RLVR vs. Distillation: Understanding Accuracy and Capability in LLM Mathematical Reasoning

Minwu Kim[*†] **Anubhav Shrestha**[*] **Safal Shrestha** **Aadim Nepal** **Keith Ross**
New York University Abu Dhabi

Abstract

Recent studies have shown that reinforcement learning with verifiable rewards (RLVR) enhances overall accuracy (pass@1) but often fails to improve capability (pass@k) in mathematical reasoning of LLMs, while distillation can improve both. In this paper, we investigate the mechanisms behind these phenomena. First, we demonstrate that RLVR struggles to improve capability because it focuses on improving the accuracy of the less-difficult questions to the detriment of the accuracy of the most difficult questions, thereby leading to no improvement in capability. Second, from the experiment distilling teacher responses to in-distribution problems, we find that capability does not always improve with distillation. We conjecture that capability improves only when new knowledge is introduced, whereas distilling reasoning patterns only improves accuracy but not capability, sacrificing performance on the most difficult questions, similar to RLVR. Together, these findings offer a clearer understanding of how RLVR and distillation shape reasoning behavior in language models. [3]

1 Introduction

Reinforcement learning with verifiable rewards (RLVR) [2, 12] and distillation of long chain-of-thought (CoT) responses [2, 15, 16, 34] are two central techniques driving recent advances in reasoning LLMs, enabling strong performance on mathematical tasks.

It is well established that RLVR improves *accuracy*—the probability of generating a correct answer, but whether it also improves *capability*—the probability that a correct answer exists in the model's output distribution—remains debated. Some studies suggest that, with sufficient compute and carefully matched training and test sets in skills and difficulty, RLVR can solve tasks that were previously unsolvable in certain domains [13, 22, 26]. Others, however, report that in more typical settings—where training and test sets contain heterogeneous problems with uncontrolled knowledge and difficulty—RLVR primarily amplifies existing reasoning rather than expanding capability [1, 29, 38, 42]. By contrast, it has been observed that distillation improves both accuracy and capability [38]. In this paper, we take a closer look at how RLVR and distillation shape mathematical reasoning in LLMs under typical settings, where training and test sets involve diverse problems with varying knowledge and difficulty.

Carrying out experiments with two models, Qwen2.5-1.5B-Math [33] and Qwen2.5-3B [10], we demonstrate that RLVR usually fails to improve capability because RLVR focuses on improving the accuracy of the less-difficult questions to the detriment of the accuracy of the hardest ones, explaining why capability may stagnate or even decline.

[*]Equal contribution.

[†]Correspondence to mwk300@nyu.edu.

[3]Code: https://github.com/minwukim/RLvsDistillation

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: MATH-AI.

We next examine distillation. A teacher model's responses convey two main elements: (1) reasoning patterns and (2) domain knowledge. To disentangle their effects, we compare three models: the base model, the publicly released DeepSeek reasoning model, which is distilled on 800k responses from DeepSeek-R1 and likely incorporates substantial new knowledge, and our own reasoning-only model, trained only on teacher responses for questions where the base model is already able to produce correct answers. We find that both distilled models yield substantial accuracy gains, but only the DeepSeek model shows clear capability improvement. These results indicate that distillation does not always expand capability, even when accuracy meaningfully improves. While further investigation is needed to confirm, we conjecture that this difference stems from whether new knowledge is introduced during distillation: introducing new knowledge may expand capability, whereas distilling only reasoning patterns improves accuracy but not capability. Interestingly, for the reasoning-only model, we also find that accuracy of the less-difficult questions improves to the detriment of the most difficult questions, mirroring the RLVR.

Taken together, our findings provide a clearer picture of different dynamics in the model behavior during RLVR training and distillation, and offer insights into strategies for enhancing the fundamental abilities of LLMs.

2 Related Work

Training reasoning models. RLVR has emerged as a key method for training LLMs to tackle complex reasoning tasks by generating long CoTs [2, 12, 17]. It has shown strong performance across model sizes [4, 7, 14, 32, 35, 40] and domains [19, 25, 31, 41]. Numerous RLVR variants have also been proposed to improve performance, data efficiency, and computational cost [3, 14, 23, 24, 27, 28, 37, 43]. Distilling high-quality CoT data is another effective approach for enhancing LLM reasoning. Such data are obtained either by prompting large models [36, 39] or by human annotation of complex reasoning traces [20, 30, 34]. A widely used strategy now involves distilling long CoT responses from RLVR-trained models into student models, often yielding substantial performance gains [9, 15, 16, 25]. Our work examines both RLVR and distillation, and evaluates how these two approaches differentially shape reasoning behavior in LLMs.

Capability expansion in RLVR. There is ongoing debate about whether RLVR develops genuinely new capabilities originally absent in a model. Several works [1, 38, 42] argue that RLVR merely amplifies correct reasoning already latent in the model. By contrast, ProRL demonstrates empirically that, given sufficient compute and diverse data, RLVR can enable models to solve previously unsolvable tasks in some domains-such as logic puzzles—suggesting the possibility of capability expansion [13]. OMEGA provides a more controlled analysis by carefully adjusting the knowledge and difficulty requirements of training and test math problems. Their results show that models can generalize to harder problems when the required knowledge is the same, but remain weak at chaining compositional skills or adopting novel strategies [26] . Similarly, e3 finds that only problems with a sufficiently large verification–generation gap benefit from test-time scaling, through experimenting under settings where the problem types of training and test sets are strictly controlled [22]. However, outside such carefully constrained conditions—in typical scenarios where both training and test sets consist of heterogeneous problems with uncontrolled knowledge and difficulty—studies consistently find that RLVR does not substantially expand capability. Theoretical analysis conducted by Wu et al. further argues that, in general, the shrinkage of empirical support outweighs its expansion in such scenarios [29]. In this work, we analyze RLVR under such general, uncontrolled math problem settings. By examining how accuracy shifts across difficulty levels, we show that RLVR tends to deliver gains on easier problems at the expense of performance on harder ones.

Reasoning pattern and knowledge in distillation. Several studies have examined the respective roles of domain knowledge and reasoning patterns in improving accuracy through distillation. For instance, Shrestha et al. distill teacher responses from logic puzzles—where domain knowledge is minimal—and show that transferring reasoning patterns alone can yield substantial performance gains across domains such as mathematics and coding [25]. Likewise, Huan et al. demonstrate that distilling ath problems responses produces notable improvements in other domains [8]. However, work on capability remains limited. Yue et al. suggested that distillation can drive capability expansion, but their analysis does not disentangle the effects of reasoning patterns and knowledge injection [38]. In contrast, our study explicitly controls for this distinction and investigates how each factor differentially influences model capability.

3 Why Doesn't RLVR Improve Capability?

Prior work has shown across multiple models that RLVR yields substantial gains in accuracy but often fails to improve capability, as measured by pass@k with sufficiently large k [38]. In this section, we extend this observation and analyze the phenomenon in greater depth. Specifically, we aim to answer: *Why does RLVR raise accuracy while leaving capability unchanged or even degraded?*

3.1 Capability Analysis

We first replicated the pass@k experiments of Yue et al., confirming that RLVR increases accuracy but not capability (Fig. 5). Our evaluation covered 2 base models (Qwen2.5-1.5B-Math [33] and Qwen2.5-3B[10]) along with their RLVR-trained counterparts. Training was conducted on the MATH train set, with evaluation on the MATH 500 test set [5]; these datasets are used throughout this section. Training details and full pass@k results are provided in Appendix A.10 and A.5, respectively.

