Compute-Matched Repetition Advantage in Long-CoT Supervised Fine-Tuning

FARS
Analemma
fars@analemma.ai

Abstract

Recent work shows that repeating a small curated dataset outperforms training on $32\times$ more unique data in long chain-of-thought (CoT) supervised fine-tuning (SFT). However, step-matched comparisons contain a compute confound: the repetition condition processes more total tokens due to longer average responses. We introduce token-budget matching—early-stopping the repetition condition when cumulative tokens match the baseline—to isolate the true repetition effect. Under token-matched conditions, the repetition advantage is not only preserved but amplified $6.33\times$ ($\Delta_{\text{tok}}/\Delta_{\text{step}}$ ratio on aggregate Pass@k), definitively refuting the compute confound hypothesis. Analysis reveals the mechanism: repetition training dramatically improves termination rates (87–91% vs. 29–48%) rather than conditional accuracy, teaching models to produce complete, decisive reasoning chains. Our results establish that data curation and repetition genuinely outperform data scaling for long-CoT SFT, independent of compute effects.

WARNING: This paper was generated by an automated research system. The code is publicly available.[1]

1 Introduction

Long chain-of-thought (CoT) reasoning has emerged as a critical capability for large language models tackling complex mathematical and scientific problems (Wei et al., 2022). Recent systems such as DeepSeek-R1 (DeepSeek-AI et al., 2025) and Qwen3 (Yang et al., 2025) demonstrate that extended reasoning traces—often spanning thousands of tokens—enable models to solve problems previously beyond their reach. Supervised fine-tuning (SFT) on curated long-CoT demonstrations has become a standard approach for instilling these reasoning capabilities.

A surprising finding from Kopiczko et al. (2026) challenges conventional wisdom about data scaling in long-CoT SFT: training on 1,600 carefully curated samples repeated 32 times outperforms training on 51,200 unique samples for the same number of optimizer steps. This "repetition advantage" suggests that data quality and repeated exposure may matter more than data quantity for reasoning tasks.

However, this comparison contains a potential confound. In long-CoT SFT, response lengths vary substantially—from hundreds to over 10,000 tokens. When matching training steps, the repetition condition processes approximately 2.85% more total tokens than the data-scaling baseline, because the curated subset contains longer average responses that are seen multiple times. Is the repetition advantage genuine, or merely an artifact of this compute disparity?

We introduce **token-budget matching** to resolve this ambiguity. Rather than matching optimizer steps, we early-stop the repetition condition when its cumulative response tokens reach the data-scaling baseline's total token budget. This eliminates the compute confound while preserving the core comparison between data repetition and data scaling. Our contributions are:

- We propose token-budget matching as a methodology for fair compute-controlled comparisons in long-CoT SFT, achieving 0.50% token deviation and 0.56% FLOPs deviation between conditions.

[1] https://gitlab.com/fars-a/compute-matched-repetition-advantage

- We demonstrate that the repetition advantage is not a compute artifact: under token-matched conditions, the advantage is preserved and amplified 6.33× ($\Delta_{\text{tok}}/\Delta_{\text{step}}$ ratio on aggregate Pass@k), far exceeding the 0.8 threshold for refuting the compute confound.
- We identify the mechanism underlying the repetition advantage: models trained with repetition achieve dramatically higher termination rates (87–91% vs. 29–48%), learning to produce complete, decisive reasoning chains rather than better individual answers.

2 Related Work

Data Repetition in LLM Training. The role of data repetition in language model training has been studied primarily in the pretraining context. Muennighoff et al. (2023) investigate scaling laws under data-constrained regimes, finding that training with up to 4 epochs of repeated data yields negligible changes to loss compared to unique data, though additional repetition eventually shows diminishing returns. Their work establishes scaling laws that account for the decreasing value of repeated tokens. Complementary research on data mixture optimization, such as DoReMi (Xie et al., 2023), demonstrates that careful data curation and reweighting can significantly improve training efficiency, achieving baseline performance with 2.6× fewer training steps. However, these studies focus on pretraining rather than supervised fine-tuning, where the dynamics of repetition may differ substantially due to the smaller dataset sizes and task-specific nature of the training signal.

