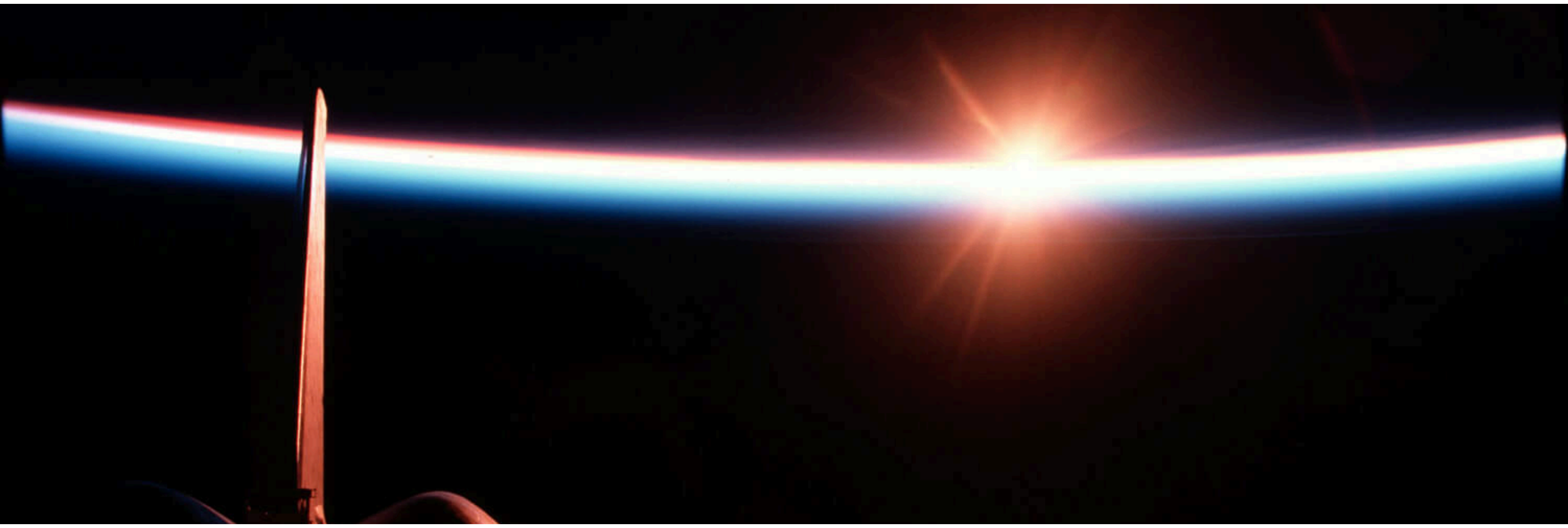

# DeepScaleR: Surpassing O1-Preview with a 1.5B Model by Scaling RL

**Michael Luo*, Sijun Tan*, Justin Wong†, Xiaoxiang Shi, William Tang, Manan Roongta, Colin Cai, Jeffrey Luo**

**Advisors: Tianjun Zhang*, Li Erran Li, Raluca Ada Popa, Ion Stoica**

*: Project Leads; †: Significant Contributor

## ✨ TL;DR

RL magic is in the air! We introduce `DeepScaleR-1.5B-Preview`, a language model finetuned from `Deepseek-R1-Distilled-Qwen-1.5B` using simple reinforcement learning (RL). It achieves an impressive **43.1% Pass@1** accuracy on AIME2024 (+**14.3% improvement** over the base model), surpassing the performance of OpenAI's `o1-preview` with just **1.5B** parameters. We **open sourced** our dataset, code and training logs for everyone to progress on scaling intelligence with RL.

