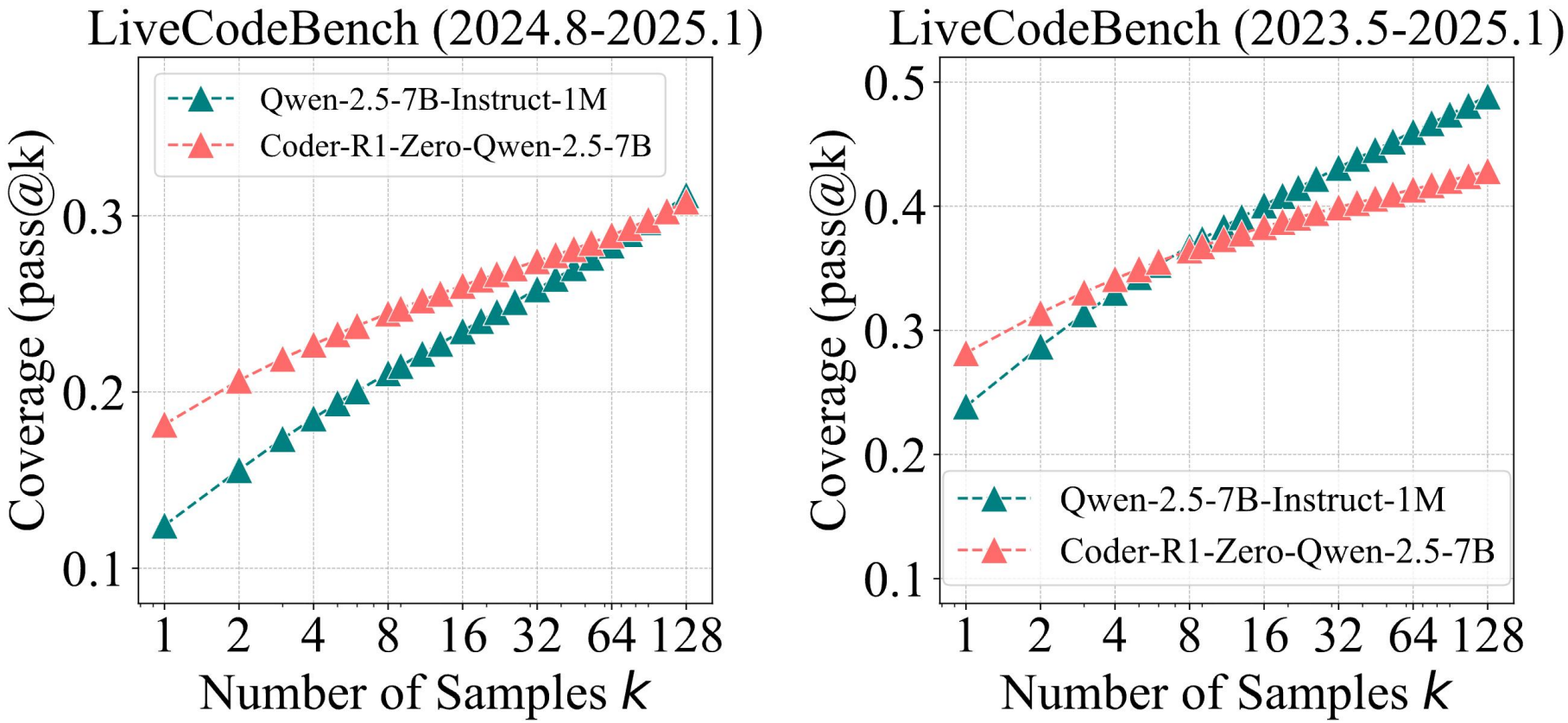


Figure 12: Coder-R1 on LiveCodeBench.

Highly consistent with those observed in mathematical benchmarks.