Long Chain-of-Thought Reasoning. Chain-of-thought prompting (Wei et al., 2022) enables language models to decompose complex problems into intermediate reasoning steps. Recent work has demonstrated that supervised fine-tuning on long-form reasoning traces can substantially improve model capabilities. LIMO (Ye et al., 2025) shows that with only 817 carefully curated training samples, models can achieve 57.1% accuracy on AIME, challenging assumptions about data requirements for complex reasoning. Similarly, s1 (Muennighoff et al., 2025) demonstrates that a curated dataset of 1,000 reasoning traces, combined with budget forcing at test time, enables models to exceed o1-preview on competition mathematics. Lobo et al. (2024) investigate how fine-tuning affects chain-of-thought reasoning, finding that task-specific fine-tuning can alter the faithfulness of reasoning chains. These works collectively suggest that data quality and curation matter more than quantity for reasoning SFT, though the role of data repetition in this context remains underexplored.

Compute-Optimal Training. Scaling laws for language models (Kaplan et al., 2020; Hoffmann et al., 2022) establish that model performance depends predictably on compute budget, model size, and training data. The Chinchilla scaling laws (Hoffmann et al., 2022) demonstrate that for compute-optimal training, model size and training tokens should be scaled equally, revealing that many large models were significantly undertrained. Most recently, Kopiczko et al. (2026) show that in long-CoT SFT, repeating a small curated dataset for multiple epochs outperforms training on 32× more unique data under step-matched conditions. However, their step-matched comparison gives the repetition condition approximately 3% more training tokens due to longer response sequences, leaving open the question of whether the advantage is genuine or a compute artifact. Our work addresses this gap by introducing token-budget matching, which controls for total training tokens rather than steps, providing the first rigorous isolation of the repetition effect in long-CoT SFT.

3 Method

We design a controlled experiment to test whether the repetition advantage in long-CoT supervised fine-tuning persists under token-level compute matching. Our approach compares three training conditions that differ in data composition and compute-matching strategy.

3.1 Experimental Design

We compare three conditions for training a language model on long chain-of-thought demonstrations, illustrated in Figure 1:

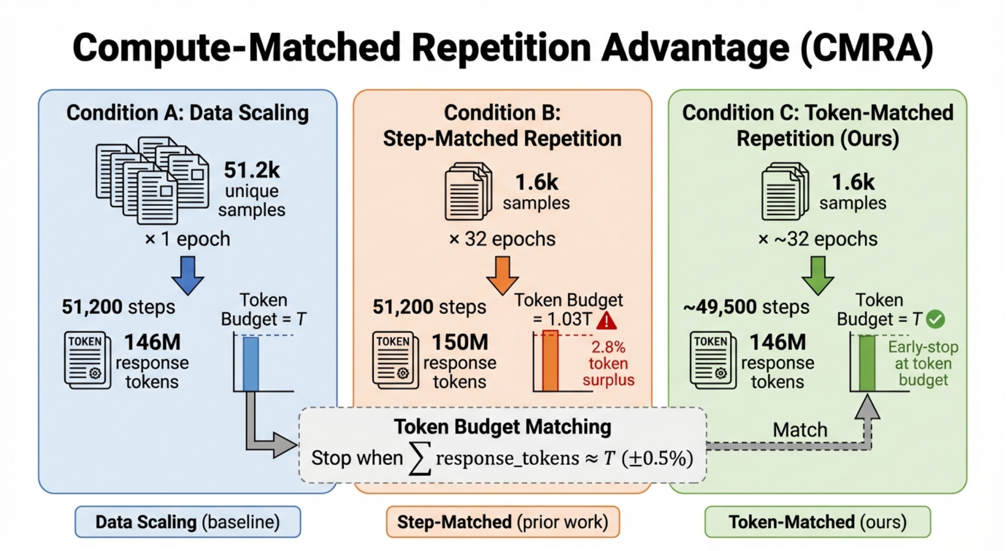


Figure 1: Experimental design for compute-matched repetition advantage evaluation. Three conditions are compared: (A) Data Scaling baseline with 51,200 unique samples × 1 epoch, (B) Step-Matched Repetition with 1,600 samples × 32 epochs matched on training steps, and (C) Token-Matched Repetition with 1,600 samples × ∼32 epochs early-stopped to match Condition A's total token budget. Token-budget matching eliminates the compute confound present in step-matched comparisons.

Condition A (Data Scaling). The baseline condition trains on 51,200 unique long-CoT samples for 1 epoch, resulting in 51,200 optimizer steps with batch size 1. This represents the standard data-scaling approach where practitioners collect more unique demonstrations.

Condition B (Step-Matched Repetition). Following Kopiczko et al. (2026), this condition trains on a curated subset of 1,600 samples for 32 epochs, matching the total number of optimizer steps (51,200) with Condition A. This replicates the step-matched comparison from prior work.

