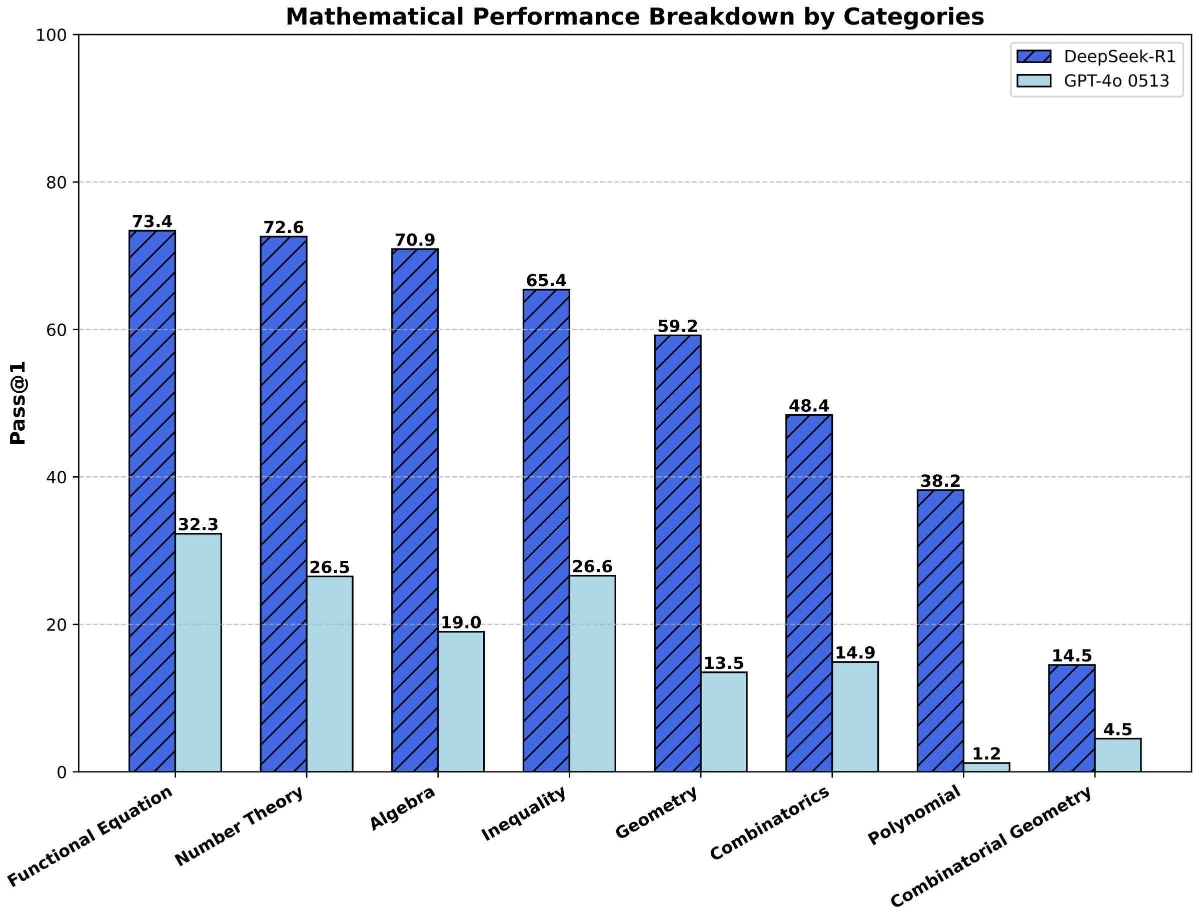

Figure 17 | Performance breakdown by different categories of quantitative reasoning problems from a collection of contests in 2024.