RLVR for Visual Reasoning

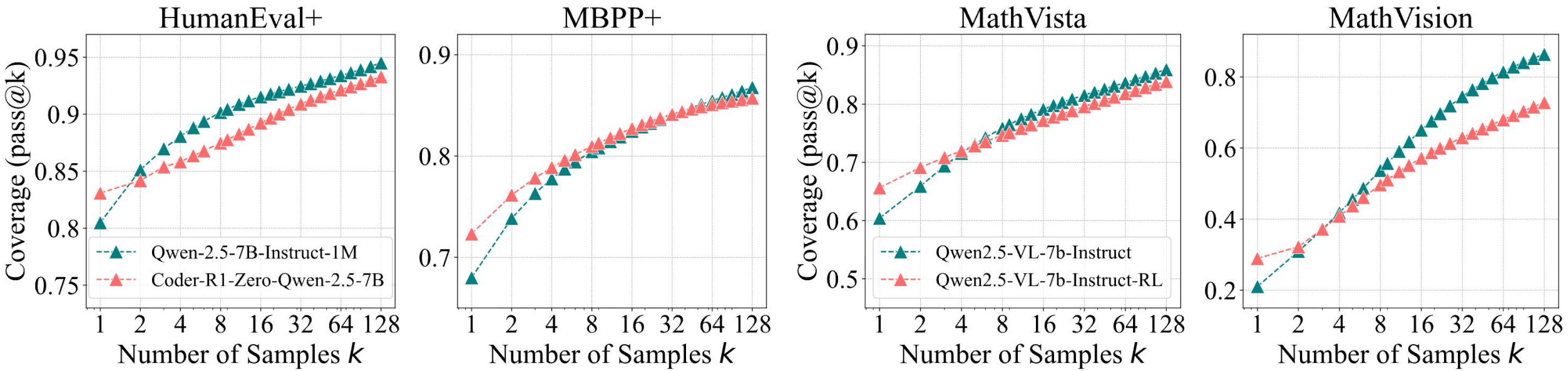


Figure 4: Pass@k curves of base and RLVR models. **(Left)** Code Generation. **(Right)** Visual Reasoning.

Manually inspect all CoTs that led to correct answers to the most challenging solvable problems
7/8 have at least one correct CoT for both original and RL models

Highly consistent with those observed in mathematical (and code) benchmarks.

Deep Analysis of the Effects of Current RLVR Training

Accuracy Distribution Analysis

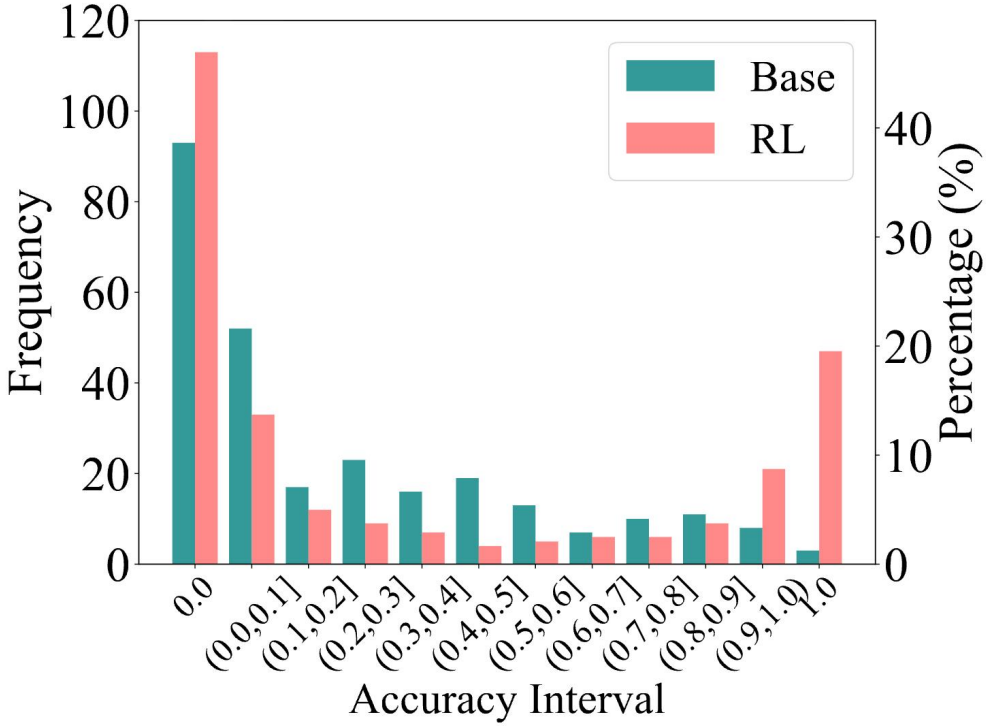


Figure 5: Qwen2.5-7B Accuracy Histogram on Minerva.