We report here results on the 1.5B model evaluated on the test set due to space constraints. However, the same pattern holds consistently across both the train and test sets, and across both model sizes (1.5B and 3B). Full results are provided in Appendix A.6. For clarity, we refer to the original 1.5B model as the base model and the RLVR-trained version as the RL model.

3.2 A Deeper-Dive: Analysis Based on Question Difficulty

To better understand the accuracy–capability dynamics of RLVR, we conduct a fine-grained analysis at the question difficulty level. For each question in the training and test sets, we generate 256 responses from the base model and compute its per-question success rate. Questions are then grouped into bins according to these rates: [0], [1–4], [5–16], [17–64], [65–128], and [129–256]. Within each bin, we collect the corresponding questions, retrieve the RL model's responses to the same questions, and compute average success rates for both models. We then calculate the average success rate of the base and RL models in each bin and plot their absolute difference (Fig. 1 (left)).

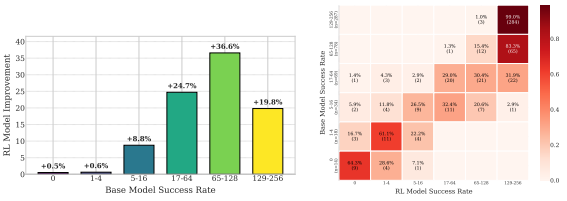


Figure 1: Analysis of success rate changes and transitions after RLVR training on MATH 500 test set. Left: absolute difference in success rates across bins. Right: success rate transition matrix.

We find two clear patterns. (1) When the base model already achieves a moderate success rate, the RL model yields large gains. For example, the [65–128] bin improves by an absolute 36.6 percentage points, and the [17–64] bin by 24.7 points. (2) When the base model has zero or near-zero success, the gains are negligible: the [0] bin increases by only 0.5 points and the [1–4] bin by 0.6.

To further understand the pattern observed in success rate improvements, we visualize how individual questions move across success rate bins before and after RLVR training. Figure 1 (right) presents this transition in test set as a matrix. Here, each row corresponds to a success-rate bin based on the base model's performance, and each column corresponds to the same bin based on the RL model's performance. Each cell shows the percentage (and count) of questions that started in a specific base model bin and ended up in a particular RL model bin after training.

Notably, we observe two clear trends. (1) Questions already in high-success bins tend to stay there or shift upward after RLVR. For example, in the [65–128] bin, 15.4% remain in place while 83.3% move to the top [129–256] bin; only 1.3% (1 question out of 78) drop lower. A similar upward shift appears in the [17–64] bin. (2) In contrast, questions in low-success bins—especially those near zero—tend to stagnate or regress. In the [1–4] bin, 61.1% remain and 16.7% fall to [0]; likewise, in the [5–16] bin, 44.2% stay or drop lower. This pattern shows a clearer picture of how RLVR fails to help previously unsolved questions and can even increase their number, as many with a small chance of being answered correctly end up never being solved after training.

To understand this behavior, we can consider the internal dynamics of RLVR training using GRPO as an example. With only a limited number of generations per question, difficult questions often fail to produce a single correct answer, resulting in no parameter update. Consequently, updates are driven

almost entirely by relatively easier questions. Over time, this imbalance shifts training toward easier cases, while harder ones remain unsolved or even become less likely to be solved. To test this, we increase the number of generations and observe that the sacrifice-of-hard-problems issue is slightly alleviated (see Appendix A.7). Nevertheless, because difficult questions still rarely contribute to updates, the problem remains an inherent limitation of RLVR.

To summarize, these results suggest the following insight: RLVR improves accuracy but not capability as *RLVR focuses on improving the accuracy of the less-difficult questions to the detriment of the accuracy of the most difficult questions.*

4 Under What Conditions Does Distillation Increase Capability?

Distillation from teacher reasoning models is another effective approach for improving accuracy [15, 16, 34]. Here, we ask: *Can distillation also improve capability, and under what conditions?*

4.1 Capability Analysis

[38] briefly explored this issue by comparing two models: Qwen-2.5-Math-7B [33] and DeepSeek-R1-Distill-Qwen-7B [2]. The latter is a publicly available model obtained by distilling 800K DeepSeek-R1 responses into the Qwen-2.5-Math-7B student model. Their experiments showed that the distilled model demonstrates improved capability, as evidenced by higher pass@k scores.

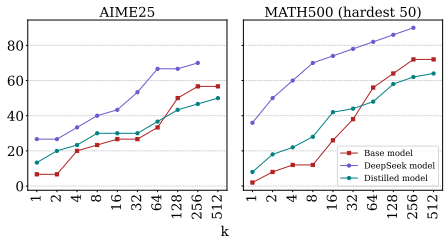


Figure 2: Pass@k results for AIME25 and MATH500 (hardest 50 questions under base model), comparing the 1.5B model with its distillation-trained variants.

However, the source of this improvement remains uncertain. Teacher responses contain two key elements: (1) the model's *reasoning patterns*, and (2) its *domain knowledge*. In the case of DeepSeek-R1-Distill-Qwen-7B, the large volume of teacher responses almost certainly injected new mathematical knowledge that was absent from the student model's pre-training data. This makes it unclear whether the observed capability gains stem from adopting more effective reasoning pattern or from learning new knowledge. To disentangle these effects, we design a comparative study across three models: (1) Qwen2.5-Math-1.5B, a non-reasoning base model; (2) DeepSeek-R1-Distill-Qwen-1.5B, which, like its 7B counterpart, was trained on 800K responses from DeepSeek-R1 [2] and likely benefits from both new knowledge and improved reasoning; (3) our own distilled model, designed to isolate the effect of reasoning-pattern transfer without introducing any new domain knowledge beyond that of base model. For simplicity, we refer to these as the base model, the DeepSeek model, and the reasoning-only model.

We train our reasoning-only model as follows. From the base model, we sample 256 responses for each of the 7,500 MATH training questions and discard those with zero success rate. The remaining questions, each answered correctly at least once, are treated as *in-distribution*. As the teacher, we use QwQ-32B [21], which we confirm has higher capability than the student (see Appendix A.8). For each in-distribution question, QwQ-32B generates 8 candidate responses; we randomly select up to 4 correct ones (using all if fewer exist) for supervised fine-tuning of the base model. Because all questions are ones the base model already solves, this procedure avoids introducing new knowledge. The distilled model reaches 70% accuracy on MATH 500, outperforming the base model's 60%, indicating a successful distillation.

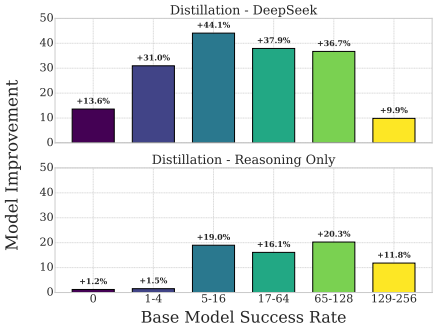


Figure 3: Absolute success-rate differences of DeepSeek and reasoning-only models, grouped by base-model bins.