🌐 Website, 🧑‍💻 Github, 🤗 HF Model, 🤗 HF Dataset, 📈 Wandb Logs, 🔎 Eval Logs

# DeepScaleR-1.5B-Preview

| Model | AIME 2024 | MATH 500 | AMC 2023 | Minerva Math | Olympiad Bench | Avg. |
|---|---|---|---|---|---|---|
| **DeepScaleR-1.5B-Preview** | **43.1** | **87.8** | **73.6** | **30.2** | **50.0** | **57.0** |
| DeepSeek-R1-Distill-Qwen-1.5B | 28.8 | 82.8 | 62.9 | 26.5 | 43.3 | 48.9 |
| O1-Preview | 40.0 | 81.4 | - | - | - | - |

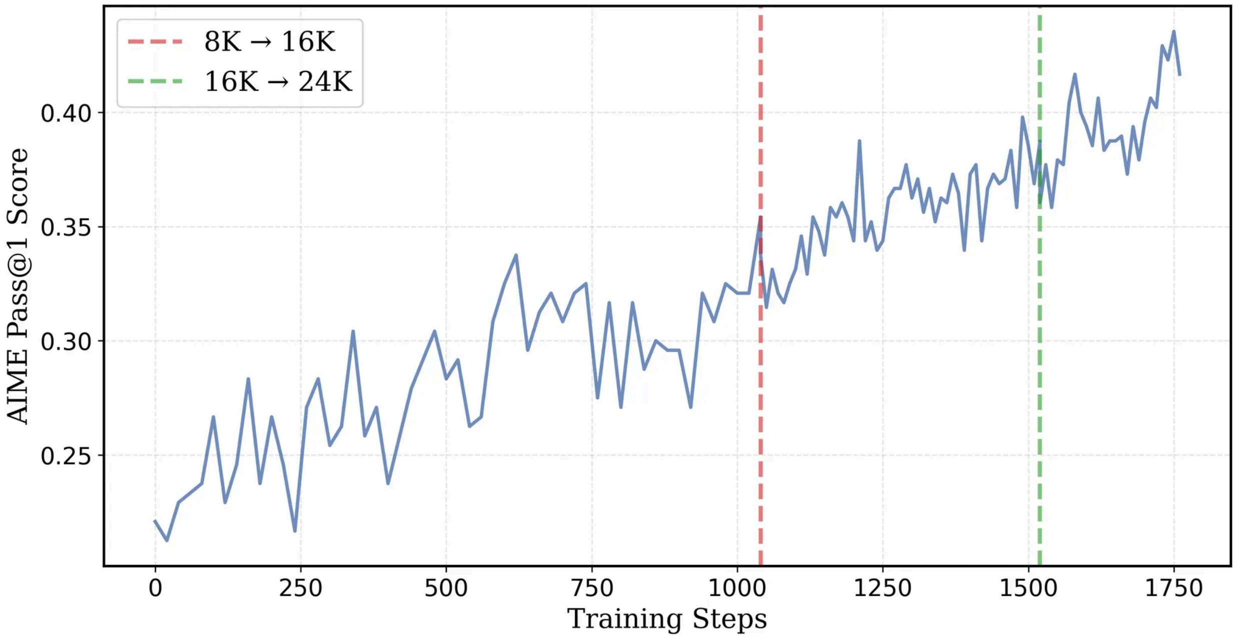

Figure1: DeepScaleR’s Pass@1 accuracy on AIME2024 as training progresses. At step 1040 and 1520, the context length is extended to 16K and 24K.

In this blog, we take a step towards unveiling the recipe of using RL to turn a small model into a strong reasoning model. We introduce **DeepScaleR-1.5B-Preview**, trained on 40K high-quality math problems with 3,800 A100 hours ($4500), outperforming OpenAI’s o1-preview on multiple competition-level math benchmarks.

# Introduction: Towards Democratizing RL for LLMs

The recent open-source release of Deepseek-R1 (a model comparable to OpenAI's o1) marks a significant leap forward in democratizing reasoning models. Yet, its' exact training recipe, hyperparameters, and underlying systems are still unavailable. In this work, we take a major step towards a fully open-recipe that scales up RL for reasoning models.

One of the biggest challenges in scaling RL is the high computational cost. For instance, we found that directly replicating DeepSeek-R1's experiments (⩾32K context, ~8000 steps) takes at least **70,000** A100 GPU hours—even for a 1.5B model. To address this, we leverage a distilled model and introduce a novel iterative lengthening scheme for RL, reducing the compute requirement to just **3,800** A100 GPU hours—an **18.42× reduction**—while achieving performance surpassing OpenAI's `o1-preview` with just a 1.5B model.