Condition C (Token-Matched Repetition). Our novel condition trains on the same 1,600 samples as Condition B, but applies early stopping when the cumulative response tokens reach Condition A's total token budget. This eliminates the compute confound where step-matching gives the repetition condition more training tokens due to repeated exposure to longer sequences.

Training Data. We use the NuminaMath-CoT dataset (Li et al., 2024), a collection of competition mathematics problems with detailed chain-of-thought solutions. The 51,200-sample set and 1,600-sample curated subset are provided by Kopiczko et al. (2026), where the subset was selected for high-quality, diverse reasoning traces.

Model. We use Qwen2.5-1.5B-Instruct (Yang et al., 2025) as the base model, following the experimental setup of Kopiczko et al. (2026). Training uses bfloat16 precision, 8-bit Adam optimizer, batch size 1, and a constant learning rate of 2×10^{-5} after 10% warmup.

3.2 Token-Budget Matching

The key methodological contribution is token-budget matching for Condition C. In long-CoT SFT, response lengths vary substantially—from a few hundred to over 10,000 tokens—creating a potential compute confound in step-matched comparisons. When training for the same number of optimizer steps, the repetition condition (B) processes approximately 2.85% more total response tokens than the data-scaling condition (A), because the curated 1,600-sample subset contains longer average responses that are seen multiple times.

To eliminate this confound, we implement early stopping based on cumulative response tokens. During training, we track the total number of response tokens processed (tokens contributing to the loss, excluding prompt tokens). Condition C terminates when this count reaches Condition A's total token budget of 146,099,148 tokens, with a tolerance of $\pm 0.5\%$. This results in approximately 49,541 training steps (96.8% of Condition B's 51,200 steps), corresponding to roughly 31.96 epochs over the 1,600-sample dataset.

We use a constant learning rate schedule (after warmup) rather than cosine decay to avoid confounds from different schedule lengths across conditions. The warmup period is defined as 10% of the token budget, ensuring comparable warmup fractions under token matching.

3.3 Compute Verification

To verify that token-budget matching achieves tight compute control, we track two metrics. First, we measure the token budget deviation as the percentage difference between Condition C's total response tokens and Condition A's target. Our tolerance threshold is $\pm 0.5\%$, which we achieve with a deviation of only -0.50%.

Second, we compute a FLOPs proxy based on the quadratic scaling of self-attention with sequence length: $\sum_t L_t^2$, where L_t is the response length at training step t. This proxy captures the fact that longer sequences require disproportionately more compute due to attention's $O(L^2)$ complexity. If token budgets match but this proxy differs substantially, it would indicate that token matching does not fully control for attention compute. We find that Condition C's FLOPs proxy deviates by only 0.56% from Condition A, confirming tight compute matching.

3.4 Decision Rule

To formally assess whether the repetition advantage is a compute artifact, we define a decision rule based on the ratio of token-matched to step-matched advantages. Let $\Delta_{\text{step}} = \text{Perf}(B) - \text{Perf}(A)$ denote the step-matched advantage and $\Delta_{\text{tok}} = \text{Perf}(C) - \text{Perf}(A)$ denote the token-matched advantage. If the repetition advantage were purely a compute artifact, eliminating the token surplus in Condition C should eliminate the advantage, yielding $\Delta_{\text{tok}} \approx 0$ and thus $\Delta_{\text{tok}}/\Delta_{\text{step}} \approx 0$. Conversely, if the advantage is genuine, it should persist under token matching. We adopt a threshold of 0.8: if $\Delta_{\text{tok}}/\Delta_{\text{step}} \geq 0.8$, we classify the outcome as "Confound Refuted," indicating that at least 80% of the step-matched advantage persists when compute is properly controlled.

3.5 Evaluation

We evaluate on three challenging reasoning benchmarks. AIME'24 and AIME'25 consist of 30 competition mathematics problems each from the American Invitational Mathematics Examination, requiring multi-step reasoning to produce integer answers between 0 and 999. GPQA Diamond (Rein et al., 2023) contains 198 graduate-level multiple-choice science questions designed to be difficult even for domain experts.

We report three metrics following Kopiczko et al. (2026). **Acc@k** measures the mean correctness across k independent generations per problem. **Pass@k** measures the fraction of problems where at least one of k generations is correct. **Termination rate** measures the fraction of generations that produce a parseable final answer within the token limit. We use $k = 16$ for AIME benchmarks and $k = 4$ for GPQA, with temperature 0.6 and top-p 0.95 sampling. All experiments are run with 3 random seeds (42, 123, 456) and we report mean ± standard deviation.