We assess the three models' capability by conducting pass@k experiments on AIME25 [18] and MATH500 (Fig. 2). Two consistent patterns emerge. (1) The DeepSeek model outperforms the base model across all k. On AIME25, it reaches 70.0% at pass@256 versus 56.7% for the base model. To ensure this gap does not close at higher k, we extend evaluation of the base model up

to pass@512 and confirm that its performance plateaus, remaining at 56.7%. A similar trend holds for MATH500, showing that DeepSeek improves both accuracy and capability, consistent with [38]. (2) The reasoning-only model outperforms the base model at small k, but the curves converge and even cross as k increases. Replicating the experiment with Qwen2.5-3B yields the same pattern (Appendix A.9). These results show that the reasoning-only model improves accuracy but does not expand capability, similar to the case of RLVR discussed in Section 3.

These results show that distillation does not always lead to capability expansion, even when it yields significant accuracy gains. We conjecture that the difference arises from whether new knowledge is introduced during distillation. Specifically, *Distillation may improve capability when it introduces new knowledge, whereas learning reasoning patterns alone boosts accuracy but not capability.*

4.2 In-Depth Analysis Based on Question Difficulty

To better understand the dynamics of the two distillation settings, we perform a bin-based analysis similar to Section 3.2, comparing model performance across question bins grouped by base model success rates, based on 256 sampled responses. As shown in Figure 3, two contrasting patterns emerge: (1) The DeepSeek model shows substantial improvement across all bins, including those with zero or near-zero success rates. (2) In contrast, the reasoning-only model shows improvement primarily in bins with moderately high success rates, but little gain in the zero or near-zero bins, mirroring the behavior of RLVR discussed in Section 3.

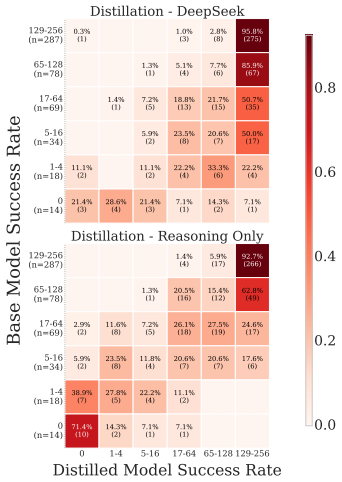


Figure 4: Success rate transition matrix in two distillation settings on MATH 500 test set.

As in Section 3.2, We further examine per-question transitions before and after distillation (Figure 4). We observe: (1) For the DeepSeek model, questions consistently move to higher success-rate bins, even those that started in low-success bins. For instance, in the [1–4] bin, only 11.1% (2 out of 18) drop to the [0] bin, and in the [5–16] bin, no question moves downward. (2) In contrast, for the reasoning-only model, we interestingly observe the same "sacrificing difficult problems" effect seen in RLVR. In the [1–4] bin, 38.9% (7 out of 18) drop to the [0] bin, and in the [5–16] bin, 29.4% (10 out of 34) move to lower bins.

We again conjecture that the key factor underlying this difference is whether new knowledge is introduced during distillation. Specifically, distillation with new knowledge improves both accuracy and capability because *it enables the model to solve questions across all difficulty levels, including the most difficult ones*. In contrast, reasoning-only distillation improves accuracy but not capability because, like RLVR, *it focuses on easier questions—often at the cost of performance on the hardest ones*. We hope this result motivates further empirical study to validate this conjecture and clarify the role of new knowledge in capability expansion.

5 Conclusions

Recent work has shown that RLVR improves accuracy but not capability of LLMs in mathematical reasoning tasks, while distillation from a strong teacher often improves both. In this paper, we conduct extensive experiments to understand these dynamics in greater depth. Our contributions can be summarized as follows: (1) We explain why RLVR improves accuracy but not capability by showing that it disproportionately favors easier questions to the detriment of harder ones—often degrading performance for difficult questions. (2) While distillation consistently improves accuracy, its effect on capability is less clear. We conjecture that capability improves only when new knowledge is introduced, whereas distilling reasoning patterns only improves accuracy but not capability, sacrificing performance on the most difficult questions, similar to RLVR. Taken together, our findings provide a clearer picture of different dynamics in the model behavior during RLVR training and distillation, and offer insights into strategies for enhancing the fundamental abilities of LLMs.

References

[1] Xingyu Dang, Christina Baek, J. Zico Kolter, and Aditi Raghunathan. Assessing diversity collapse in reasoning. In *OpenReview (ICLR Workshop / Supplementary Track)*, 2025. Available at https://openreview.net/forum?id=AMiKsHLjQh.

[2] DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL https://arxiv.org/abs/2501.12948. arXiv:2501.12948.

[3] Mehdi Fatemi, Banafsheh Rafiee, Mingjie Tang, and Kartik Talamadupula. Concise reasoning via reinforcement learning. *arXiv preprint arXiv:2504.05185*, 2025. URL https://arxiv.org/abs/2504.05185.

[4] 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, 2025. URL https://arxiv.org/abs/2503.01307. arXiv:2503.01307.

[5] 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. In *Advances in Neural Information Processing Systems*, volume 34, pages 24241–24253, 2021. URL https://arxiv.org/abs/2103.03874.

[6] Andreas Hochlehnert, Hardik Bhatnagar, Vishaal Udandarao, Samuel Albanie, Ameya Prabhu, and Matthias Bethge. A sober look at progress in language model reasoning: Pitfalls and paths to reproducibility, 2025. URL https://arxiv.org/abs/2504.07086.

[7] Jingcheng Hu, Yinmin Zhang, Qi Han, Daxin Jiang, Xiangyu Zhang, and Heung-Yeung Shum. Open-reasoner-zero: An open source approach to scaling up reinforcement learning on the base model. *arXiv preprint arXiv:2503.24290*, 2025. URL https://arxiv.org/abs/2503.24290.

[8] Maggie Huan, Yuetai Li, Tuney Zheng, Xiaoyu Xu, Seungone Kim, Minxin Du, Radha Poovendran, Graham Neubig, and Xiang Yue. Does math reasoning improve general llm capabilities? understanding transferability of llm reasoning. *arXiv preprint arXiv:2507.00432*, 2025. Also available at https://arxiv.org/abs/2507.00432.

[9] Zhen Huang, Haoyang Zou, Xuefeng Li, Yixiu Liu, Yuxiang Zheng, Ethan Chern, Shijie Xia, Yiwei Qin, Weizhe Yuan, and Pengfei Liu. O1 replication journey–part 2: Surpassing o1-preview through simple distillation, big progress or bitter lesson? *arXiv preprint arXiv:2411.16489*, 2024. URL https://arxiv.org/abs/2411.16489.

[10] Binyuan Hui, Jian Yang, Zeyu Cui, Jiaxi Yang, Dayiheng Liu, Lei Zhang, Tianyu Liu, Jiajun Zhang, Bowen Yu, Kai Dang, An Yang, Rui Men, Fei Huang, Xingzhang Ren, Xuancheng Ren, Jingren Zhou, and Junyang Lin. Qwen2.5 technical report, 2024. URL https://arxiv.org/abs/2412.15115.

[11] 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. *arXiv preprint arXiv:2309.06180*, 2023. URL https://arxiv.org/abs/2309.06180.

[12] 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. Tulu 3: Pushing frontiers in open language model post-training, 2024. URL https://arxiv.org/abs/2411.15124. arXiv:2411.15124.

[13] Mingjie Liu, Shizhe Diao, Ximing Lu, Jian Hu, Xin Dong, Yejin Choi, Jan Kautz, and Yi Dong. Prorl: Prolonged reinforcement learning expands reasoning boundaries in large language models. *arXiv preprint arXiv:2505.24864*, 2025.