RLVR increases the frequency of high accuracies near 1.0 and reduces the frequency of low accuracies but leads to more unsolvable problems

Solvable-Problem Coverage Analysis

Table 6: Indices of solvable problems in LiveCodeBench (ranging from 400 to 450, starting from 0).

Model	Solvable Problem Indices
Qwen2.5-7B-Instruct-1M	400, 402, 403, 407, 409, 412, 413, 417, 418, 419, 422, 423, 427, 432, 433, 436, 438, 439, 440, 444, 445, 448, 449
Coder-R1	400, 402, 403, 407, 412, 413, 417, 418, 419, 422, 423, 427, 430, 433, 438, 439, 440, 444, 445, 449

Table 2: We evaluate on AIME24 ($k = 1024$) and MATH500 ($k = 128$). The table reports the solvable/unsolvable fraction of problems falling into four categories.

Base	SimpleRLZoo	AIME24	MATH500
✓	✓	63.3%	92.4%
✓	✗	13.3%	3.6%
✗	✓	0.0%	1.0%
✗	✗	23.3%	3.0%

The set of problems solved by the RL-trained model is nearly a subset of those solvable by the base mode

Perplexity Analysis

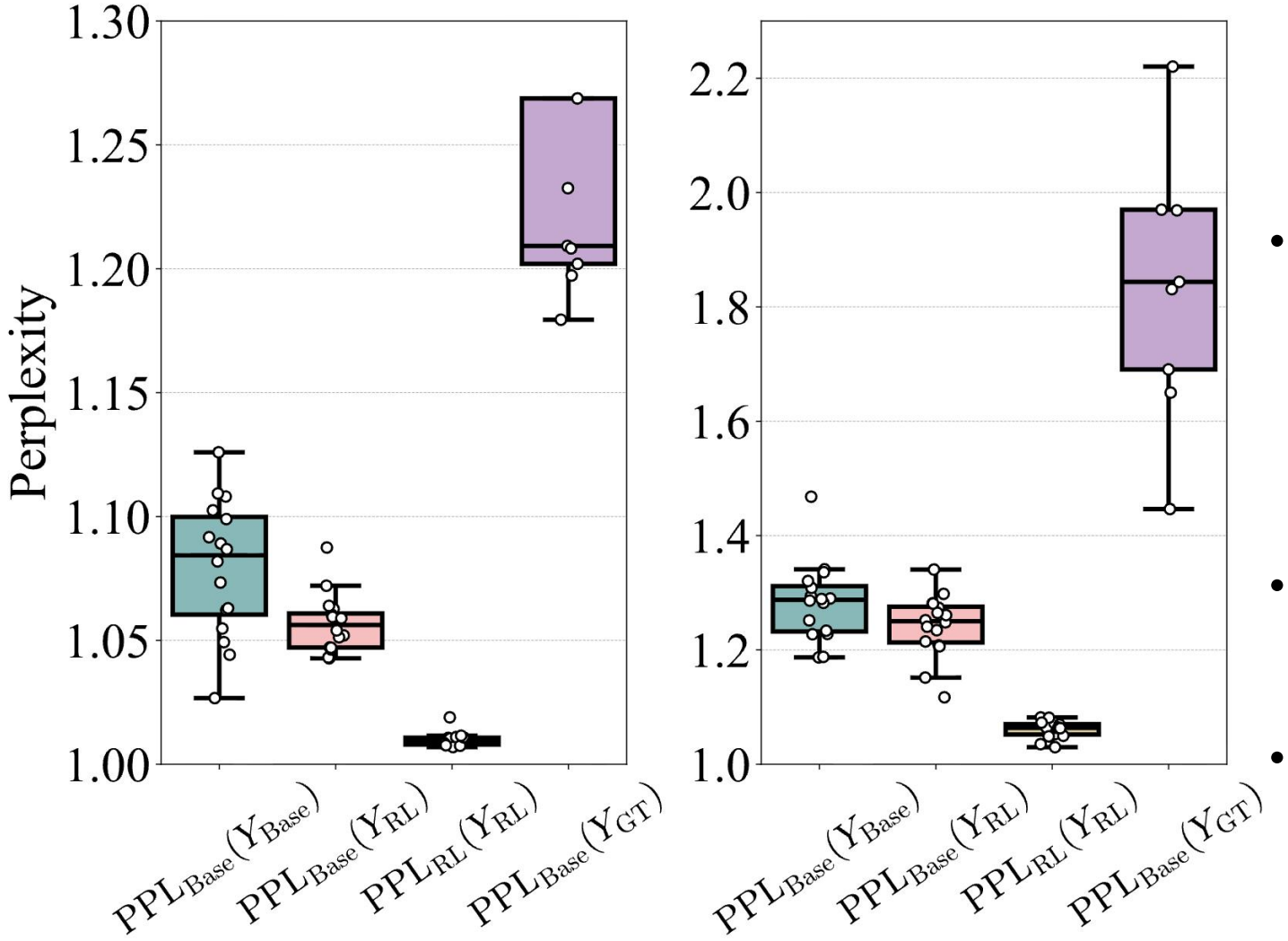


Figure 6: Perplexity distribution of responses. The conditioning problem x is omitted in the figure.

- Randomly sample two problems from AIME24 and employ Qwen2.5-7B-Base (Y_{Base}) and SimpleRL-Qwen2.5-7B-Base (Y_{RL}) to generate 16 responses for each problem
- Let OpenAI-o1 generate 8 responses (Y_{GT})
- Responses from RL-trained models are highly likely to be generated by the base model

Summary for RLVR

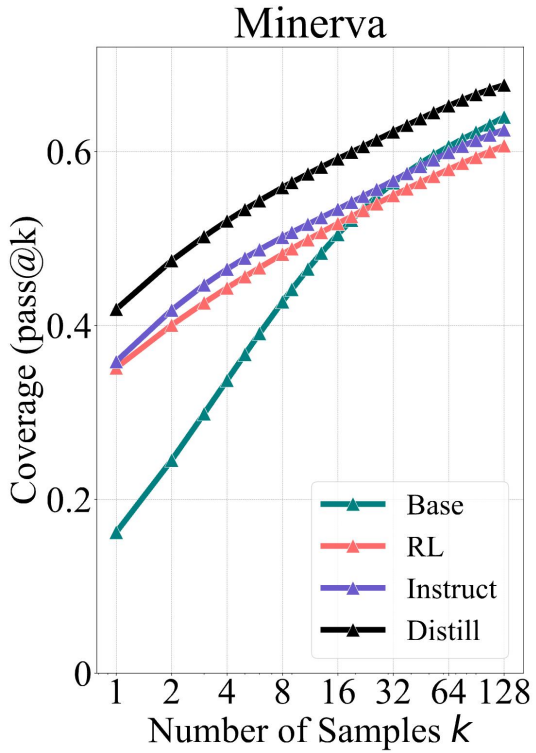
Problems solved by the RLVR model are also solvable by the base model. The observed improvement in average scores stems from **more efficient sampling** on these already solvable problems, **rather than learning to solve new problem**

After RLVR training, the model often exhibits **narrower reasoning coverage** compared to its base mode

All the reasoning paths exploited by the RLVR model are **already present** in the sampling distribution of the base model