Our work demonstrates that developing customized reasoning models through RL can be both scalable and cost-efficient. In the remaining blog post, we'll walk through our dataset curation and training approach, present evaluation results, and share key insights from our findings.

# DeepScaleR's Recipe

## Dataset Curation

For our training dataset, we compiled **AIME problems from 1984-2023** and **AMC problems prior to 2023**, along with questions from the Omni-MATH and Still datasets, which feature problems from various national and international math competitions.

Our data processing pipeline consists of three key steps:

1. **Extracting Answers**: For datasets such as AMC and AIME, we use `gemini-1.5-pro-002` to extract answers from official AoPS solutions.
2. **Removing Redundant Questions**: We employ RAG with embeddings from `sentence-transformers/all-MiniLM-L6-v2` to eliminate duplicate problems. To prevent data contamination, we also check for overlaps between the training and test sets.

3. **Filtering Ungradable Questions:** Some datasets, such as Omni-MATH, include problems that cannot be evaluated using `sympy` and require an LLM judge. Since using LLM judges may slow down training and introduce noisy reward signals, we apply an additional filtering step to remove these ungradable questions.

After deduplication and filtering, our final training dataset consists of approximately **40,000** unique problem-answer pairs. We will expand our dataset for future runs.

## Reward Function

As advocated in Deepseek-R1, we employ an Outcome Reward Model (ORM) as opposed to a Process Reward Model(PRM) to avoid reward hacking. In summary, our reward function returns:

- `1` - If the LLM's answer passes basic LaTeX/Sympy checks.
- `0` - If the LLM's answer is incorrect or formatted incorrectly (e.g. missing `<think>`, `</think>` delimiters).

## Iterative Context Lengthening: Think Shorter, then Longer

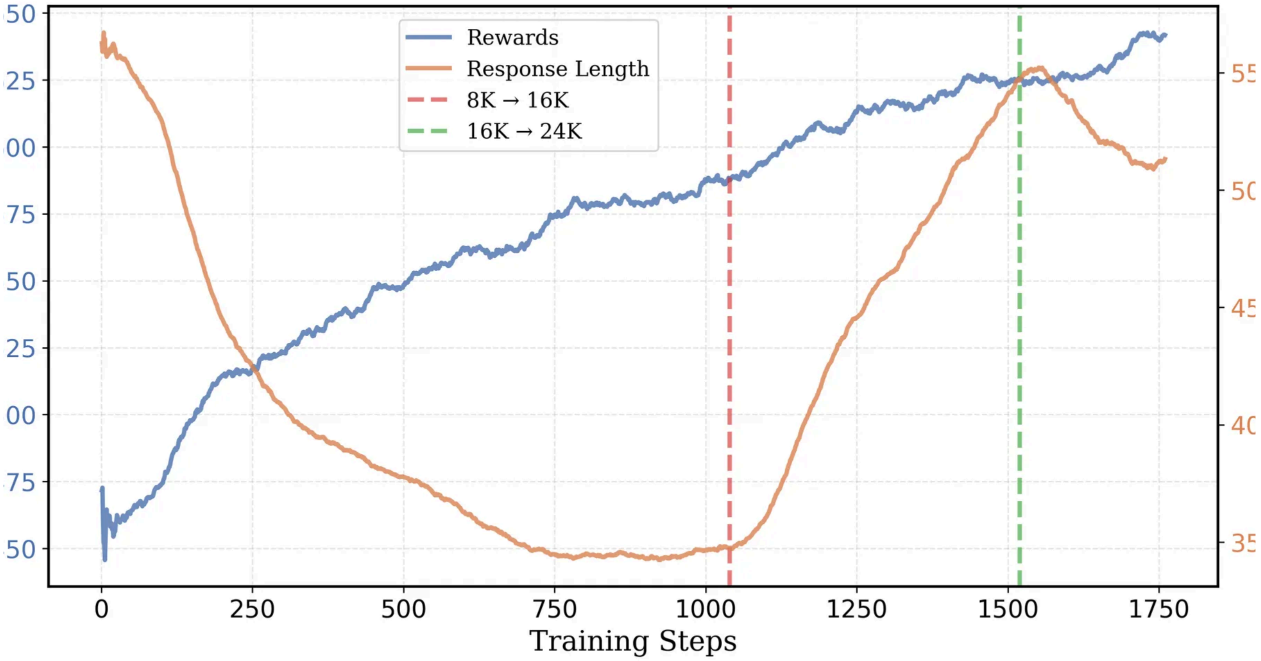

Figure2: DeepScaleR's average response length and training rewards as training progresses. The curves shows the running average over a window size of 100.