[14] 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.

[15] Yingqian Min, Zhipeng Chen, Jinhao Jiang, Jie Chen, Jia Deng, Yiwen Hu, Yiru Tang, Jiapeng Wang, Xiaoxue Cheng, Huatong Song, Wayne Xin Zhao, Zheng Liu, Zhongyuan Wang, and Ji-Rong Wen. Imitate, explore, and self-improve: A reproduction report on slow-thinking reasoning systems, 2024. URL https://arxiv.org/abs/2412.09413.

[16] Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. s1: Simple test-time scaling, 2025. URL https://arxiv.org/abs/2501.19393. arXiv:2501.19393.

[17] OpenAI. Openai o1 system card, 2024. URL https://arxiv.org/abs/2412.16720. arXiv:2412.16720.

[18] OpenCompass. Aime2025 dataset, 2025. URL https://huggingface.co/datasets/opencompass/AIME2025.

[19] Jiayi Pan, Junjie Zhang, Xingyao Wang, Lifan Yuan, Hao Peng, and Alane Suhr. Tinyzero: A minimal reproduction of reasoning models, 2025. Available at https://github.com/Jiayi-Pan/TinyZero.

[20] Yiwei Qin, Xuefeng Li, Haoyang Zou, Yixiu Liu, Shijie Xia, Zhen Huang, Yixin Ye, Weizhe Yuan, Hector Liu, Yuanzhi Li, and Pengfei Liu. O1 replication journey: A strategic progress report – part 1. *arXiv preprint arXiv:2410.18982*, 2024. URL https://arxiv.org/abs/2410.18982.

[21] Qwen. Qwq-32b preview: Reflect deeply on the boundaries of the unknown. https://qwenlm.github.io/blog/qwq-32b-preview/, 2024. Accessed: 2025-05-05.

[22] Amrith Setlur, Matthew Y. R. Yang, Charlie Snell, Jeremy Greer, Ian Wu, Virginia Smith, Max Simchowitz, and Aviral Kumar. e3: Learning to explore enables extrapolation of test-time compute for llms. *arXiv preprint arXiv:2506.09026*, 2025.

[23] Rulin Shao, Shuyue Stella Li, Rui Xin, Scott Geng, Yiping Wang, Sewoong Oh, Simon Shaolei Du, Nathan Lambert, Sewon Min, Ranjay Krishna, Yulia Tsvetkov, Hannaneh Hajishirzi, Pang Wei Koh, and Luke Zettlemoyer. Spurious rewards: Rethinking training signals in rlvr. *arXiv preprint arXiv:2506.10947*, 2025.

[24] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024. URL https://arxiv.org/abs/2402.03300.

[25] Safal Shrestha, Minwu Kim, Aadim Nepal, Anubhav Shrestha, and Keith Ross. Warm up before you train: Unlocking general reasoning in resource-constrained settings. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2025.

[26] Yiyou Sun, Shawn Hu, Georgia Zhou, Ken Zheng, Hannaneh Hajishirzi, Nouha Dziri, and Dawn Song. Omega: Can llms reason outside the box in math? evaluating exploratory, compositional, and transformative generalization. *arXiv preprint arXiv:2506.18880*, 2025.

[27] Yiping Wang, Qing Yang, Zhiyuan Zeng, Liliang Ren, Lucas Liu, Baolin Peng, Hao Cheng, Xuehai He, Kuan Wang, Jianfeng Gao, Weizhu Chen, Shuohang Wang, Simon Shaolei Du, and Yelong Shen. Reinforcement learning for reasoning in large language models with one training example. *arXiv preprint arXiv:2504.20571*, 2025. URL https://arxiv.org/abs/2504.20571.

[28] Yiping Wang, Qing Yang, Zhiyuan Zeng, Liliang Ren, Lucas Liu, Baolin Peng, Hao Cheng, Xuehai He, Kuan Wang, Jianfeng Gao, Weizhu Chen, Shuohang Wang, Simon Shaolei Du, and Yelong Shen. Reinforcement learning for reasoning in large language models with one training example. In *Proceedings of the 38th International Conference on Neural Information Processing Systems (NeurIPS)*, 2025.

[29] Fang Wu, Weihao Xuan, Ximing Lu, Zaid Harchaoui, and Yejin Choi. The invisible leash: Why rlvr may not escape its origin. *arXiv preprint arXiv:2507.14843*, 2025.

[30] Violet Xiang, Charlie Snell, Kanishk Gandhi, Alon Albalak, Anikait Singh, Chase Blagden, Duy Phung, Rafael Rafailov, Nathan Lile, Dakota Mahan, Louis Castricato, Jan-Philipp Franken, Nick Haber, and Chelsea Finn. Towards system 2 reasoning in llms: Learning how to think with meta chain-of-thought, 2025. URL https://arxiv.org/abs/2501.04682.

[31] Tian Xie, Zitian Gao, Qingnan Ren, Haoming Luo, Yuqian Hong, Bryan Dai, Joey Zhou, Kai Qiu, Zhirong Wu, and Chong Luo. Logic-rl: Unleashing llm reasoning with rule-based reinforcement learning. *arXiv preprint arXiv:2502.14768*, 2025. URL https://arxiv.org/abs/2502.14768.

[32] Haoran Xu, Baolin Peng, Hany Awadalla, Dongdong Chen, Yen-Chun Chen, Mei Gao, Young Jin Kim, Yunsheng Li, Liliang Ren, Yelong Shen, et al. Phi-4-mini-reasoning: Exploring the limits of small reasoning language models in math. *arXiv preprint arXiv:2504.21233*, 2025. URL https://arxiv.org/abs/2504.21233.

[33] An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, Keming Lu, Mingfeng Xue, Runji Lin, Tianyu Liu, Xingzhang Ren, and Zhenru Zhang. Qwen2.5-math technical report: Toward mathematical expert model via self-improvement, 2024.

[34] Yixin Ye, Zhen Huang, Yang Xiao, Ethan Chern, Shijie Xia, and Pengfei Liu. Limo: Less is more for reasoning, 2025. URL https://arxiv.org/abs/2502.03387.

[35] Edward Yeo, Yuxuan Tong, Morry Niu, Graham Neubig, and Xiang Yue. Demystifying long chain-of-thought reasoning in llms. *arXiv preprint arXiv:2502.03373*, 2025. URL https://arxiv.org/abs/2502.03373.

[36] Ping Yu, Jing Xu, Jason Weston, and Ilia Kulikov. Distilling system 2 into system 1. *arXiv preprint arXiv:2407.06023*, 2024. URL https://arxiv.org/abs/2407.06023.

[37] Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Weinan Dai, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Mu Qiao, Yonghui Wu, and Mingxuan Wang. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025. URL https://arxiv.org/abs/2503.14476.

[38] Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model?, 2025. URL https://arxiv.org/abs/2504.13837.

[39] Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah D. Goodman. Star: Bootstrapping reasoning with reasoning. *arXiv preprint arXiv:2203.14465*, 2022. URL https://arxiv.org/abs/2203.14465.

[40] Weihao Zeng et al. Simplerl-zoo: Investigating and taming zero reinforcement learning for open base models in the wild, 2025. URL https://arxiv.org/abs/2503.18892. arXiv:2503.18892.

