demonstrate the effectiveness of the distillation technique, leaving the exploration of the RL stage to the broader research community. For details on distillation training, please see Appendix B.4.3.

Table 15 | Comparison of DeepSeek-R1 distilled models and other comparable models on reasoning-related benchmarks. Numbers in bold denote the performance is statistically significant (t−test with $p < 0.01$).

Model	AIME 2024		MATH	GPQA Diamond	LiveCode Bench	CodeForces
	pass@1	cons@64	pass@1	pass@1	pass@1	rating
GPT-4o-0513	9.3	13.4	74.6	49.9	32.9	759
Claude-3.5-Sonnet-1022	16.0	26.7	78.3	65.0	38.9	717
DeepSeek-R1-Distill-Qwen-1.5B	28.9	52.7	83.9	33.8	16.9	954
DeepSeek-R1-Distill-Qwen-7B	55.5	83.3	92.8	49.1	37.6	1189
DeepSeek-R1-Distill-Qwen-14B	69.7	80.0	93.9	59.1	53.1	1481
DeepSeek-R1-Distill-Qwen-32B	**72.6**	83.3	94.3	62.1	57.2	1691
DeepSeek-R1-Distill-Llama-8B	50.4	80.0	89.1	49.0	39.6	1205
DeepSeek-R1-Distill-Llama-70B	70.0	**86.7**	**94.5**	**65.2**	**57.5**	**1633**

We evaluate the distilled models on AIME, GPQA, Codeforces, as well as MATH-500 (Lightman et al., 2024) and LiveCodeBench (Jain et al., 2024). For comparison, we use two well-established LLMs as baselines: GPT-4o and Claude-3.5-Sonnet. As shown in Table 15, the straightforward distillation of outputs from DeepSeek-R1 allows the distilled model, DeepSeek-R1-Distill-Qwen-1.5B, to surpass non-reasoning baselines on mathematical benchmarks. Notably, it is remarkable that a model with only 1.5 billion parameters achieves superior performance compared to the best closed-source models. Furthermore, model performance improves progressively as the parameter size of the student model increases.

Our experimental results demonstrate that smaller models can achieve strong performance through distillation. Furthermore, as shown in Appendix F, the distillation approach yields superior performance compared to reinforcement learning alone when applied to smaller model architectures. This finding has significant implications for democratizing AI access, as reduced computational requirements enable broader societal benefits.

F.1. Distillation v.s. Reinforcement Learning

Table 16 | Comparison of distilled and RL Models on Reasoning-Related Benchmarks.

Model	AIME 2024		MATH	GPQA Diamond	LiveCode Bench
	pass@1	cons@64	pass@1	pass@1	pass@1
QwQ-32B-Preview	50.0	60.0	90.6	54.5	41.9
Qwen2.5-32B-Zero	47.0	60.0	91.6	55.0	40.2
DeepSeek-R1-Distill-Qwen-32B	**72.6**	**83.3**	**94.3**	**62.1**	**57.2**

In Section F, we can see that by distilling DeepSeek-R1, the small model can achieve impressive results. However, there is still one question left: can the model achieve comparable

Table 17 | Performance of different models on AIME 2024 and AIME 2025.

Average Score	**AIME 2024**	**AIME 2025**
GPT-4o-0513	9.3%	-
Qwen2-Math-7B-Instruct	7.9%	4.6%
Qwen2-Math-7B-Zero	22.3%	18.1%

performance through the large-scale RL training discussed in the paper without distillation?

To answer this question, we conduct large-scale RL training on Qwen2.5-32B-Base using math, code, and STEM data, training for over 10K steps, resulting in Qwen2.5-32B-Zero, as described in B.4.1. The experimental results, shown in Table 16, demonstrate that the 32B base model, after large-scale RL training, achieves performance on par with QwQ-32B-Preview. However, DeepSeek-R1-Distill-Qwen-32B, which is distilled from DeepSeek-R1, performs significantly better than Qwen2.5-32B-Zero across all benchmarks.

Therefore, we can draw two conclusions: First, distilling more powerful models into smaller ones yields excellent results, whereas smaller models relying on the large-scale RL mentioned in this paper require enormous computational power and may not even achieve the performance of distillation. Second, while distillation strategies are both economical and effective, advancing beyond the boundaries of human intelligence may still require more powerful base models and larger-scale reinforcement learning.

