A Practical Two-Stage Recipe for Mathematical LLMs: Maximizing Accuracy with SFT and Efficiency with Reinforcement Learning

Hiroshi Yoshihara [1 2] **Taiki Yamaguchi** [3] **Yuichi Inoue** [4]

Abstract

Enhancing the mathematical reasoning of Large Language Models (LLMs) is a pivotal challenge in advancing AI capabilities. While Supervised Fine-Tuning (SFT) and Reinforcement Learning (RL) are the dominant training paradigms, a systematic methodology for combining them to maximize both accuracy and efficiency remains largely unexplored. This paper introduces a practical and effective training recipe that strategically integrates extended SFT with RL from online inference (GRPO). We posit that these methods play complementary, not competing, roles: a prolonged SFT phase first pushes the model's accuracy to its limits, after which a GRPO phase dramatically improves token efficiency while preserving this peak performance. Our experiments reveal that extending SFT for as many as 10 epochs is crucial for performance breakthroughs, and that the primary role of GRPO in this framework is to optimize solution length. The efficacy of our recipe is rigorously validated through top-tier performance on challenging benchmarks, including a high rank among over $2,200$ teams in the strictly leak-free AI Mathematical Olympiad (AIMO). This work provides the community with a battle-tested blueprint for developing state-of-the-art mathematical reasoners that are both exceptionally accurate and practically efficient. To ensure full reproducibility and empower future research, we will open-source our entire framework, including all code, model checkpoints, and training configurations at https://github.com/analokmaus/kaggle-aimo2-fast-math-r1.

1. Introduction

The remarkable advancements of Large Language Models (LLMs) have demonstrated their potential across a vast spectrum of applications. Beyond their well-established capabilities in natural language understanding and conversation, the frontier of AI research is increasingly focused on enhancing their reasoning abilities, which are essential to solve complex and challenging problems (Qwen-Team, 2024; OpenAI, 2024; 2025; DeepSeek-AI, 2025). Among the diverse domains for evaluating reasoning, mathematical problem-solving stands out as an ideal testbed. It demands not only factual knowledge but also a complex interplay of strategic planning, step-by-step logical inference, and self-correction, thereby providing a comprehensive benchmark for the problem-solving capabilities of current models.

To enhance the mathematical reasoning of LLMs, two primary paradigms have been explored: Supervised Fine-Tuning (SFT) and Reinforcement Learning (RL). SFT, which uses high-quality step-by-step solutions datasets, has been instrumental in the bootstrapping of model capabilities (Chen et al., 2024; Qwen-Team, 2024). However, its effectiveness is intrinsically tied to the scale and quality of the demonstration data, potentially leading to a performance plateau. On the other hand, RL offers a promising avenue for models to learn from their own generated solutions, moving beyond the confines of static datasets. Recently, Group Relative Policy Optimization (GRPO) has emerged, showing promise in improving sampling efficiency (Shao et al., 2024; DeepSeek-AI, 2025). Yet, it remains unclear whether GRPO alone is sufficient to maximize performance, and a principled, systematic methodology for combining it with SFT is notably absent.

This paper bridges this gap by proposing a practical recipe that strategically integrates SFT and GRPO to unlock new levels of mathematical reasoning in LLMs. We posit that these two methods are not competing but rather play highly complementary roles. Our core idea is that an extended SFT phase is first employed to establish a strong performance baseline, which is then refined by a GRPO phase focused on enhancing efficiency without compromising accuracy. This sequential and synergistic approach provides a clear, reproducible pathway for developing high-performing

[1] Aillis Inc., Tokyo, Japan [2] Department of Health Policy and Public Health, Graduate School of Pharmaceutical Sciences, The University of Tokyo, Tokyo, Japan [3] Rist Inc., Kyoto, Japan [4] Sakana AI, Tokyo, Japan. Correspondence to: Yuichi Inoue <y.inoue@sakana.ai>.