[41] Sheng Zhang, Qianchu Liu, Guanghui Qin, Tristan Naumann, and Hoifung Poon. Med-rlvr: Emerging medical reasoning from a 3b base model via reinforcement learning. *arXiv preprint arXiv:2502.19655*, 2025. URL https://arxiv.org/abs/2502.19655.

[42] Rosie Zhao, Alexandru Meterez, Sham Kakade, Cengiz Pehlevan, Samy Jelassi, and Eran Malach. Echo chamber: Rl post-training amplifies behaviors learned in pretraining. In *Proceedings of COLM 2025*, 2025. Also available as arXiv preprint arXiv:2504.07912.

[43] 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. URL https://arxiv.org/abs/2504.16084.

A Appendix

A.1 Acknowledgement

We gratefully acknowledge the support of the Center for AI and Robotics (CAIR) at New York University Abu Dhabi for this research.

A.2 Limitations

While our study provides an in-depth analysis of RLVR and distillation, it also has limitations that suggest directions for future work.

First, due to resource constraints, our experiments are restricted to a single domain, mathematics, and different patterns may emerge in other tasks. There remains an ongoing debate about whether RLVR truly improves capability. As discussed in Section 2, some studies argue that RLVR does not enhance capability in general mathematical settings where both training and test sets contain heterogeneous problems with uncontrolled knowledge and difficulty. Others, however, show that RLVR can indeed expand capability when sufficient training compute is available and when training and test sets are carefully controlled in terms of problem type and difficulty. Therefore, follow-up work is needed to unify these perspectives and develop a more comprehensive understanding of the phenomenon.

Second, our experiments are limited to relatively small models (1.5B and 3B) and a single RL algorithm family (GRPO & Dr. GRPO). Larger models or different RL algorithms may exhibit different dynamics. A more comprehensive study across model scales and training methods is needed to test the generality of our findings.

Third, our distillation experiments are limited in both scale and control. The DeepSeek model used for comparison is distilled on approximately 800k teacher responses and trained from a different base model, whereas our reasoning-only distillation relies on roughly 30k responses from the same model. In addition, when we extend the setup to include teacher responses to out-of-distribution (OOD) questions, we do not observe measurable capability improvement, possibly due to the small number of OOD samples or limited coverage of new knowledge. Consequently, we cannot conclusively determine whether capability gains depend on the introduction of new knowledge. Future work should validate this conjecture under more controlled settings.

A.3 Accuracy and Capability

A.3.1 Formal Definitions

We evaluate models along two dimensions: *accuracy* and *capability*. Informally, accuracy measures how likely a model is to generate a correct answer in a single attempt, while capability measures whether a correct answer exists within the distribution of responses the model can generate.

Formally, we define accuracy and capability with respect to given model M and evaluation dataset $\mathcal{D} = \{1, \ldots, N\}$ of N questions. Let p_i^M denote the probability that model M successfully solves question i in a single attempt. Note that this can be obtained by sampling model M for n times on question i, computing the fraction of correct responses, and taking the limit as $n \to \infty$. In theory, an LLM using softmax sampling assigns non-zero probability to all valid outputs, so any answer could eventually be produced. To make capability practically meaningful, we consider a question i to be *in-distribution* for model M if $p_i^M > \epsilon$, where ϵ is a small threshold (typically 10^{-2} to 10^{-3}).

To evaluate performance under multiple attempts, let $p_{i,k}^M$ denote the probability that model M solves question i at least once across k independent attempts. This probability satisfies

$$p_{i,k}^M = 1 - (1 - p_i^M)^k.$$

With these definitions in place, we define the model's *accuracy* as the average success rate over the entire dataset:

$$\mathrm{Acc}(M) = \frac{1}{N} \sum_{i \in \mathcal{D}} p_i^M.$$

We define the model's *pass@k capability* as the average success probability over $\mathcal{D}$ given k passes per question:

$$\mathrm{Cap}_k(M) = \frac{1}{N}\sum_{i\in\mathcal{D}} p_{i,k}^M = \frac{1}{N}\sum_{i\in\mathcal{D}} \left(1-(1-p_i^M)^k\right)$$

It is important to note that if model M' has higher accuracy than model M ($p_i^{M'} > p_i^M$) for a specific question, then it will also have higher pass@k capability for that question. However, this relationship does not always hold taking into account the entire dataset. In fact, as shown in Appendix A.4, it is possible for $\mathrm{Acc}(M') > \mathrm{Acc}(M)$ while $\mathrm{Cap}_k(M') < \mathrm{Cap}_k(M)$.

A.3.2 Estimating Accuracy and Capability

In practice, it is infeasible to compute the exact accuracy and capability of a model, as this would require a prohibitively large number of samples per question. Instead, we estimate these quantities empirically using a finite number of samples k. Let $X_{i,k}$ be the number of correct responses among k samples for question i.

We estimate *accuracy* as:

$$\mathrm{Acc}(M) \approx \frac{1}{N}\sum_{i\in\mathcal{D}} \frac{X_{i,k}}{k}$$

We estimate *pass@k capability* as:

$$\mathrm{Cap}_k(M) \approx \frac{1}{N}\sum_{i\in\mathcal{D}} 1(X_{i,k} > 0)$$

These estimators are unbiased. Throughout this work, we report results using these estimators, typically with $k = 256$. We also consider a question i to be out-of-distribution if $X_{i,256} = 0$, that is, none of the 256 responses to question i are correct. Under this definition, we can say with 95% confidence that $p_i^M < 1-(0.05)^{1/256} \approx 0.012$, that is, question i is truly out-of-distribution under the threshold $\epsilon = 0.012$.

A.4 Accuracy vs. Capability Example

As discussed in A.3, we provide an example to illustrate that a model can have higher *accuracy* but lower *capability* on an evaluation dataset with more than one question.

Recall the definitions:

$$\mathrm{Acc}(M) = \frac{1}{N}\sum_{i=1}^{N} p_i^M,$$

$$\mathrm{Cap}_k(M) = \frac{1}{N}\sum_{i=1}^{N} \left(1-(1-p_i^M)^k\right).$$

We compare two models, M_1 and M_2, on a toy dataset of $N = 3$ questions. Their single-attempt success probabilities p_i^M are shown below:

Question	$p_i^{M_1}$	$p_i^{M_2}$
1	0.9	0.5
2	0.9	0.5
3	0.003	0.5

Table 1: Single-pass success probabilities for models M_1 and M_2.

We first compute the accuracy of two models on this toy dataset.

$$\mathrm{Acc}(M_1) = \frac{1}{3}(0.9+0.9+0.003) = 0.601,$$

$$\mathrm{Acc}(M_2) = \frac{1}{3}(0.5+0.5+0.5) = 0.5.$$

Thus, M_1 has higher accuracy.

We now compute capability with $k = 256$, which is large enough to expose the low success probability on Question 3 for M_1:

Using the formula:

$$p_{i,k}^{M} = 1 - (1 - p_i^{M})^k,$$

we compute:

Model M_1:

$$\begin{aligned} p_{1,256}^{M_1} &= 1 - (1 - 0.9)^{256} \approx 1, \\ p_{2,256}^{M_1} &= 1 - (1 - 0.9)^{256} \approx 1, \\ p_{3,256}^{M_1} &= 1 - (1 - 0.003)^{256} \approx 0.537. \\ \mathrm{Cap}_{256}(M_1) &= \frac{1}{3}(1 + 1 + 0.537) \approx 0.845. \end{aligned}$$

Model M_2:

$$\begin{aligned} p_{i,256}^{M_2} &= 1 - (1 - 0.5)^{256} = 1 - 2^{-256} \approx 1 \quad \text{for all } i, \\ \mathrm{Cap}_{256}(M_2) &= \frac{1}{3}(1 + 1 + 1) = 1.0. \end{aligned}$$

As shown, although M_1 has significantly higher probabilities to the first two questions—resulting in higher overall accuracy—its probability on the third question is extremely low. As a result, even with many sampling attempts, M_1 is unlikely to solve all questions. In contrast, M_2 maintains moderate but consistent success probabilities across all three questions, which leads to a higher chance of solving every question at least once when given sufficient attempts.