Distillation Expands the Reasoning Boundary

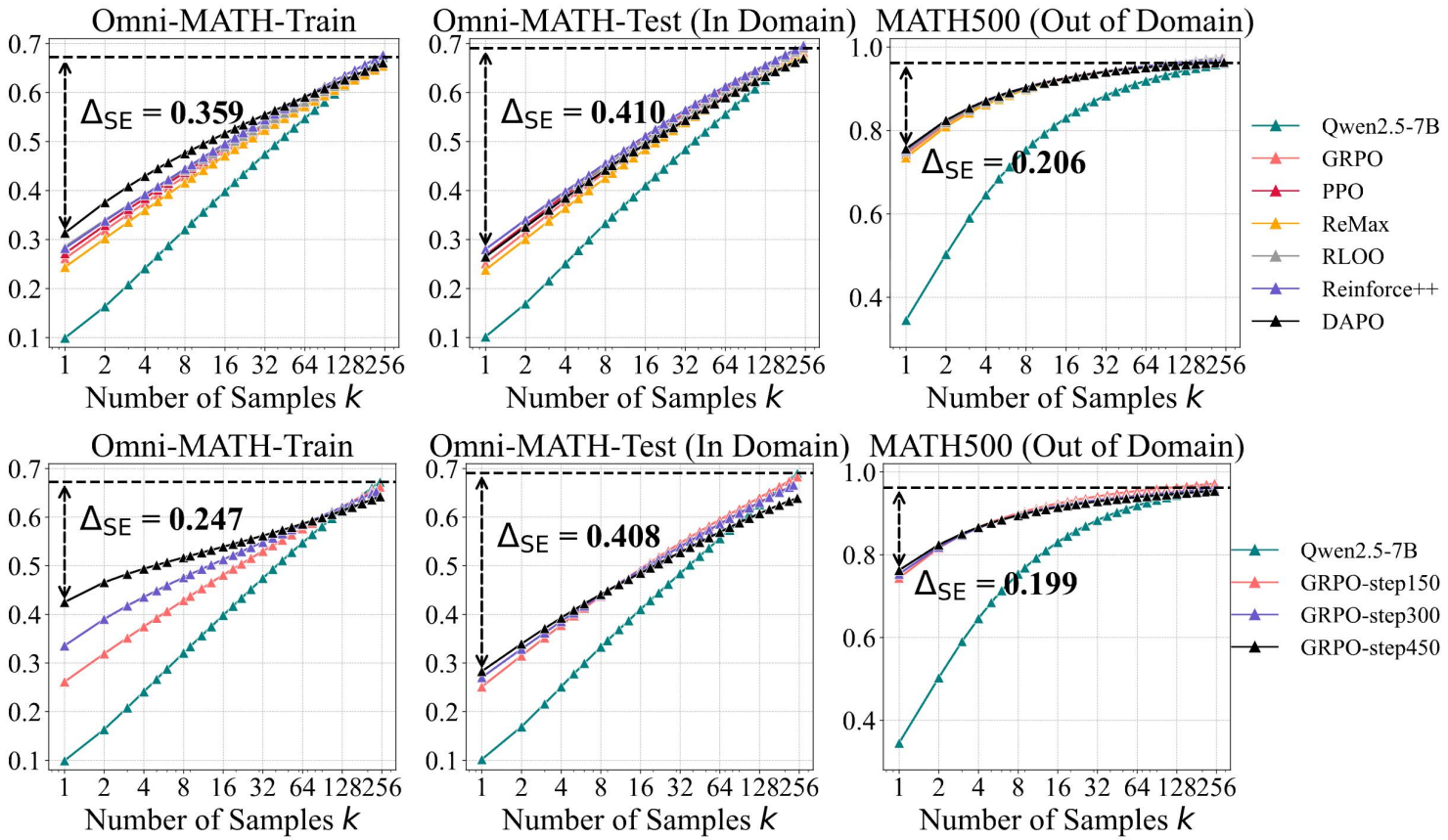
- **Distill:** DeepSeek-R1-Distill-Qwen-7B
- **Base**: Qwen2.5-Math-7B
- **RL**: Qwen2.5-Math-7B-Oat-Zero
- **Instruct**: Qwen2.5-Math-7B-Instruct



The distilled model is capable of surpassing the reasoning boundary of the base model

Effects of Different RL Algorithms

➢ Quantify sampling efficiency enhancement using *Sampling Efficiency Gap* (pass@256 of base model - pass@1 of RL model) - **Re-implement**



- Different RL algorithms yield **slightly different** sampling efficiency gap
- Sampling efficiency gap remains consistently **above 40** points across different algorithms

Figure 8: **(Top)** Different RL algorithms. **(Bottom)** Different RL training steps. The detailed values for each point at pass@1 and pass@256 are provided in Table 3 and Table 4.