The second AI for MATH Workshop at the 42nd International Conference on Machine Learning, Vancouver, Canada. 2025. Copyright 2025 by the author(s).

mathematical reasoners.

Our investigation yields several key insights into the effective training of mathematical LLMs. First, we experimentally demonstrate that prolonged SFT is crucial for performance breakthroughs. Contrary to the common practice of short-duration fine-tuning such as cold start, we find that although initial epochs may show a temporary dip in performance, extending the SFT process for as long as 10 epochs consistently and significantly boosts the model's problem-solving accuracy. Second, we uncover a new primary role for GRPO in this combined framework. While prior work often associates preference optimization with direct accuracy gains, our findings indicate that GRPO excels at dramatically improving token efficiency. After our intensive SFT stage, GRPO maintains or slightly improves the high accuracy achieved, while substantially shortening the length of generated solutions. Third, we establish a clear synergistic relationship: SFT is responsible for pushing the performance ceiling of the model, while GRPO is responsible for optimizing the solution generation process, making the high-performing model more practical and efficient for real-world applications.

To validate the efficacy of our proposed recipe, we conduct extensive evaluations on a suite of challenging benchmarks, including AIME 2024 and AIME 2025. Crucially, we also test our model on the AI Mathematical Olympiad (AIMO) (Frieder et al., 2024), one of the most competitive benchmarks with stringent measures against data leakage. Our method achieves top-tier performance, demonstrating its practical effectiveness and robustness in a highly competitive setting. These results not only underscore the power of our combined SFT-GRPO strategy but also offer the community a valuable, battle-tested recipe for advancing the frontiers of mathematical reasoning with LLMs.

To support the reproducibility of our work, we will release key artifacts at https://github.com/analokmaus/kaggle-aimo2-fast-math-r1. The release will include our final model weights, the complete source code for the SFT and GRPO training and evaluation procedures, all curated datasets, and the full set of checkpoints from the GRPO stage. We believe these resources will enable the community to rigorously verify and build upon our proposed training methodology.

2. Related Work

LLM Reasoning. The performance of Large Language Models (LLMs) is well-known to improve with the scaling of training compute budgets, as established by scaling laws (Kaplan et al., 2020; Hoffmann et al., 2022). More recently, a parallel research direction has focused on increasing the computational budget at inference time, a strategy often referred to as test-time scaling. Several studies have demonstrated that allocating more compute during the inference phase—for instance, by generating a larger number of tokens or candidate solutions—can significantly enhance LLM performance on complex reasoning tasks (Li et al., 2022; Lewkowycz et al., 2022; Brown et al., 2024; Wu et al., 2025; Misaki et al., 2025). This line of work underscores the value of trading inference-time resources for higher accuracy.

Post-training for Reasoning LLMs. A significant body of work has focused on enhancing the reasoning abilities of LLMs through post-training refinement (Qwen-Team, 2024; OpenAI, 2025; DeepSeek-AI, 2025; Ye et al., 2025; Muennighoff et al., 2025; Team, 2025). Methods such as those used for OpenAI's o1 (OpenAI, 2024) and o3 (OpenAI, 2025), and DeepSeek-R1 (DeepSeek-AI, 2025), fine-tune models using supervised learning and/or reinforcement learning. This trend has been mirrored in the open-source community, with a growing number of studies reporting substantial improvements in reasoning by post-training models (Chen et al., 2024; Face, 2025; Zeng et al., 2025; Wen et al., 2025; Dang & Ngo, 2025). Despite this progress, a definitive and principled training recipe has yet to emerge, leaving the optimal combination and scheduling of techniques an open question. Furthermore, much of the existing work has centered on maximizing accuracy, often relying on increased token generation at inference time as a primary driver of performance. Consequently, a practical training methodology that explicitly considers both accuracy and inference efficiency remains a critical, yet largely underexplored, area of research.

