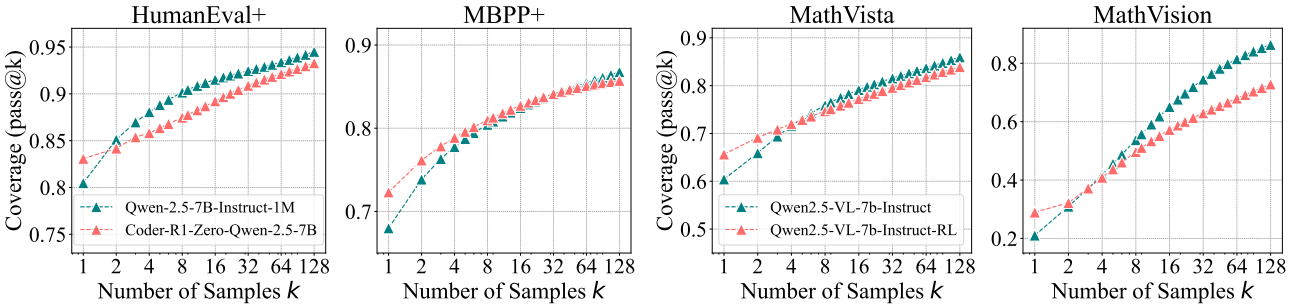

Figure 4: Pass@k curves of base and RLVR models. **(Left)** Code Generation. **(Right)** Visual Reasoning.

3.2. RLVR for Code Generation

Models and Benchmarks. We adopt the open-sourced RLVR-trained model, CodeR1-Zero-Qwen2.5-7B (Liu & Zhang, 2025), which trains zero-RL models on 12K LeetCode and TACO samples over 832 steps, based on Qwen2.5-7B-Instruct-1M (Yang et al., 2025b). For evaluation, models are assessed on LiveCodeBench v5, comprising 279 problems that span from August 2024 to January 2025 (Jain et al., 2025), as well as HumanEval+ and MBPP+ (Liu et al., 2023). We also evaluate the most powerful open-source RLVR-trained coding LLM, DeepCoder-14B (Luo et al., 2025), built on DeepSeek-R1-Distill-Qwen-14B. Here both models take 32k response length. Due to their high computational cost, we evaluate them only on LiveCodeBench as a representative benchmark.

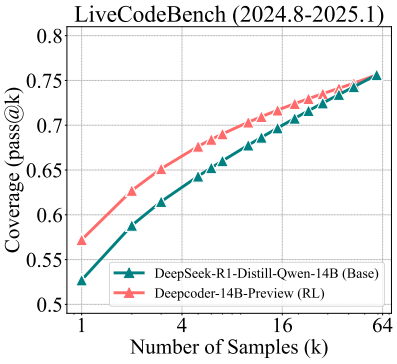

Figure 3: RLVR for Coding.

The Effect of RLVR. Since passing all unit tests is nearly impossible to achieve by guesswork, pass@k provides a reliable measure of a model's reasoning boundary. As shown in Figure 3, Figure 12, and Figure 4 (left), the effects of RLVR on three code generation benchmarks exhibit trends that are highly consistent with those observed in mathematical benchmarks.

3.3. RLVR for Visual Reasoning

Models and Benchmarks. In visual reasoning, models must jointly interpret visual and textual inputs to solve complex reasoning problems. This has gained significant attention in the multimodal community since the rise of LLM reasoning (Chen et al., 2025a; Shen et al., 2025; Zheng et al., 2025). For our experiments, we select math within visual contexts as a representative task. We use the EasyR1 framework (Zheng et al., 2025) to train Qwen2.5-VL-7B (Bai et al., 2025) on Geometry3K (Lu et al., 2021), and evaluate its visual reasoning capabilities on filtered MathVista-TestMini (Lu et al., 2024) and MathVision-TestMini (Wang et al., 2024), where multiple-choice questions are removed.

The Effect of RLVR. As shown in Figure 4 (right), the effects of RLVR on visual reasoning are highly consistent with those observed in math and coding benchmarks. This suggests that the original model has broader coverage of solvable questions even in multimodal tasks.

Validity of Chain-of-Thought. Similarly, we manually inspect a subset of the most challenging problems, *i.e.*those with an average accuracy below 5%. We find that for both the original and RL models, 7 out of 8 problems have *at least one* correct CoT. These results support the validity of CoTs.

4. Deep Analysis

In this section, we conduct a deeper analysis of the effects of current RLVR training. We also highlight the distinct characteristics of distillation in comparison to RLVR. In addition, we design controlled experiments to examine the impact of different RL algorithms and design choices.