Effects of RL Training

C.6. Effects of KL and Rollout Number

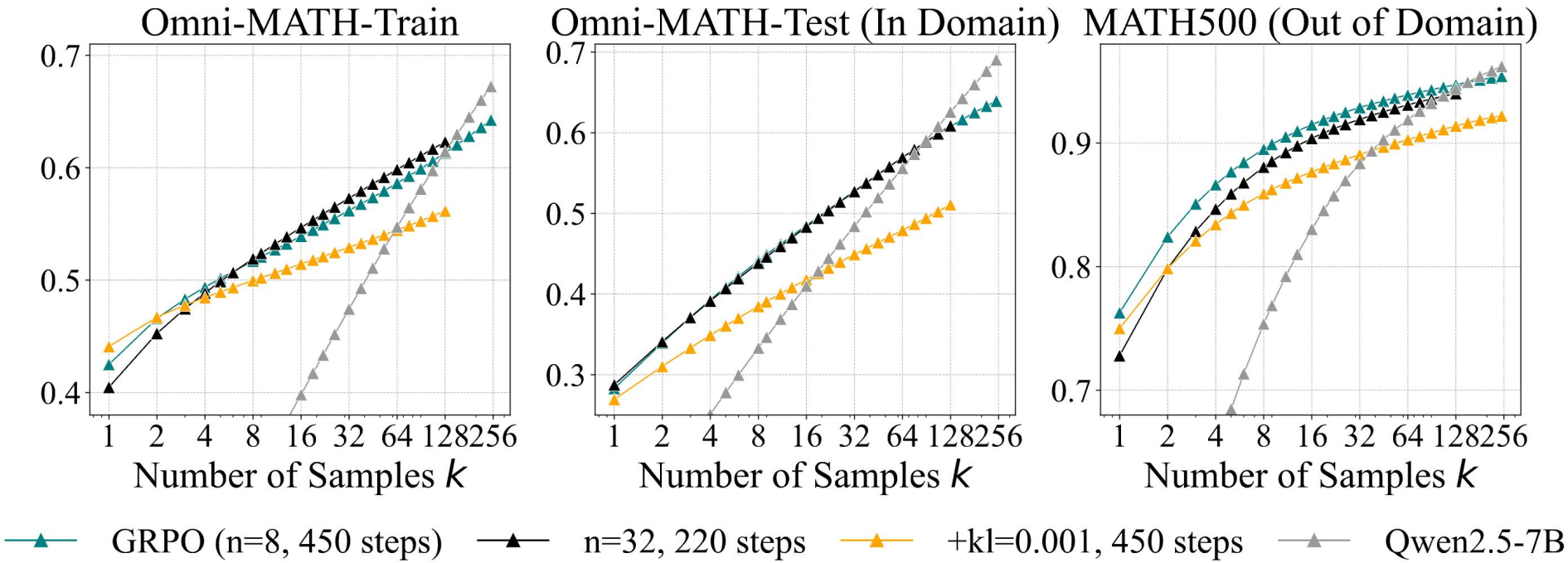


Figure 16: **Ablation Study on KL Loss and Rollout Number n.** For increasing n from 8 to 32, we keep the prompt batch size unchanged, which results in increased computation per training step. Due to resource constraints, we train for only 220 steps under this setting, leading to lower pass@1 as the model has not yet converged. Nevertheless, the model with $n = 32$ achieves a higher pass@128 highlighting the positive effect of larger rollout numbers in improving pass@k at higher values of k.