Efficient Reasoning. While the aforementioned studies focus on improving reasoning accuracy, a parallel line of inquiry addresses the challenge of reasoning efficiency. The advent of Chain-of-Thought (CoT) prompting (Wei et al., 2022), despite its success, often leads to the "overthinking" phenomenon, where models generate excessively verbose rationales, incurring substantial computational overhead and latency. To mitigate this, one major research direction has focused on augmenting the model's intrinsic ability to reason concisely through training-centric paradigms. The first approach involves Reinforcement Learning (RL), where reward functions are explicitly designed to penalize generation length, thereby guiding the model to discover shorter yet effective reasoning paths (Luo et al., 2025; Yeo et al., 2025; Aggarwal & Welleck, 2025). A second approach leverages Supervised Fine-Tuning (SFT) with curated, variable-length CoT data. Such datasets are typically created either by post-processing and compressing verbose reasoning traces from a teacher model (Kang et al., 2025; Xia et al., 2025) or by prompting models to generate shorter solutions during the data collection phase itself (Liu et al., 2024; Munkhbat et al., 2025). Our work is situated within this context but

Table 1: **Performance and the mean token usage across model sizes for AIME 2024 and 2025.** *Acc.* represents mean Pass@1(%). *Original* means Deepseek-R1-Distill-Qwen 14B.

	AIME 2024						**AIME 2025**					
	1.5B		7B		14B		1.5B		7B		14B	
Method	Acc.	Tokens	Acc.	Tokens	Acc.	Tokens	Acc.	Tokens	Acc.	Tokens	Acc.	Tokens
Original	27.8	11,235	51.0	10,136	63.3	9,590	22.3	11,154	38.1	10,612	46.7	10,602
Original + RL	**29.5**	**10,702**	53.6	7,735	60.9	7,255	23.2	10,354	39.0	8,176	41.8	8,246
+ SFT (10 epochs)	26.0	13,014	52.0	10,647	65.2	10,268	22.1	12,826	38.3	11,507	49.7	11,264
+ SFT (10 epochs) + RL	27.3	11,767	**54.7**	**9,577**	**66.0**	**7,932**	**23.2**	**11,480**	**39.8**	**10,445**	**49.2**	**9,066**

offers a distinct, synergistic perspective. While prior works often treat accuracy and efficiency as a direct trade-off to be optimized jointly, our recipe decouples these objectives. We first employ an extended SFT phase dedicated to maximizing problem-solving accuracy, and subsequently use GRPO not primarily for further accuracy gains, but to refine the high-performing model to be significantly more token-efficient.

3. Methods

Our training methodology proceeds in two principal stages. The first stage involves intensive Supervised Fine-Tuning (SFT) using a specially curated dataset of high-difficulty mathematical reasoning problems. This is followed by a second stage where GRPO is applied to enhance the model's token efficiency while preserving its reasoning accuracy.

3.1. Stage 1: Supervised Fine-Tuning (SFT)

The SFT dataset was meticulously constructed by amalgamating data from three distinct sources. From OpenR1 Math (Face, 2025), we selected approximately 3,000 examples where the reference DeepSeek-R1-Distill-Qwen model's solution trace was notably long, exceeding 12,800 tokens, and where its accuracy was above 50%. An additional 3,000 examples were included from the same source where the R1 model's accuracy fell between 50% and 75%. The openr1 hard dataset contributed around 2,500 challenging samples sourced from `open-r1-math-220k` (Face, 2025); these were problems that the DeepSeek-R1-Distill-Qwen-32B model was unable to solve in four attempts. Finally, we incorporated the second-stage SFT data from the Light-R1-SFT Data (Wen et al., 2025).

After merging these sources, we removed duplicate entries. For each unique problem, we selected the correct generation that exhibited the shortest token length. In instances where samples from Light-R1-SFT Data lacked ground truth answers, we extracted and substituted the answers from the R1 model's solution traces. This comprehensive process yielded a high-difficulty dataset consisting of 7,900 problem-solution trace-answer triplets.