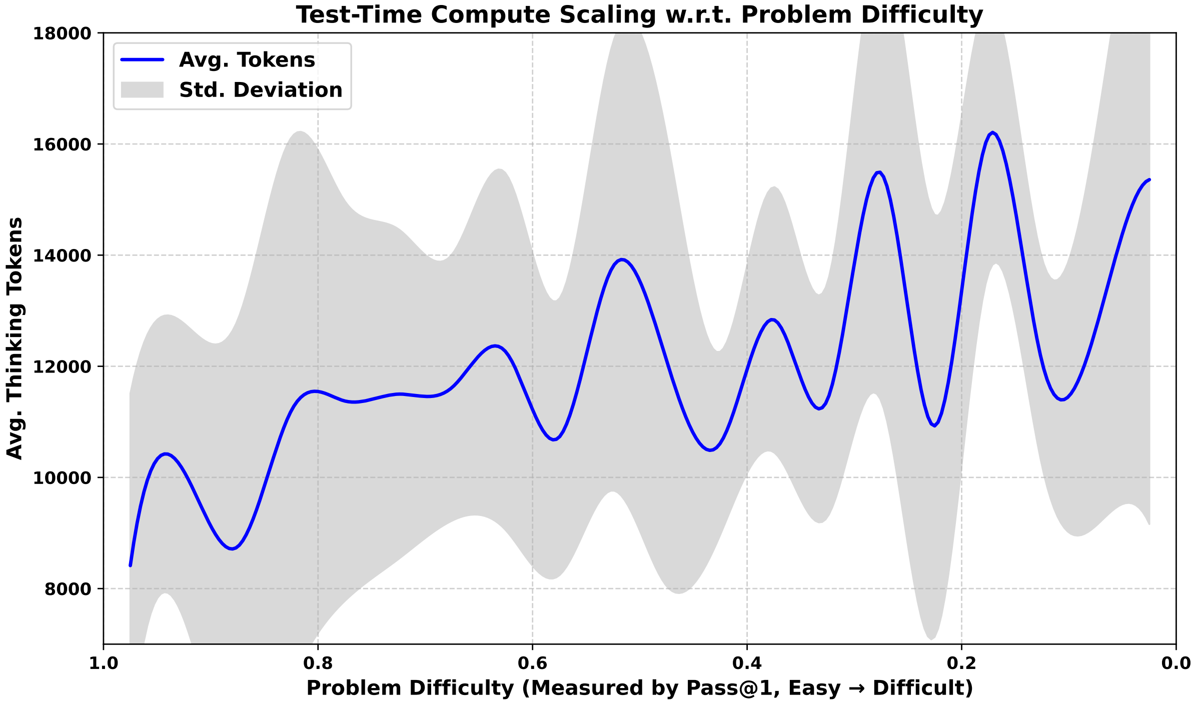

Figure 18 | Test-time compute scaling (measured by the number of thinking tokens generated to reach correct answers) as problem difficulty (measured by Pass@1) increases. The picture is smoothed using UnivariateSpline from SciPy with a smoothing factor of 5.

Figure 18 demonstrates how DeepSeek-R1 scales test-time compute to solve challenging problems from math competitions held in 2024 (the same set of problems used in Figure 17). DeepSeek-R1 achieves a 61.8% solve rate (Pass@1) by scaling test-time compute to an average of 8,793 thinking tokens per problem. Notably, the model adaptively adjusts its computational effort based on problem difficulty, using fewer than 7,000 thinking tokens for simple problems while dedicating more than 18,000 thinking tokens to the most challenging ones, which demonstrates DeepSeek-R1 allocates test-time compute adaptively based on problem complexity: on more complex problems, it tends to think for longer. Looking forward, we hypothesize that if token budget allocation were explicitly modeled during training, the disparity in token usage between easy and hard questions at test time could become even more pronounced.