A.5 Pass@k Experiments Results Before & After RLVR

In this paper, we used two models to evaluate the effect of RLVR training: Qwen2.5-1.5B-Math, and Qwen2.5-3B. For corresponding RL model of 1.5B model, we used the Qwen2.5-Math-1.5B-Oat-Zero, a publicly available model trained with MATH train dataset by Liu et al.. For Qwen2.5-3B, we conducted the RLVR training ourselves, also with MATH train dataset. Further details for training can be found at Appendix A.10 and A.11, respectively.

Split	Model	Qwen2.5-1.5B-Math			Qwen2.5-3B		
		Accuracy	Maj@256	Pass@256	Accuracy	Maj@256	Pass@256
Train	Base	64.0%	76.8%	97.2%	59.3%	80.9%	92.7%
	RL	80.9%	82.1%	97.1%	67.9%	82.2%	92.1%
	Difference	+16.9%	+5.3%	-0.1%	+8.6%	+1.3%	-0.6%
Test	Base	60.6%	72.0%	97.2%	54.9%	76.5%	95.8%
	RL	74.2%	80.8%	97.0%	63.6%	79.5%	95.8%
	Difference	+13.9%	+8.8%	-0.2%	+8.7%	+3.0%	+0.0%

Table 2: Performance comparison of base and RL models for Qwen2.5-1.5B-Math and Qwen2.5-3B

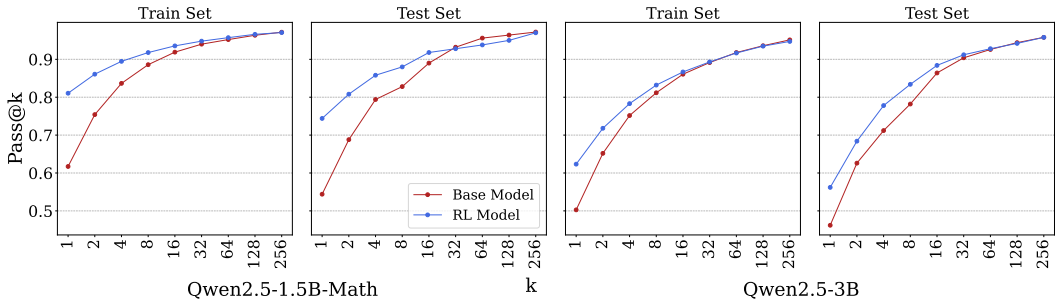


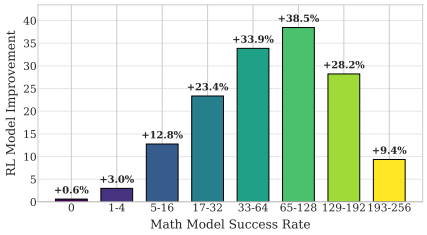
Figure 5: Pass@k comparison between base and RLVR-trained models on train and test sets.

Similar to the work done by Yue et al., we conducted the pass@k experiments with these models. For both the base and RL models, we generated 256 responses per question on the MATH train set and MATH500 test set. Using these responses, we estimated accuracy and pass@k capability for $k = 1$ to 256, following the metric defined in Section A.3.2. Additionally, we computed majority vote accuracy (maj@256), which is the percentage of questions where the most frequent answer among the 256 responses is correct.

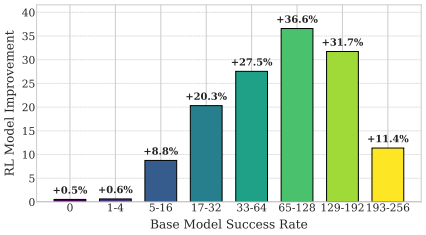
As expected, we observed that RLVR significantly improved both accuracy and majority vote performance across training and test sets. As shown in Table 2, these gains appeared consistently in both the 3B and 1.5B models, indicating that RLVR leads to generalizable improvement in accuracy without signs of overfitting. In contrast, we observed no meaningful improvement in capability. For both the 1.5B and the 3B models, pass@k either remained stable or slightly declined across the training and test sets. As shown in Figure 5, the RL model outperformed the base model at small k, but their curves converged as k increases—a pattern consistent with prior work [24, 38].

A.6 Question-Difficulty-Based Analysis Results

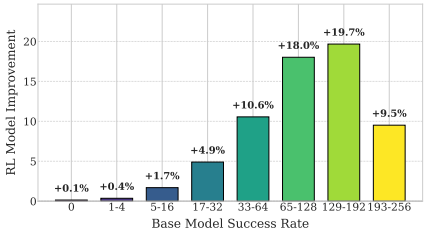
As discussed in Sections 3, we performed detailed analyses based on question difficulty across different training settings. The results are presented below. Figure 6 shows success rate improvements across difficulty bins for both 1.5B and 3B models on train and test. Figure 7 presents the corresponding transition matrices that illustrate how questions move between success rate bins before and after training.



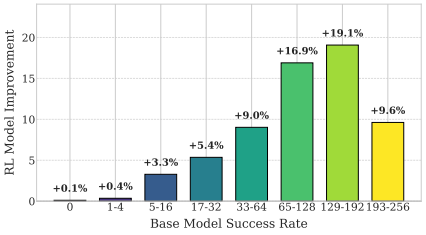
(a) Qwen2.5-1.5B-Math (Train)



(b) Qwen2.5-1.5B-Math (Test)

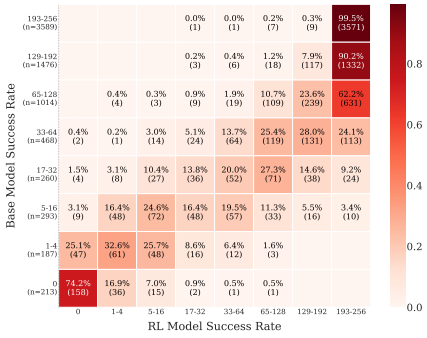


(c) Qwen2.5-3B (Train)

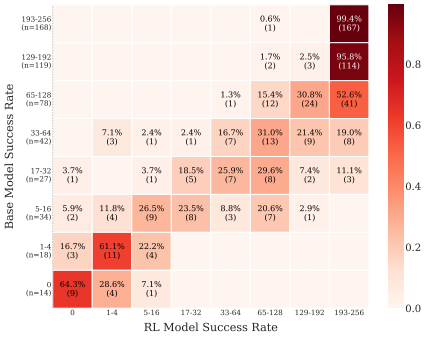


(d) Qwen2.5-3B (Test)

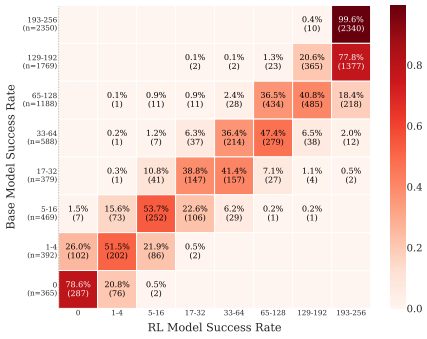
Figure 6: Change in success rates (absolute %) across difficulty bins for Qwen2.5-1.5B-Math and Qwen2.5-3B on the MATH training and test sets. In both models, RLVR significantly improves questions in the mid-success bins (e.g., [17–64], [65–128]), but yields minimal gains in the lowest bins ([0], [1–4]).