For the SFT process, full-parameter SFT was performed, executed on a system equipped with 8 NVIDIA H200 GPUs. The training was configured with a per device train batch size of 1 and gradient accumulation steps of 8. The training process was extended to 10 epochs. The maximum sequence length was set to 24,000, and packing was enabled. A learning rate of 1e-5 was used with a cosine learning rate scheduler. The models were trained using the system prompt: "Please reason step by step, and put your final answer within \boxed{{}}".

3.2. Stage 2: GRPO for Enhanced Token Efficiency

While the SFT stage improved the model's accuracy, it also led to a tendency to generate longer, sometimes redundant, reasoning traces. To mitigate this and specifically enhance token efficiency, we subsequently applied GRPO.

The dataset employed for GRPO training was the second-stage data from Light-R1-SFT Data (Wen et al., 2025), maintaining consistency with one of the SFT data sources. The SFT-tuned model resulting from Stage 1 served as the initialization point for making GRPO phase stable. The reward function for GRPO was designed with three key components to guide the learning process. Firstly, a Format Reward provided a binary signal (+1 or 0) based on adherence to the expected output structure, particularly matching the regular expression pattern `r"^.*?oxed\{(.*?)\}.*?</think>.*?$"`. Secondly, a Cosine Similarity Reward was implemented (Yeo et al., 2025). For outputs that conformed to the required format, this reward measured the cosine similarity between the embedding of the model's generated reasoning trace and that of a reference correct trace, where available, or used a proxy based on answer correctness. This reward was scaled to range from 0.1 to 1.0 for correct answers, thereby more subtly penalizing longer correct traces, and from -1.0 to -0.1 for incorrect answers, which more severely penalized shorter incorrect traces. The maximum trace length considered for this reward was 30,000 tokens. This component offers a more nuanced, continuous feedback signal compared to a simple accuracy-based reward. Thirdly, a Length Penalty was applied, proportional to the length of the generated output, to explicitly discourage overly verbose solutions and promote conciseness.

The GRPO training was configured with num generations set to 8 and a beta value of 0.04. The per device train batch size was 2, with gradient accumulation steps of 8. This stage was conducted for 50 steps, using a maximum completion length of 16,384. The learning rate was 4e-6, again with a cosine scheduler. The system prompt was: "You are a helpful and harmless assistant. You are Qwen developed by Alibaba. You should think step-by-step. Return final answer within \boxed{{}}".

4. Experiments

In this section, we present a series of experiments designed to rigorously evaluate our proposed training recipe, which was developed with the primary goal of achieving high performance on the AI Mathematical Olympiad (AIMO). We first describe our experimental setup, including the models and benchmarks used for evaluation. We then analyze the impact of our combined SFT and RL approach on both accuracy and efficiency across standard benchmarks. Finally, we report the performance of our best model on AIMO, validating the effectiveness of our recipe in the highly competitive environment.

4.1. Benchmarks

A significant challenge in evaluating LLMs is the rapid overfitting of models to publicly available benchmarks. To address this, the AI Mathematical Olympiad (AIMO) is structured as a competition with strictly controlled test data (Frieder et al., 2024). It provides a dedicated evaluation environment during the competition period, ensuring that the benchmark remains entirely leak-free as participants have no access to the test cases. AIMO uses a public set of 50 problems for in-competition performance monitoring and a private set of 50 problems for the final performance assessment after the competition concludes. In this paper, we evaluate our models using data from the second AIMO competition. In addition, to evaluate the performance and efficiency of our method, we used the competition-level benchmarks AIME 2024 and AIME 2025, and the standard mathematical reasoning benchmark MATH-500 (Hendrycks et al., 2021). Throughout the following sections, we report model performance as pass@1, averaged over 64 sampling runs.

4.2. Models

Due to the constraints of the AIMO competition, where many teams utilized 14B-parameter models, we also focused on maximizing the performance of a model of this size. To investigate how our training recipe generalizes to other model scales, we also experimented with 1.5B and 7B parameter models. For our base models, we used the DeepSeek-R1-Distill-Qwen models (DeepSeek-AI, 2025), which were also a popular choice among AIMO participants.