Effects of Model Size Scaling

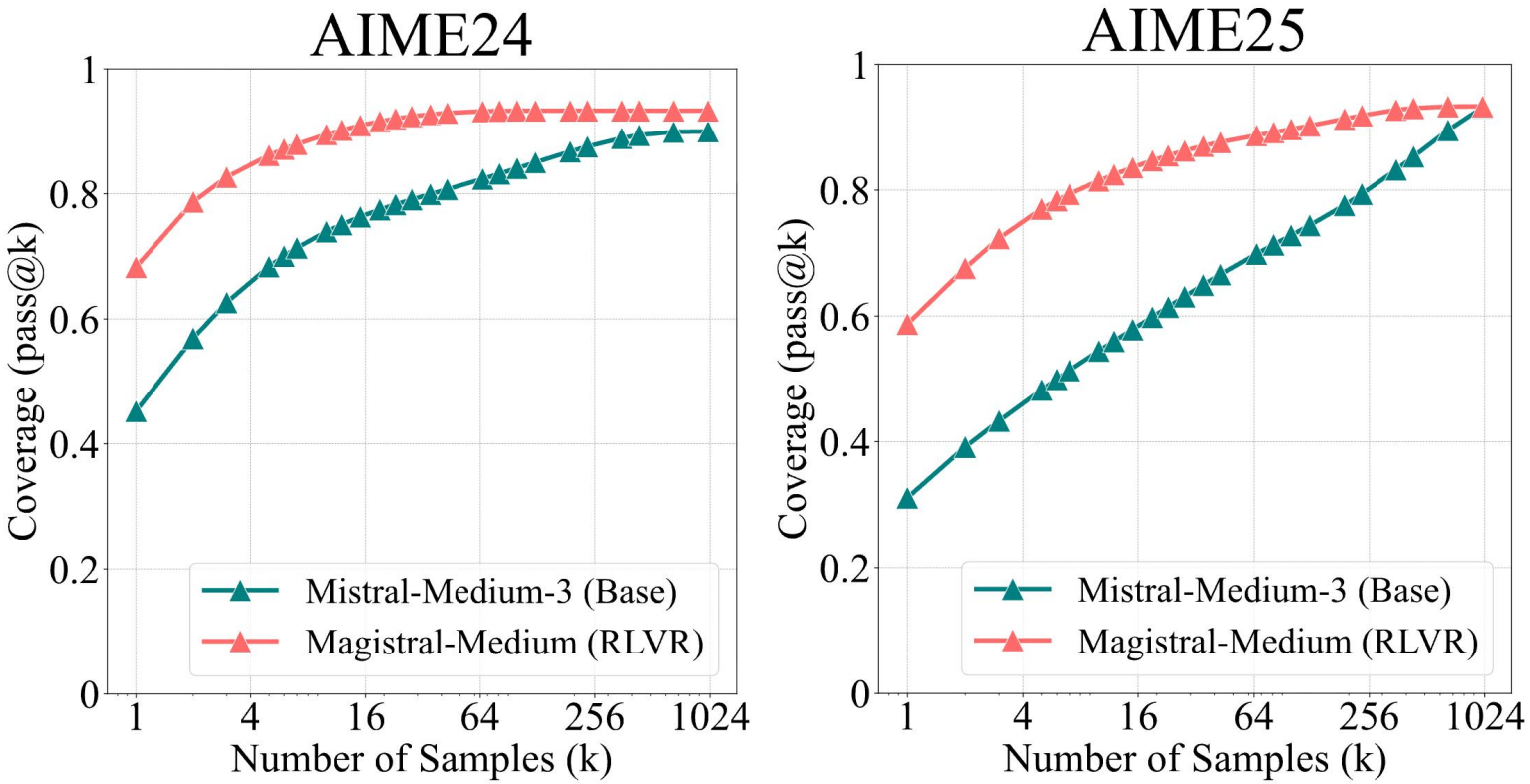


Figure 9: pass@k curves of Magistral-Medium.

RLVR provides significant gains at low k, but little or no improvement at higher k

Discussion

Discussion 1: Key Differences Between Traditional RL and RLVR for LLMs are Vast Action Space and Pretrained Priors. Traditional RL such as AlphaGo Zero and the DQN series (Silver et al., 2017; Mnih et al., 2015; Yue et al., 2023) can continuously improve the performance of a policy in environments like Go and Atari games *without an explicit upper bound*. There are two key differences between traditional RL and RLVR for LLMs. First the action space in language models is exponentially larger than that of Go or Atari games (Ramamurthy et al., 2023). RL algorithms were not originally designed to handle such a vast action space, which makes it nearly impossible to explore the reward signal effectively if training starts from scratch. Therefore, the second distinction is that RLVR for LLMs starts with a pretrained base model with useful prior, whereas traditional RL in Atari and GO games often begins from scratch. This pretrained prior guides the LLM in generating reasonable responses, making the exploration process significantly easier, and the policy can receive positive reward feedback.

Discussion 2: Priors as a Double-Edged Sword in This Vast Action Space. Since the sampling of responses is guided by the pretrained prior, the policy may struggle to explore new reasoning patterns beyond what the prior already provides. Specifically, in such a complex and highly combinatorial space, most responses generated by *naive token-level sampling exploration* are constrained by the base model's prior. Any sample deviating from the prior is highly likely to produce invalid or non-sensical outputs, leading to negative *outcome reward*. As discussed in Section 2.1, policy gradient algorithms aim to maximize the log-likelihood of responses within the prior that receive positive rewards, while minimizing the likelihood of responses outside the prior that receive negative rewards. As a result, the trained policy tends to produce responses already present in the prior, constraining its reasoning ability within the boundaries of the base model. From this perspective, training RL models from a distilled model may temporarily provide a beneficial solution, as distillation helps inject a better prior.