4.1. Reasoning Paths Already Present in Base Models

Accuracy Distribution Analysis. Experiments in Section 3 reveal a surprising trend: the base model covers a wider range of solvable problems than the RLVR-trained model. To better understand this, we analyze how the accuracy distribution changes before and after RLVR training. As shown in Figure 5, RLVR increases the frequency of high accuracies near 1.0 and reduces the frequency of low accuracies (*e.g.*, 0.1, 0.2). However, a deviation from this trend is the *increased frequency at accuracy 0* — indicating that RLVR leads to more unsolvable problems. This also explains the improvement of RLVR in average scores, driven not by solving new problems but by improving sampling efficiency on problems already solvable by the base model. Additional accuracy histograms are provided in Figure 14.

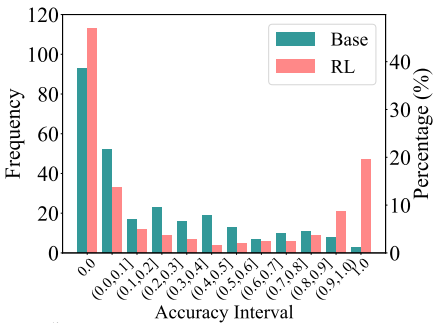

Figure 5: Qwen2.5-7B Accuracy Histogram on Minerva.

Solvable-Problem Coverage Analysis. To further investigate, we compare the set of solvable questions for both the base model and its corresponding RL-trained version on AIME24 and MATH500. We find that there are many cases where the base model solves a problem but the RLVR model fails, and very few where RLVR succeeds while the base model does not, as shown in Table 2. Details can be found at Section C.7. As shown in Table 5, the set of problems solved by the RL-trained model is nearly a subset of those solvable by the base model. A similar trend is observed in coding tasks as shown in Table 6. This raises the natural question: Do all reasoning paths generated by RL-trained models already exist within the output distribution of their base models?

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%

Perplexity Analysis. To answer this question, we utilize the metric *perplexity*. Given a model m, a problem x, and a response $\mathbf{Y} = (y_1, \ldots, y_T)$ (can be generated by the same model, another model, or humans), the perplexity is defined as the exponentiated average negative log-likelihood of a sequence:

$$\mathrm{PPL}_m(\mathbf{Y} \mid x) = \exp\left(-\frac{1}{T}\sum_{t=1}^{T}\log P(y_t \mid x, y_1, \ldots, y_{t-1})\right),$$

which reflects the model's ability to predict the given response $\mathbf{Y}$ conditioned on the prompt x. Lower perplexity indicates that the model has a higher likelihood of generating this response.

We randomly sample two problems from AIME24 and employ Qwen2.5-7B-Base and SimpleRL-Qwen2.5-7B-Base to generate 16 responses for each problem, denoted as $\mathbf{Y}_{\text{base}}$ and $\mathbf{Y}_{\text{RL}}$, respectively. We also let OpenAI-o1 (Jaech et al., 2024) generate 8 responses, denoted as $\mathbf{Y}_{\text{GT}}$. As shown in Figure 6, the distribution of $\mathrm{PPL}_{\text{Base}}(\mathbf{Y}_{\text{RL}}|x)$ closely matches the lower portion of the $\mathrm{PPL}_{\text{Base}}(\mathbf{Y}_{\text{Base}}|x)$ distribution, corresponding to responses that the base model tends to generate. This suggests that the responses from RL-trained models are highly likely to be generated by the base model. In Section C.4, we show that $\mathrm{PPL}_{\text{Base}}(\mathbf{Y}_{\text{RL}}|x)$ gradually decreases as RL training progresses, indicating that RLVR mainly sharpens the distribution within the base model's prior rather than expanding beyond it.

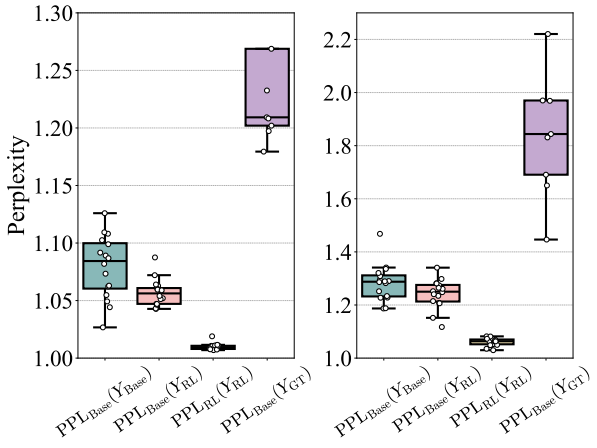

Figure 6: Perplexity distribution of responses. The conditioning problem x is omitted in the figure.