Comparison of non-reasoning models: A key advantage of reasoning models like DeepSeek-R1 over non-reasoning models such as GPT-4o 0513 is their ability to scale effectively along the dimension of reasoning. Non-reasoning models typically generate solutions directly, without intermediate thinking steps, and rarely demonstrate advanced problem-solving techniques like self-reflection, backtracking, or exploring alternative approaches. On this same set of math problems, GPT-4o 0513 achieves only a 24.7% solve rate while generating 711 output tokens on average — an order of magnitude less than DeepSeek-R1. Notably, non-reasoning models can also scale test-time compute with traditional methods like majority voting, but those methods fail to close the performance gap with reasoning models, even when controlling for the total number of tokens generated. For example, majority voting across 16 samples per problem yields minimal improvement in GPT-4o's solve rate on the 2024 collection of competition-level math problems, despite consuming more total tokens than DeepSeek-R1. On AIME 2024, majority voting across 64 samples only increases GPT-4o's solve rate from 9.3% to 13.4%—still dramatically lower than DeepSeek-R1's 79.8% solve rate or o1's 79.2% solve rate. This persistent performance gap stems from a fundamental limitation: in majority voting, samples are generated independently rather than building upon each other. Since non-reasoning models lack the ability to backtrack or self-correct, scaling the sample size merely results in repeatedly sampling potentially incorrect final solutions without increasing the probability of finding correct solutions in any single attempt, making this approach highly token-inefficient.

Drawback: However, DeepSeek-R1's extended reasoning chains still sometimes fail to be thorough or become trapped in incorrect logic paths. Independently sampling multiple reasoning chains increases the probability of discovering correct solutions, as evidenced by the fact that DeepSeek-R1's Pass@64 score on AIME 2024 is 90.0%, significantly higher than its Pass@1 score of 79.8%. Therefore, traditional test-time scaling methods like majority voting or Monte Carlo Tree Search (MCTS) can complement DeepSeek-R1's long reasoning; specifically, majority voting further improves DeepSeek-R1's accuracy from 79.8% to 86.7%.

E.5. Performance of Each Stage on Problems of Varying Difficulty

Table 14 | Experimental results for each stage of DeepSeek-R1 on problems with varying difficulty levels in the LiveCodeBench dataset.

Difficulty Level	**DeepSeek-R1 Zero**	**DeepSeek-R1 Dev1**	**DeepSeek-R1 Dev2**	**DeepSeek-R1 Dev3**	**DeepSeek R1**
Easy	98.07	99.52	100.00	100.00	**100.00**
Medium	58.78	73.31	81.76	81.42	**83.45**
Hard	17.09	23.21	30.36	33.16	**34.44**

To further evaluate the performance of each stage of DeepSeek-R1 on problems of varying difficulty, we present the experimental results for each stage of DeepSeek-R1 on the LiveCodeBench dataset, as shown in Table 14. It can be observed that for each stage, simple problems are generally solved correctly, while the main improvements come from medium and hard problems. This fine-grained analysis demonstrates that each stage brings significant improvement on complex coding reasoning problems.

F. DeepSeek-R1 Distillation

LLMs are energy-intensive, requiring substantial computational resources, including high-performance GPUs and considerable electricity, for training and deployment. These resource demands present a significant barrier to democratizing access to AI-powered technologies, particularly in under-resourced or marginalized communities.

To address this challenge, we adopt a model distillation approach, a well-established technique for efficient knowledge transfer that has demonstrated strong empirical performance in prior work (Busbridge et al., 2025; Hinton et al., 2015). Specifically, we fine-tune open-source foundation models such as Qwen (Qwen, 2024b) and LLaMA (AI@Meta, 2024; Touvron et al., 2023) using a curated dataset comprising 800,000 samples generated with DeepSeek-R1. Details of the dataset construction are provided in Appendix B.3.3. We find that models distilled from high-quality teacher outputs consistently outperform those trained directly on human-generated data, corroborating prior findings on the efficacy of distillation (Busbridge et al., 2025).

For distilled models, we apply only SFT and do not include an RL stage, even though incorporating RL could substantially boost model performance. Our primary goal here is to