Table 2: **Mean Pass@1 (%) and mean output token length on MATH500.**

	MATH500					
	1.5B		7B		14B	
Method	Acc.	Tokens	Acc.	Tokens	Acc.	Tokens
Original	78.2	1,915	83.8	2,382	86.4	2,556
+ SFT (10 epochs)	79.6	2,450	84.4	3,082	88.1	2,969
+ SFT (10 epochs) + RL	**83.9**	**2,210**	**89.0**	**2,849**	**91.2**	**2,084**

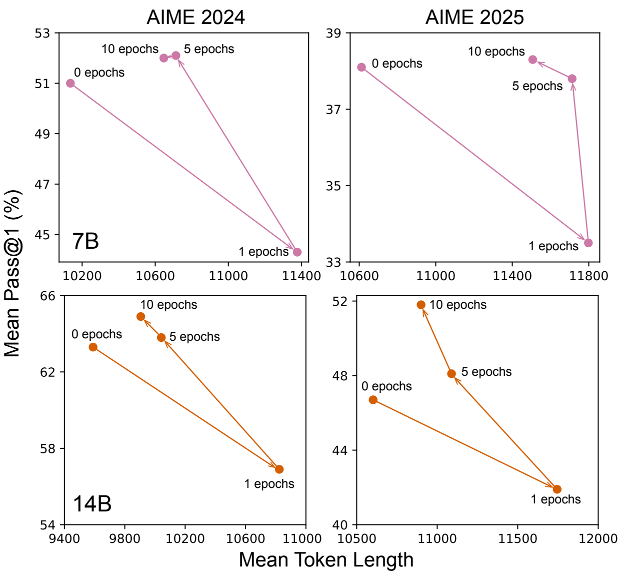

Figure 1: **Performance comparison SFT on AIME 2024 and 2025.** Mean Pass@1 accuracy and mean token length per training epoch.

4.3. Results on AIME

Table 1 displays the accuracy and the mean number of output tokens on AIME 2024 and 2025. The results show that for both benchmarks, our proposed method of 10-epoch SFT followed by RL improves both accuracy and token efficiency over the original models across nearly all model sizes. Interestingly, the accuracy gains from our method become more pronounced as the model size increases. When RL is applied directly to the original model, the 14B model shows improved token efficiency but a degradation in accuracy. However, by first applying 10-epoch SFT, we significantly boost the model's accuracy, and the subsequent RL phase further enhances it. The 7B model exhibits a similar trend,

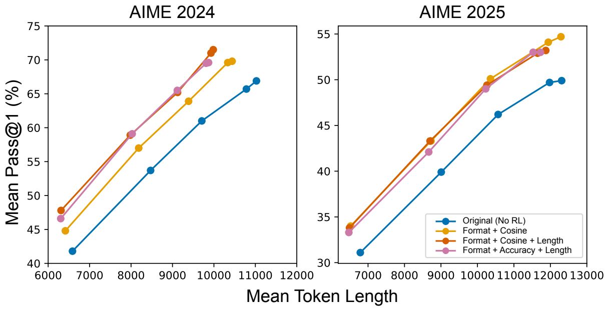

Figure 2: **Ablation study of Reward functions.** The mean Pass@1 accuracy versus the mean token length for different combinations of reward functions. To clearly illustrate the performance at different token budgets, points are also plotted for outputs truncated at maximum token lengths of 8k, 12k, 16k, 24k, and 32k.

with accuracy improving incrementally first with extensive SFT and then with RL. In contrast, for the 1.5B model, the 10-epoch SFT did not yield a substantial accuracy increase.

4.4. Results on MATH-500

Table 2 presents the results on MATH-500, a benchmark considered less difficult than AIME. The lower number of tokens used for inference suggests that MATH-500 has a different difficulty profile. Despite this difference, we observe a trend similar to the AIME results when applying our training recipe. The 10-epoch SFT phase increases both accuracy and mean token length, while the subsequent RL phase successfully improves accuracy further while reducing the token length. This demonstrates that our method can enhance accuracy while managing token usage, even on benchmarks with different difficulty levels and token requirements for inference.