A key challenge in scaling RL for reasoning tasks is selecting the optimal context window for training. Reasoning workloads are highly compute-intensive, as they generate much longer outputs than standard tasks, slowing both trajectory sampling and policy gradient updates. Doubling the context window size increases training compute by at least 2×.

This introduces a fundamental tradeoff: longer contexts provide models more space to think yet significantly slows training, while shorter contexts accelerate training but may limit the model's ability to solve harder problems, which require long contexts. Therefore, striking the right balance between efficiency and accuracy is crucial.

In summary, our training recipe, which employs Deepseek's GRPO algorithm, follows two steps:

- First, we perform RL training with 8K max context for more effective reasoning and efficient training.
- Next, we scale up training to 16K and 24K contexts so that the model can solve more challenging, previously unsolved problems.

## Bootstrapping effective CoT with 8K context

Before launching our full training run, we evaluated `Deepseek-R1-Distilled-Qwen-1.5B` on AIME2024 and analyzed trajectory statistics. On average, incorrect responses contained **three times** more tokens than correct ones (20,346 vs. 6,395). This suggests that longer responses often lead to incorrect results. Hence, immediately training with long context windows may be inefficient, as most tokens are effectively wasted. Additionally, we observed in our evaluation logs that lengthy responses exhibit repetitive patterns, indicating that they do not contribute meaningfully to effective chain-of-thought (CoT) reasoning.

Given this insight, we initiated training with an 8K context, achieving an initial AIME2024 accuracy of 22.9%—just 6% below the original model. This strategy proved effective: Over the course of training, mean training rewards increased from 46% to 58%, while average response length dropped from 5,500 to 3,500 tokens (see Figure 2).

|  | Base model | DeepScaleR-1.5b-8k | Change |
|---|---|---|---|
| **AIME Pass@1** | 28.9% | 33.9% | +5% |
| **Average tokens for correct responses** | 6396.0 | 3661.2 | -2734.8 |
| **Average tokens for incorrect responses** | 20346.3 | 6976.8 | -13369.5 |
| **Average tokens overall** | 16335.6 | 5850.9 | –10484.7 |

More importantly, constraining output to 8K tokens led the model to utilize context more effectively. As shown in the table, our model generates significantly shorter responses for both correct and incorrect answers while surpassing the base model’s AIME accuracy by **5%**—with only **one-third** of the tokens.

## Extending to 16K context at the turning point

After approximately 1,000 steps, an interesting shift occurs for our 8K run: response length begins to increase again. However, this leads to diminishing returns—accuracy plateaus and eventually declines. At the same time, the response clipping ratio rises from 4.2% to 6.5%, indicating that more responses are being truncated at the context limit.

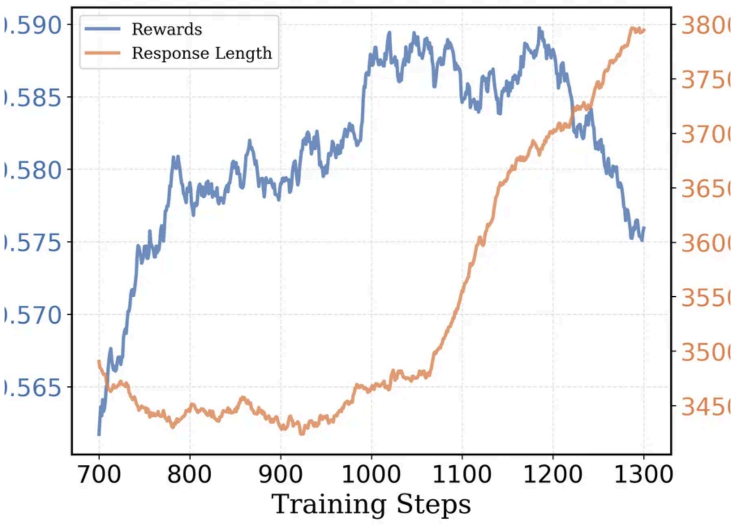

Figure 3: Response length goes back up after 1000 steps, but training rewards eventually declines for our 8K run.