Summary. Combining the above analyses, we arrive at three key observations. First, 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 problems. Second, after RLVR training, the model often exhibits narrower reasoning coverage compared to its base model. Third, all the reasoning paths exploited by the RLVR model are already present in the sampling distribution of the base model. These findings indicate that RLVR does not introduce fundamentally new reasoning capabilities and that the reasoning capacity of the trained model remains bounded by that of its base model.

4.2. Distillation Expands the Reasoning Boundary

In addition to direct RL training, another effective approach to improving the reasoning ability of small base models is distillation from a powerful reasoning model (Guo et al., 2025). This process is analogous to instruction-following fine-tuning in post-training. However, instead of using short instruction-response pairs, the training data consist of long CoT reasoning traces generated by the teacher model. Given the limitations of current RLVR in expanding reasoning capabilities, it is natural to ask whether distillation exhibits similar behavior. We focus on a representative model, DeepSeek-R1-Distill-Qwen-7B, which distills DeepSeek-R1 into Qwen2.5-Math-7B. We compare it with the base model Qwen2.5-Math-7B and its RL-trained counterpart Qwen2.5-Math-7B-Oat-Zero and include Qwen2.5-Math-7B-Instruct as an additional baseline. As shown in Figure 7, the pass@k curve of the distilled model is consistently and significantly above that of the base model. This indicates that, unlike RL that is fundamentally bounded by the reasoning capacity of the base model, distillation introduces new reasoning patterns learned from a stronger teacher model. As a result, the distilled model is capable of surpassing the reasoning boundary of the base model.

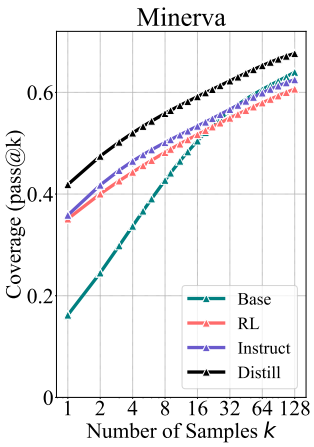

Figure 7: pass@k of base, Instruct, RLVR, and distilled models.

4.3. Effects of Different RL Algorithms

As discussed previously, the primary effect of RL is to enhance sampling efficiency rather than to expand a model's reasoning capacity. To quantify this, we propose the *Sampling Efficiency Gap* (Δ_{SE}), defined as the difference between the RL-trained model's pass@1 and the base model's pass@k (we use $k = 256$ in our evaluation). Lower Δ_{SE} is better. Here we conduct clean experiments to study the effect of different RL algorithms in enhancing sampling efficiency.

Experiment Setup. We re-implement popular RL algorithms using the VeRL framework (Sheng et al., 2024) for fair comparison, including PPO (Schulman et al., 2017), GRPO (Shao et al., 2024), Reinforce++ (Hu, 2025), RLOO (Ahmadian et al., 2024), ReMax (Li et al., 2024), and DAPO (Yu et al., 2025). Following DAPO (Yu et al., 2025) and Oat-Zero (Liu et al., 2025b), we remove the KL term to avoid constraining model learning. During training, we use the AdamW optimizer (Loshchilov & Hutter, 2017) with a constant learning rate of 10^{-6}. For rollout, we employ a prompt batch size of 256 and generate 8 responses per prompt. The maximum rollout length is set to 8,192 tokens, and the sampling temperature is set as 1.0. We use a PPO mini-batch size of 256.

To assess in-domain and out-of-domain generalization under RLVR, we split Omni-MATH-Rule, a subset of Omni-MATH (Gao et al., 2025) containing verifiable problems, into a training set (2,000 samples) and an in-domain test set (821 samples), and use MATH500 as the out-of-domain benchmark.

Results. As shown in Figure 8 (top), although different RL algorithms exhibit slight variations in both pass@1 and pass@256, these differences are not fundamental. Different RL algorithms yield slightly different Δ_{SE} values (*i.e.*, ranging from GRPO's 43.9 to RLOO's best 42.6 on the in-domain test set). Furthermore, we observe that Δ_{SE} remains consistently above 40 points across different algorithms, highlighting that existing RL methods are still far from achieving optimal sampling efficiency. This suggests that novel RL algorithms or entirely new paradigms may be necessary to approach the upper bound. Additional observations can be found at Section C.5.