4.5. The Impact of Extensive SFT

As shown in Table 1, applying RL within our framework helps reduce the number of tokens required for inference. However, for larger models like the 14B variant, applying RL alone can be unstable, leading to a drop in accuracy. Indeed, prior work has reported the necessity of an initial SFT phase to stabilize the RL process (DeepSeek-AI, 2025). We investigated how much SFT is necessary before RL when building a specialized mathematical model. As shown in Figure 1, there is a clear trend where accuracy improves as the number of SFT epochs increases. Interestingly, training for only one epoch significantly increases the average token length but leads to a sharp drop in accuracy. These findings suggest that for creating a specialized math model, a prolonged SFT phase is crucial for enabling stable and effective subsequent RL training.

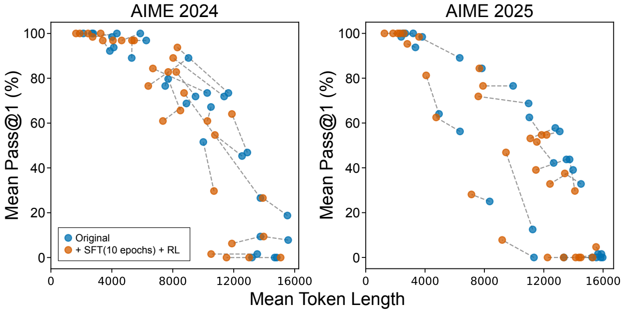

Figure 3: **Per-problem changes in mean pass@1 and token length from the original model to our proposed recipe.** This plot illustrates the shift in performance and efficiency for each problem after applying our training recipe.

4.6. Ablation on Reward Functions

We investigated how the combination of the reward functions for RL affects our model's performance. To test this, For this analysis, we evaluated three distinct reward configurations. The *accuracy reward* is a binary reward for correct answers (DeepSeek-AI, 2025), while the *length penalty* penalizes longer solutions. The *cosine reward* (Yeo et al., 2025) provides a reward based on both accuracy and solution length. A *format reward* was used in all experiments to ensure proper output structure. To clearly illustrate the trade-off between inference tokens and accuracy, we also plot accuracy at different token budget cutoffs. Figure 2 shows that incorporating a length penalty effectively reduces the average number of tokens required for inference. When comparing the accuracy-based rewards, the cosine reward yields slightly higher accuracy than the binary accuracy reward alone. In our final training recipe, we adopted a combination of the cosine and length penalty to strike a balance between token efficiency and accuracy.

4.7. Analysis of Per-Problem Performance

To gain a more granular understanding of our recipe's impact, we conducted a per-problem analysis of its effects on both accuracy and token efficiency. For this experiment, we evaluated the performance of our final model against the original DeepSeek-R1-Distill-Qwen-14B baseline on every problem in the AIME 2024 and AIME 2025 benchmarks. As illustrated in Figure 3, the results demonstrate a clear and consistent improvement across the majority of problems. For most questions, our recipe not only enhances the mean pass@1 score but also substantially reduces the solution length, indicating a significant gain in overall efficiency. For problems where the baseline model already achieved a high accuracy (i.e., approaching 100%), our method successfully maintains or further improves accuracy without any instances of performance degradation. In the mid-range of difficulty, where the accuracy was between 10% and 80%, our model demonstrates its most significant impact. While a few problems show minor regressions, the overwhelming trend is a simultaneous improvement in both accuracy and token efficiency. However, for the most challenging problems where the accuracy was initially low, our recipe shows limited gains in improving its performance. This suggests that improving performance on the most difficult problems remains a key challenge for future work.