拓展阅读-2025/12/08

Carnegie Mellon University

Language Technologies Institute

On the Interplay of Pre-Training, Mid-Training, and RL on Reasoning Language Models

Charlie Zhang* **Graham Neubig** **Xiang Yue†**
Carnegie Mellon University, Language Technologies Institute
Interplay-LM-Reasoning Interplay-LM-Reasoning
{chariezhang0106,xiangyue.work}@gmail.com gneubig@cs.cmu.edu

小红书解读 - 强化学习RL到底能不能让大模型“变聪明”

拓展阅读-2025/12/08

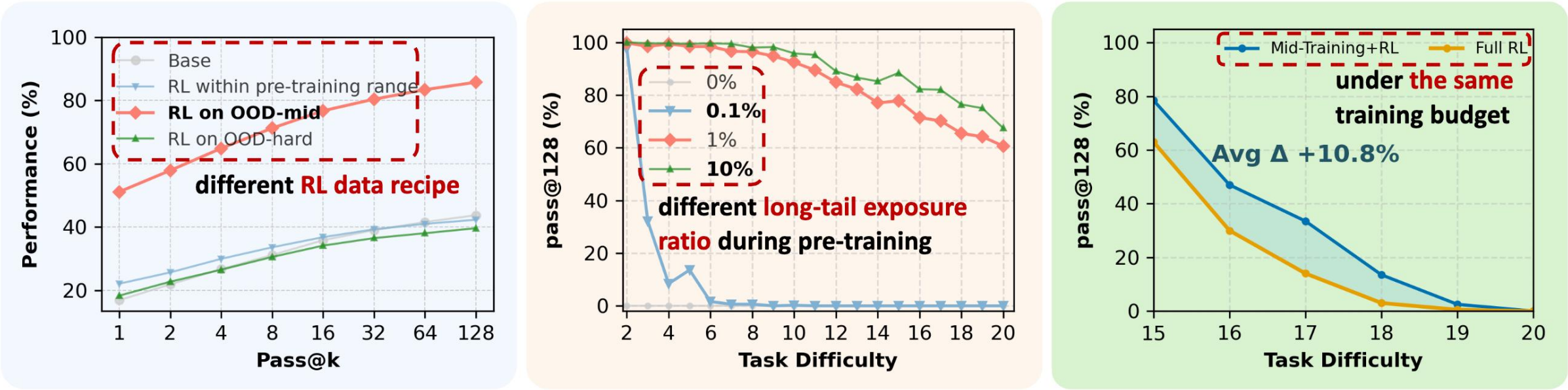


Figure 1: **Interplay of pre-, mid-, and post-training in LM reasoning. Left:** RL yields genuine extrapolative gains only when task difficulty slightly exceeds the pre-training range; gains vanish when tasks are already covered or too out-of-distribution (up to +42% pass@128 when well-calibrated). **Mid:** Contextual generalization requires minimal yet sufficient pre-training exposure to long-tail contexts. RL fails with near-zero exposure but generalizes robustly with sparse exposure ($\geq$1%), yielding up to +60% pass@128. **Right:** A mid-training stage bridging pre-training and RL substantially improves OOD reasoning under fixed compute, with mid-training + RL outperforming RL alone by +10.8% on OOD-hard tasks.

拓展阅读-2025/12/08

- 预训练中有才能被RL放大
  - 完全没出现过没法放大
  - 出现次数很少也可以被放大 (1%)

- RL能不能提升推理能力取决于难度是不是在能力边界
  - 太简单, 学不到东西
  - 太难, 学不会
  - 刚好超过能力边界, 学得很好

- Mid-training被严重低估, 能把分布对齐到更接近RL的样子, 对RL信号更敏感

  OctoThinker: Mid-training Incentivizes Reinforcement Learning Scaling

- 加入过程奖励比只用最终答案奖励更好

Thanks & QA