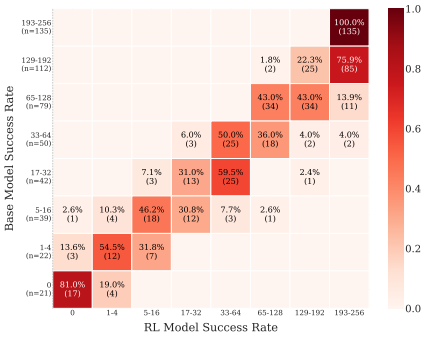
(a) Qwen2.5-1.5B-Math (Train)



(b) Qwen2.5-1.5B-Math (Test)



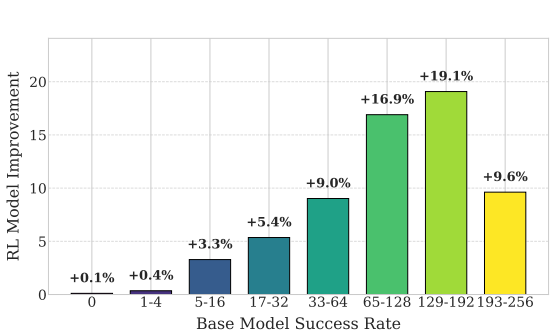
(c) Qwen2.5-3B (Train)



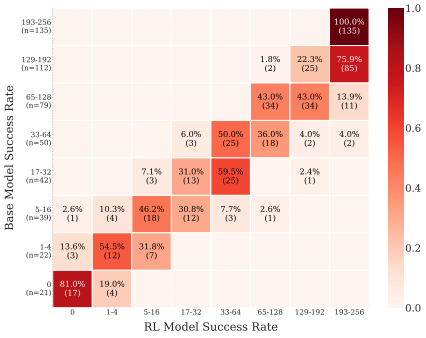
(d) Qwen2.5-3B (Test)

Figure 7: Transition matrices comparing base and RLVR success-rate bins for Qwen2.5-1.5B-Math and Qwen2.5-3B. Each cell shows the percentage and count of questions moving between success bins. Most upward transitions occur from mid-success bins; questions in low-success bins are more likely to remain unchanged or regress.

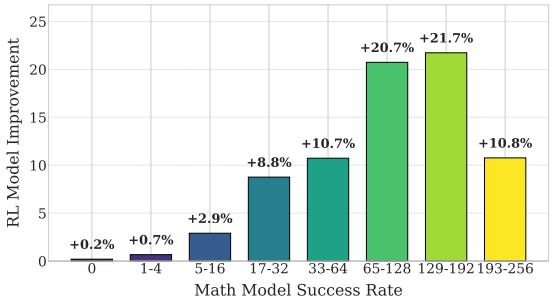
A.7 RLVR with different numbers of generations per question



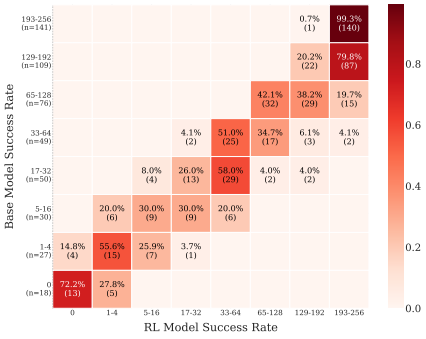
(a) 10 generations: success-rate changes across bins



(b) 10 generations: transition matrix



(c) 24 generations: success-rate changes across bins



(d) 24 generations: transition matrix

Figure 8: Comparison of success-rate changes and transition matrices on MATH 500 for Qwen2.5-3B RLVR models trained with 10 vs. 24 generations per question.

In Section 3, we discussed how increasing the number of generations per question in GRPO training can mitigate the sacrifice-of-hard-problems issue. To test this, we trained two Qwen2.5-3B models under the same conditions: one with 10 generations per question (as used in the main experiments) and another with 24. As shown in Figure 8, the model trained with 24 generations exhibited less regression and more improvement on hard questions compared to the model trained with 10.

A.8 QwQ-32B Capability Experiment

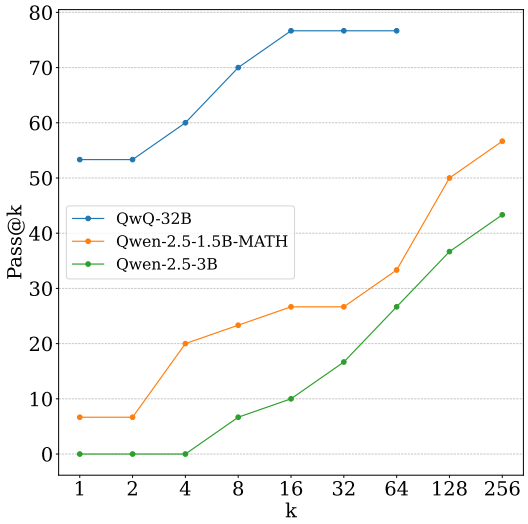


Figure 9: Pass@k results of QwQ-32B, Qwen2.5-3B-Math, and Qwen2.5-1.5B-Math on AIME 25

In Section 4, we selected QwQ-32B as the teacher model for our reasoning-only distillation experiment. To ensure a fair test of whether distillation can improve capability without introducing new knowledge, the teacher must have higher capability than the student models—Qwen2.5-3B and Qwen2.5-1.5B-Math.

To validate this, we conducted a pass@k evaluation on AIME 25 using 64 responses per question from QwQ-32B, and compared the results with the two student models. As shown in Figure 9, QwQ-32B consistently outperforms both students across all k values, with no sign of convergence. Notably, its pass@64 score reached 76.7%, compared to just 43.3% and 56.7% at pass@256 for Qwen2.5-3B and Qwen2.5-1.5B-Math, respectively. These results confirm that QwQ-32B has substantially higher capability, making it a suitable teacher model for our distillation setup.

A.9 Distillation Pass@k results

As discussed in Section 4, we conducted the pass@k on AIME 25 and MATH 500 for both Qwen2.5-1.5B-Math and Qwen2.5-3B and each their 2 distilled variants. The results are shown below in Figure 10.