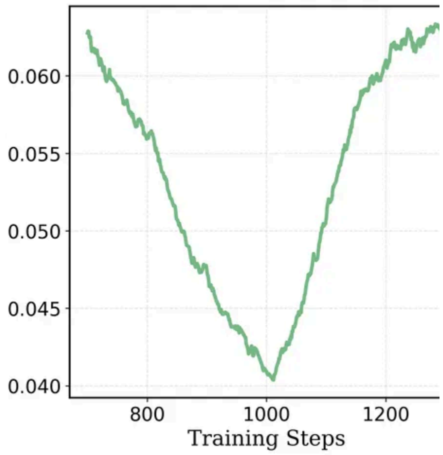

Figure 4: The response length clip ratio rises after 1000 steps for the 8K context run.

These results suggest that the model attempts to improve training rewards by "thinking longer." However, as it generates longer responses, it increasingly encounters the 8K context window ceiling, thus limiting further improvements.

Recognizing this as a natural transition point, we decided to "set the cage free and let the bird fly." We took the checkpoint at step 1,040—where response length began trending upward—and relaunched training with a 16K context window. This two-stage approach is significantly more efficient than training at 16K from the start: the 8K bootstrapping keeps the average response length at 3,000 tokens instead of 9,000, making training at this stage at least 2x faster.

Following this switch, we observe steady improvements in training rewards, response length, and AIME accuracy. After 500 additional steps, the average response length increases from 3500 to 5500 tokens, and the AIME2024 Pass@1 accuracy reaches 38%.

### Surpassing O1-preview with the 24K magic🪄

After training an additional 500 steps on 16K context, we noticed performance beginning to plateau—mean training rewards converged at 62.5%, AIME Pass@1 accuracy hovered around 38%, and response length started to decline again. Meanwhile, the maximum response clipping ratio crept up to 2%.

To make the final push towards o1-level performance, we decided to rollout the 24k magic— increasing the context window to 24K. We take our 16K run's checkpoint at step 480, and relaunch a training run with 24K context window.

With the extended context window, the model finally broke free. After around 50 steps, our model finally surpass 40% AIME accuracy and eventually reaches 43% at step 200. The *24K magic* was in full effect!

> 24k magic in the air 🔮
> —Bruno Mars

Overall, our training run consists of ~1,750 steps. The initial 8K phase was trained on 8 A100 GPUs, while the 16K and 24K phases scaled up training to 32 A100 GPUs. In total, the training took around 3,800 A100 hours, equivalent to roughly 5 days on 32 A100s and $4500 in terms of compute cost.

# Evaluation

We evaluate our model on competition-level mathematics benchmarks, including AIME 2024, AMC 2023, MATH-500, Minerva Math, and OlympiadBench. Below, Pass@1 accuracy is reported, averaged over 16 samples for each problem. The baselines we ran to verify scores are underlined.

| Model | AIME 2024 | MATH 500 | AMC 2023 | Minerva Math | Olympiad |
|---|---|---|---|---|---|
| Qwen-2.5-Math-7B-Instruct | 13.3 | 79.8 | 50.6 | 34.6 | 40.7 |
| rStar-Math-7B | 26.7 | 78.4 | 47.5 | - | 47.1 |
| Eurus-2-7B-PRIME | 26.7 | 79.2 | 57.8 | 38.6 | 42.1 |
| Qwen2.5-7B-SimpleRL | 26.7 | 82.4 | 62.5 | **39.7** | 43.3 |
| DeepSeek-R1-Distill-Qwen-1.5B | 28.8 | 82.8 | 62.9 | 26.5 | 43.3 |
| Still-1.5B | 32.5 | 84.4 | 66.7 | 29.0 | 45.4 |
| **DeepScaleR-1.5B-Preview** | **43.1** | **87.8** | **73.6** | 30.2 | **50.0** |
| O1-Preview | 40.0 | 81.4 | - | - | - |