4.8. Final Performance on the AIMO Benchmark

Finally, we evaluated our proposed recipe on the AIMO benchmark, which is both highly challenging and completely leak-free, to assess its real-world performance. Since the number of evaluation in the AIMO competition is strictly limited, we assess the performance of our recipe by its final ranking in the competition. Our model achieved a score of $29/50$ on the public set (equivalent to 4th place) and $28/50$ on the private set (equivalent to 8th place) out of $2,212$ competing teams. Achieving a consistently high score on both the public and private sets, amidst a large number of participants, demonstrates that our method is robust and capable of delivering genuinely high performance.

5. Conclusion

In this work, we proposed and validated a practical training recipe that synergistically combines Supervised Fine-Tuning (SFT) and Reinforcement Learning (RL) with GRPO to advance the mathematical reasoning of LLMs. Our core finding is that these two methods play complementary roles: an extended SFT phase is crucial for pushing the model's accuracy to its limits, while a subsequent GRPO phase dramatically improves token efficiency without compromising this peak performance. This sequential strategy moves beyond viewing SFT and RL as competing alternatives, establishing a clear, effective pathway to developing models that are both highly accurate and efficient.

The efficacy of our approach was rigorously demonstrated through top-tier performance on challenging benchmarks, including AIME and MATH. Most notably, our model achieved a high rank in the AI Mathematical Olympiad (AIMO), a highly competitive and strictly leak-free competition, confirming the robustness and real-world effectiveness of our method. These results provide a battle-tested blueprint for the community, showcasing a reliable method to build the next generation of state-of-the-art mathematical reasoners that balance exceptional performance with practical applicability.

References

Aggarwal, P. and Welleck, S. L1: Controlling how long a reasoning model thinks with reinforcement learning. *arXiv preprint arXiv:2503.04697*, 2025.

Brown, B., Juravsky, J., Ehrlich, R., Clark, R., Le, Q. V., Ré, C., and Mirhoseini, A. Large language monkeys: Scaling inference compute with repeated sampling. *arXiv preprint arXiv:2407.21787*, 2024.

Chen, G., Liao, M., Li, C., and Fan, K. Alphamath almost zero: Process supervision without process. In *Advances in Neural Information Processing Systems*, 2024.

Dang, Q.-A. and Ngo, C. Reinforcement learning for reasoning in small llms: What works and what doesn't. *arXiv preprint arXiv:2503.16219*, 2025.

DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. arXiv preprint arXiv:2501.12948, 2025.

Face, H. Open r1: A fully open reproduction of deepseek-r1, January 2025. URL https://github.com/huggingface/open-r1.

Frieder, S., Bealing, S., Nikolaiev, A., Smith, G. C., Buzzard, K., Gowers, T., Liu, P. J., Loh, P.-S., Mackey, L., de Moura, L., Roberts, D., Sculley, D., Tao, T., Balduzzi, D., Coyle, S., Gerko, A., Holbrook, R., Howard, A., and Markets, X. Ai mathematical olympiad - progress prize 2. https://kaggle.com/competitions/ai-mathematical-olympiad-progress-prize-2, 2024. Kaggle.

Hendrycks, D., Burns, C., Kadavath, S., Arora, A., Basart, S., Tang, E., Song, D., and Steinhardt, J. Measuring mathematical problem solving with the MATH dataset. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021. URL https://openreview.net/forum?id=7Bywt2mQsCe.

Hoffmann, J., Borgeaud, S., Mensch, A., Buchatskaya, E., Cai, T., Rutherford, E., de Las Casas, D., Hendricks, L. A., Welbl, J., Clark, A., Hennigan, T., Noland, E., Millican, K., van den Driessche, G., Damoc, B., Guy, A., Osindero, S., Simonyan, K., Elsen, E., Rae, J. W., Vinyals, O., and Sifre, L. Training compute-optimal large language models, 2022. URL https://arxiv.org/abs/2203.15556.

Kang, Y., Sun, X., Chen, L., and Zou, W. C3ot: Generating shorter chain-of-thought without compromising effectiveness. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 24312–24320, 2025.

Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., and Amodei, D. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Lewkowycz, A., Andreassen, A., Dohan, D., Dyer, E., Michalewski, H., Ramasesh, V., Slone, A., Anil, C., Schlag, I., Gutman-Solo, T., et al. Solving quantitative reasoning problems with language models. *Advances in Neural Information Processing Systems*, 2022.