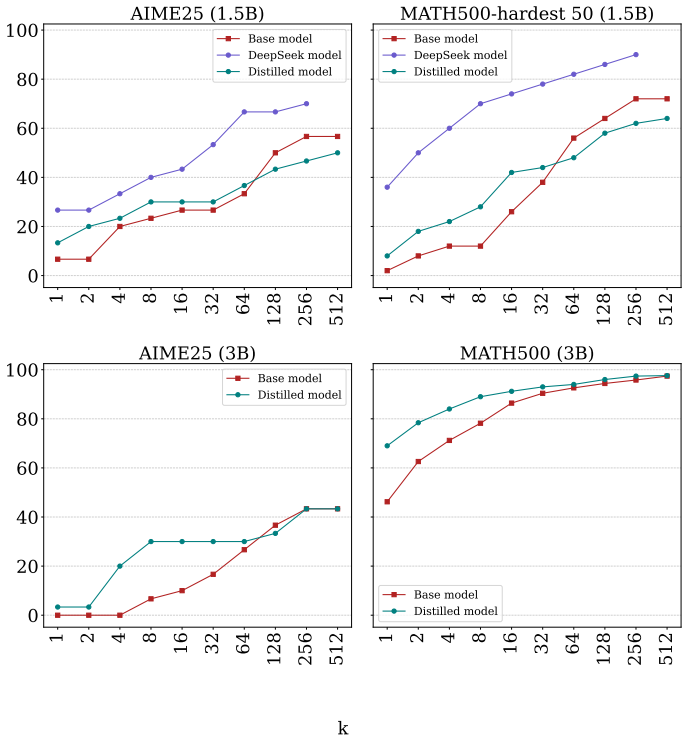


Figure 10: Pass@k comparisons across AIME25 and MATH500 datasets for both 1.5B (top) and 3B (bottom) models and their distillation-trained variants. For the MATH500 results of the 1.5B models, we show performance on the 50 questions with the lowest base-model success rates to better highlight the differences.

A.10 Qwen2.5-Math-1.5B Training Details

In this paper, we used two models as base models: Qwen2.5-1.5B-Math and Qwen2.5-3B. For the RLVR-trained version of the 1.5B model, we used Qwen2.5-Math-1.5B-Oat-Zero[4], a publicly available model trained by Liu et al.. According to their report, the model was trained with Dr.GRPO [14], a variant of the GRPO algorithm [24] designed to remove response length and question difficulty biases. The model was trained on questions from level 3 to 5 from the MATH training set. For the 3B model, we performed RLVR training ourselves. Training details are shown right below in Appendix A.11

A.11 Qwen2.5-3B RLVR Training Details

For RLVR training of Qwen2.5-3B, we used the `GRPOTrainer` from the `TRL`[5] library, which implements the standard GRPO algorithm. The model was trained on the full MATH training set, consisting of 7,500 questions.

A.11.1 Prompt Setting

Prior work has shown that the performance of smaller models can be sensitive to prompt design [6, 14]. Following Liu et al., we evaluated three prompt formats, as listed below. We ultimately adopted Template 3 (question only), which yielded the best performance.

Prompt Templates

Template 1 (R1 template) A conversation between User and Assistant. The User asks a question, and the Assistant solves it. The Assistant first thinks about the reasoning process in the mind and then provides the User with the answer. The reasoning process is enclosed within `<think> </think>` and the answer is enclosed within `<answer> </answer>` tags. `User: {question} Assistant: <think> reasoning here </think> <answer> answer here </answer>`

Template 2 (Qwen-Math template) `<|im start|>system` Please reason step by step, and put your final answer within \boxed{}. `<|im end|> <|im start|>user` {question} `<|im end|> <|im start|>assistant`

Template 3 (Question only) {question}

A.11.2 Reward Function

We adopted a minimalistic reward setting. A response received a reward of 1 if it contained the correct final answer, and -1 otherwise. Answer verification was performed using the `math_verify`[6] package.

$$R(q, a, r) = \begin{cases} 1 & \text{if the response } r \text{ to question } q \text{ matches the ground truth answer } a \\ -1 & \text{otherwise} \end{cases}$$

A.11.3 RLVR Training Hyperparameters

Table 3 summarizes the key hyperparameters used in RLVR training for the Qwen2.5-3B model.

[4] `https://huggingface.co/sail/Qwen2.5-Math-1.5B-Oat-Zero`
[5] `https://github.com/huggingface/trl`
[6] `https://github.com/huggingface/Math-Verify`

Hyperparameter	Value
Optimizer	AdamW
Learning rate scheduler	Constant
Maximum token length	4000
Temperature	0.9
Top-p	1.0
Top-k	50
Number of generations (per question)	10
Global batch size	4 (per device) $\times$ 7 (GPUs) $\times$ 10 (accumulation) = 280
Learning rate	1×10^{-6}
Gradient clipping (max grad norm)	0.1
Number of gradient steps	225
Warmup steps	20
Mixed precision	bf16

Table 3: Key hyperparameters used for RLVR training of Qwen2.5-3B.

A.11.4 Training Progress and Evaluation

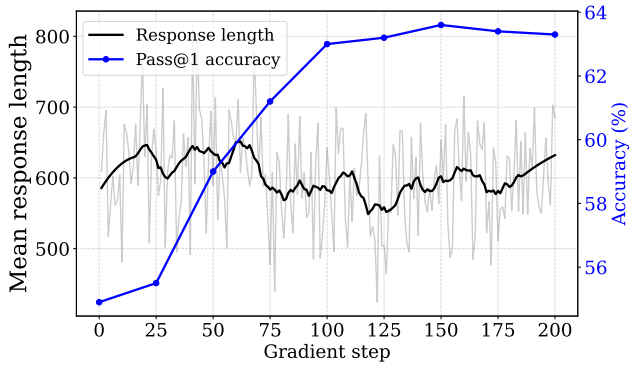


Figure 11: Change in response length (token counts) over training and accuracy on MATH 500 across checkpoints.

During RLVR training, we evaluated the model every 25 gradient steps. To ensure statistical robustness, we followed the recommendation of Hochlehnert et al. [6] and sampled responses 10 times per checkpoint, reporting the mean accuracy. For each evaluation, we used a temperature of 0.8 and a top-p of 0.9.

As shown in Figure 11, accuracy peaked at step 150, reaching 63.6%, and then plateaued. We selected this checkpoint as the RL model used throughout our experiments. The figure also shows the average response length over training.

A.12 Distillation Training Hyperparameters

For all the distillation experiments in Section 4, we used the supervised fine-tuning (SFT) hyperparameters listed in Table 4.

Hyperparameter	Value
Optimizer	AdamW
Learning rate scheduler	Constant
Weight decay	1×10^{-4}
Warmup steps	25
Max sequence length	32,768
Global batch size	4
Mixed precision	bf16

Table 4: Key hyperparameters used for supervised fine-tuning in distillation experiments.

A.13 Response Generation Details

We used vLLM[7] library [11] for response generation and `math_verify`[8] package for response grading.

We used temperature 0.9, top-p of 1.0, and top-k of 50 for all models, except where noted below. These settings were chosen to ensure response diversity. Unless otherwise specified, we used the question-only template (Template 3).

For Qwen2.5-Math-1.5B-Oat-Zero, we used the same sampling hyperparameters but followed the Qwen prompt format (Template 2), as recommended in the user guideline.[9]

For QwQ-32B, we used temperature 0.6, top-p 0.95, and top-k 50. We followed the R1 prompt template (Template 1), as recommended in the user guideline.[10]

For DeepSeek-R1-Distill-Qwen-1.5B, we used temperature 0.6, top-p 0.95, and top-k 50. We followed the R1 prompt template (Template 1), as recommended in the user guideline.[11]

[7] https://docs.vllm.ai

[8] https://github.com/huggingface/Math-Verify

[9] https://huggingface.co/sail/Qwen2.5-Math-1.5B-Oat-Zero

[10] https://huggingface.co/Qwen/QwQ-32B#usage-guidelines

[11] https://huggingface.co/deepseek-ai/DeepSeek-R1-Distill-Qwen-1.5B#usage-recommendations