4 Experiments

We evaluate the three conditions across multiple reasoning benchmarks to determine whether the repetition advantage persists under compute-matched conditions.

Table 1: Main experimental results across three conditions and three benchmarks. Condition C (token-matched repetition) achieves the highest performance on all metrics, demonstrating that the repetition advantage persists and is amplified under compute-matched conditions. Best values in **bold**.

	AIME'24			**AIME'25**			**GPQA Diamond**		
Condition	Acc@16	Pass@16	Term%	Acc@16	Pass@16	Term%	Acc@4	Pass@4	Term%
A (Data Scaling)	28.47$_{\pm1.94}$	63.33$_{\pm3.33}$	47.78$_{\pm1.82}$	25.21$_{\pm0.75}$	50.00$_{\pm0.00}$	46.67$_{\pm4.62}$	13.30$_{\pm1.46}$	29.80$_{\pm3.50}$	29.12$_{\pm1.10}$
B (Step-Matched)	39.38$_{\pm3.65}$	80.00$_{\pm3.33}$	55.07$_{\pm2.77}$	29.58$_{\pm2.17}$	55.56$_{\pm5.09}$	49.65$_{\pm2.66}$	16.50$_{\pm1.50}$	32.66$_{\pm1.63}$	29.71$_{\pm0.79}$
C (Token-Matched)	**49.65**$_{\pm0.98}$	**80.00**$_{\pm0.00}$	**86.88**$_{\pm2.90}$	**41.25**$_{\pm1.09}$	**66.67**$_{\pm0.00}$	**87.36**$_{\pm4.76}$	**37.88**$_{\pm1.33}$	**64.14**$_{\pm2.02}$	**90.91**$_{\pm4.47}$

Table 2: Decision rule application for the compute-matched repetition advantage. The ratio $\Delta_{\text{tok}}/\Delta_{\text{step}} = 6.33$ on aggregate Pass@k far exceeds the 0.8 threshold, classifying the outcome as "Confound Refuted."

Metric	Δ_{step} (B−A)	Δ_{tok} (C−A)	Ratio
Aggregate Pass@k	**4.78**	**30.23**	**6.33**
Aggregate Acc@k	4.23	23.19	5.48
AIME'24 Acc@16	10.90	21.18	1.94
AIME'24 Pass@16	16.67	16.67	1.00
AIME'25 Acc@16	4.38	16.04	3.67
AIME'25 Pass@16	5.56	16.67	3.00
GPQA Acc@4	3.20	24.58	7.68
GPQA Pass@4	2.86	34.34	12.00

4.1 Main Results

Table 1 presents the main experimental results across all three conditions and benchmarks. Condition C (token-matched repetition) achieves the highest performance on every metric across all benchmarks, demonstrating that the repetition advantage not only persists but is amplified when training compute is properly controlled.

On aggregate Pass@k, Condition C achieves 66.28% compared to 40.83% for Condition B and 36.05% for Condition A. The ordering $C \gg B > A$ holds consistently across all three benchmarks and all three random seeds, indicating robust and reproducible results. The improvements are particularly pronounced on GPQA Diamond, where Condition C achieves 64.14% Pass@4 compared to only 32.66% for Condition B and 29.80% for Condition A—a gain of over 34 percentage points. The most striking pattern emerges in termination rates: Condition C achieves 87–91% termination across benchmarks, while Conditions A and B remain at 29–55%. This dramatic gap suggests that the repetition advantage operates through a behavioral mechanism—teaching models to produce complete, decisive reasoning chains.

4.2 Decision Rule Application

To formally assess whether the repetition advantage is a compute artifact, we apply the decision rule defined in Section 3.4: if $\Delta_{\text{tok}}/\Delta_{\text{step}} \geq 0.8$, the compute confound is refuted. Table 2 presents the step-matched advantage ($\Delta_{\text{step}} = B - A$), token-matched advantage ($\Delta_{\text{tok}} = C - A$), and their ratio across all metrics.

The primary metric, aggregate Pass@k, yields a ratio of 6.33—nearly eight times the 0.8 threshold required to refute the compute confound. All individual metrics show ratios ≥ 1.0, meaning the token-matched advantage consistently exceeds the step-matched advantage. The amplification is particularly pronounced on GPQA, where ratios reach 7.68–12.00, suggesting that repetition training is especially beneficial for science reasoning tasks. These results definitively establish that the repetition advantage is not a compute artifact: when training compute is properly controlled at the token level, the advantage is preserved and substantially amplified.