Li, Y., Choi, D., Chung, J., Kushman, N., Schrittwieser, J., Leblond, R., Eccles, T., Keeling, J., Gimeno, F., Dal Lago, A., et al. Competition-level code generation with alphacode. *Science*, 378(6624):1092–1097, 2022.

Liu, T., Guo, Q., Hu, X., Jiayang, C., Zhang, Y., Qiu, X., and Zhang, Z. Can language models learn to skip steps? In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=w4AnTVxAO9.

Luo, H., Shen, L., He, H., Wang, Y., Liu, S., Li, W., Tan, N., Cao, X., and Tao, D. O1-pruner: Length-harmonizing fine-tuning for o1-like reasoning pruning. *arXiv preprint arXiv:2501.12570*, 2025.

Misaki, K., Inoue, Y., Imajuku, Y., Kuroki, S., Nakamura, T., and Akiba, T. Wider or deeper? scaling llm inference-time compute with adaptive branching tree search. *arXiv preprint arXiv:2503.04412*, 2025.

Muennighoff, N., Yang, Z., Shi, W., Li, X. L., Fei-Fei, L., Hajishirzi, H., Zettlemoyer, L., Liang, P., Candès, E., and Hashimoto, T. s1: Simple test-time scaling. arXiv preprint arXiv:2501.19393, 2025.

Munkhbat, T., Ho, N., Kim, S. H., Yang, Y., Kim, Y., and Yun, S.-Y. Self-training elicits concise reasoning in large language models. *arXiv preprint arXiv:2502.20122*, 2025.

OpenAI. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.

OpenAI. Competitive programming with large reasoning models. arXiv preprint arXiv:2502.06807, 2025.

Qwen-Team. Qwq: Reflect deeply on the boundaries of the unknown, November 2024. URL https://qwenlm.github.io/blog/qwq-32b-preview/.

Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Bi, X., Zhang, H., Zhang, M., Li, Y., Wu, Y., et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Team, K. Kimi k1.5: Scaling reinforcement learning with llms. arXiv preprint arXiv:2501.12599, 2025.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., brian ichter, Xia, F., Chi, E. H., Le, Q. V., and Zhou, D. Chain of thought prompting elicits reasoning in large language models. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=_VjQlMeSB_J.

Wen, L., Cai, Y., Xiao, F., He, X., An, Q., Duan, Z., Du, Y., Liu, J., Tang, L., Lv, X., Zou, H., Deng, Y., Jia, S., and Zhang, X. Light-r1: Curriculum sft, dpo and rl for long cot from scratch and beyond, 2025. URL https://arxiv.org/abs/2503.10460.

Wu, Y., Sun, Z., Li, S., Welleck, S., and Yang, Y. Inference scaling laws: An empirical analysis of compute-optimal inference for LLM problem-solving. In *International Conference on Learning Representations*, 2025.

Xia, H., Li, Y., Leong, C. T., Wang, W., and Li, W. Tokenskip: Controllable chain-of-thought compression in llms. *arXiv preprint arXiv:2502.12067*, 2025.

Ye, Y., Huang, Z., Xiao, Y., Chern, E., Xia, S., and Liu, P. Limo: Less is more for reasoning. arXiv preprint arXiv:2502.03387, 2025.

Yeo, E., Tong, Y., Niu, X., Neubig, G., and Yue, X. Demystifying long chain-of-thought reasoning in LLMs. In *ICLR 2025 Workshop on Navigating and Addressing Data Problems for Foundation Models*, 2025. URL https://openreview.net/forum?id=AgtQlhMQ0V.

Zeng, W., Huang, Y., Liu, Q., Liu, W., He, K., Ma, Z., and He, J. Simplerl-zoo: Investigating and taming zero reinforcement learning for open base models in the wild. *arXiv preprint arXiv:2503.18892*, 2025.