We compare DeepScaleR with the base DeepSeek model we use, as well as recent academic works exploring RL for reasoning tasks. DeepScaleR significantly outperforms the base model across all benchmarks, achieving a **14.4%** absolute gain on AIME2024 and an **8.1%** overall improvement. Additionally, DeepScaleR surpasses recent academic works such as rSTAR, Prime, and SimpleRL, which are finetuned from 7B models. As shown in Figure 5, DeepScaleR achieves O1-preview-level performance with only 1.5B parameters—a remarkable efficiency gain.

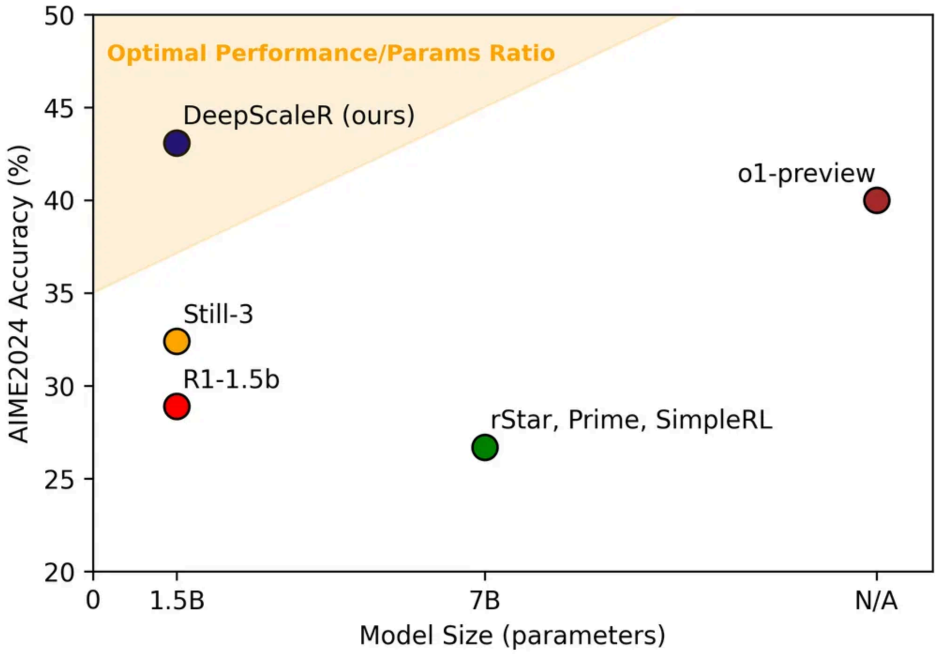

*Figure5: AIME Accuracy vs Model Size - DeepScaleR achieves the most Pareto efficient combination of performance and size.*

## Key Takeaways

**RL scaling can manifest in small models as well.** `Deepseek-R1` demonstrates that applying RL directly on small models is not as effective as distillation. Their ablations shows that RL on `Qwen-32B` achieves 47% on AIME, whereas distillation alone reaches 72.6%. A common myth is that RL scaling only benefits large models. However, with high-quality SFT data distilled from larger models, smaller models can also learn to reason more effectively with RL. Our results confirm this: RL scaling improved AIME accuracy from 28.9% to 43.1%! These findings suggest that neither SFT nor RL alone is sufficient. Instead, by combining high-quality SFT distillation with RL scaling, we can truly unlock the reasoning potential of LLMs.

**Iterative lengthening enables more effective length scaling.** Prior works [1, 2] indicate that training RL directly on 16K context yields no significant improvement over 8K, likely due to insufficient compute for the model to fully exploit the extended context. And a recent work [3] suggests longer response lengths consists of redundant self-reflection that leads to incorrect results. Our experiments are consistent with these findings. By first optimizing reasoning at shorter contexts (8K), we enable faster and more effective training in subsequent 16K and 24K runs. This iterative approach grounds the model in effective thinking patterns before scaling to longer contexts, making RL-based length scaling more efficient.

## Conclusion

Our work aims to reveal the scaling effects of RL on LLMs and make it available for