Apart from the experiment based on Qwen-2.5-32B, we conducted experiments on Qwen2-Math-7B (released August 2024) prior to the launch of the first reasoning model, OpenAI-o1 (September 2024), to ensure the base model was not exposed to any reasoning trajectory data. We trained Qwen2-Math-7B-Zero with approximately 10,000 policy gradient update steps. As shown in Table 17, Qwen2-Math-7B-Zero significantly outperformed the non-reasoning models like Qwen2-Math-7B-Instruct and GPT-4o. These results further demonstrate that the model can autonomously develop advanced reasoning strategies through large-scale reinforcement learning.

G. Discussion

G.1. Key Findings

We highlight our key findings, which may facilitate the community in better reproducing our work.

The importance of base checkpoint: During the initial phase of our development, we experimented with smaller-scale models, specifically a 7B dense model and a 16B Mixture-of-Experts (MoE) model, as the foundational architectures for RL training. However, these configurations consistently failed to yield meaningful improvements when evaluated on the AIME benchmark, which we employed as the primary validation set. We observed that as response lengths increased, these smaller models exhibited a tendency toward repetition and were unable to effectively leverage long chains of thought (CoT) to improve reasoning accuracy.

To address these limitations, we transitioned to larger-scale models, including a 32B dense model (Qwen, 2024b), a 230B MoE model (DeepSeek-AI, 2024a), and a 671B MoE model (DeepSeek-AI, 2024b). With these more capable architectures, we finally observed substantial

performance gains attributable to pure RL training. These findings suggest that the effectiveness of reinforcement learning from base models is highly dependent on the underlying model capacity. We therefore recommend that future research in this area prioritize the use of sufficiently large and expressive models when aiming to validate the efficacy of RL from scratch.

The importance of verifiers: The effectiveness of DeepSeek-R1-Zero is highly contingent upon the reliability and fidelity of the reward signal used during training. To date, our investigations indicate that two approaches—rule-based reward models (RMs) and LLMs to assess an answer's correctness against a predefined ground-truth—serve as robust mechanisms for mitigating issues related to reward hacking. The LLM-based evaluation framework demonstrates particular effectiveness for tasks with well-defined, concise answers, such as single-sentence or phrase-level responses. However, this method exhibits limited generalizability to more complex tasks, including open-ended generation and long-form writing, where the notion of correctness is inherently more subjective and nuanced.

Iterative pipeline: We propose a multi-stage training pipeline comprising both SFT and RL stages. The RL component enables the model to explore and discover optimal reasoning trajectories for tasks capabilities that cannot be fully realized through human-annotated reasoning traces alone. In particular, without the RL stage, long-chain reasoning patterns, such as those required in complex Chain-of-Thought (CoT) prompting, would remain largely unexplored. Conversely, the SFT stage plays a crucial role in tasks where reliable reward signals are difficult to define or model, such as open-ended question answering and creative writing. Therefore, both RL and SFT are indispensable components of our training pipeline. Exclusive reliance on RL can lead to reward hacking and suboptimal behavior in ill-posed tasks, while depending solely on SFT may prevent the model from optimizing its reasoning capabilities through exploration.

G.2. Unsuccessful Attempts

In the early stages of developing DeepSeek-R1, we also encountered failures and setbacks along the way. We share our failure experiences here to provide insights, but this does not imply that these approaches are incapable of developing effective reasoning models.

Process Reward Model (PRM) PRM is a reasonable method to guide the model toward better approaches for solving reasoning tasks (Lightman et al., 2024; Uesato et al., 2022; Wang et al., 2023a). However, in practice, PRM has three main limitations that may hinder its ultimate success. First, it is challenging to explicitly define a fine-grain step in general reasoning. Second, determining whether the current intermediate step is correct is a challenging task. Automated annotation using models may not yield satisfactory results, while manual annotation is not conducive to scaling up. Third, once a model-based PRM is introduced, it inevitably leads to reward hacking (Gao et al., 2022), and retraining the reward model needs additional training resources and it complicates the whole training pipeline. In conclusion, while PRM demonstrates a good ability to rerank the top-N responses generated by the model or assist in guided search (Snell et al., 2024), its advantages are limited compared to the additional computational overhead it introduces during the large-scale reinforcement learning process in our experiments.

Monte Carlo Tree Search (MCTS) Inspired by AlphaGo (Silver et al., 2017b) and AlphaZero (Silver et al., 2017a), we explored using Monte Carlo Tree Search (MCTS) to enhance test-time compute scalability. This approach involves breaking answers into smaller parts to allow the model to explore the solution space systematically. To facilitate this, we prompt the model to