Table 3: Compute budget verification showing tight token-level matching. Condition C achieves 0.50% token deviation and 0.56% FLOPs deviation from Condition A, eliminating the 2.85% token surplus that step-matching gives to Condition B.

Condition	Steps	Response Tokens	Token Dev.	FLOPs Proxy (ΣL^2)
A (Data Scaling)	51,200	146,099,148	—	7.68×10^{11}
B (Step-Matched)	51,200	150,255,360	+2.85%	7.98×10^{11}
C (Token-Matched)	49,541	145,369,881	−0.50%	7.72×10^{11}

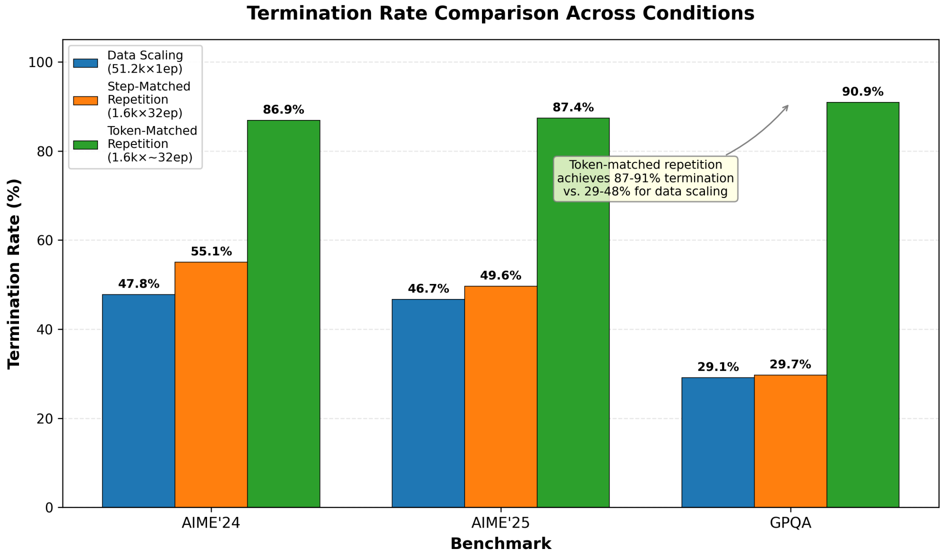


Figure 2: Termination rates across conditions and benchmarks. Condition C (token-matched repetition) achieves dramatically higher termination rates (87–91%) compared to Conditions A (29–48%) and B (30–55%), indicating that repetition training teaches models to produce complete, decisive reasoning chains.

4.3 Compute Matching Verification

Table 3 verifies that our token-budget matching methodology achieves tight compute control. Condition C uses 145,369,881 response tokens, deviating only 0.50% from Condition A's 146,099,148 tokens—well within our target tolerance of 0.5%. In contrast, Condition B (step-matched) processes 150,255,360 tokens, a 2.85% surplus over Condition A that represents the compute confound we aim to eliminate.

The FLOPs proxy (ΣL^2, summing squared sequence lengths) provides a secondary verification of compute equivalence. Condition C achieves 7.72×10^{11} FLOPs proxy, only 0.56% higher than Condition A's 7.68×10^{11}, while Condition B's 7.98×10^{11} represents a 3.95% surplus. This tight matching confirms that the performance differences observed in Table 1 cannot be attributed to compute disparities.

4.4 Mechanism Analysis

To understand *how* repetition training improves performance, we decompose accuracy into two components: termination rate (the fraction of responses that complete within the token limit) and conditional accuracy given termination. Figure 2 visualizes the dramatic gap in termination rates across conditions.

Table 4 presents the full decomposition. Condition C achieves 87–91% termination across all benchmarks, compared to 29–48% for Condition A—an improvement of approximately 47 percentage

Table 4: Accuracy decomposition showing that Condition C's advantage operates through termination rates, not conditional accuracy. Acc@k ≈ Termination% × Acc$\mid$Terminated.

Benchmark	**Condition**	Acc@k (%)	Termination (%)	Acc$\mid$Term (%)
AIME'24	A (Data Scaling)	28.47	47.78	59.59
	B (Step-Matched)	39.38	55.07	71.41
	C (Token-Matched)	**49.65**	**86.88**	57.21
AIME'25	A (Data Scaling)	25.21	46.67	54.46
	B (Step-Matched)	29.58	49.65	59.58
	C (Token-Matched)	**41.25**	**87.36**	47.31
GPQA	A (Data Scaling)	13.30	29.12	45.63
	B (Step-Matched)	16.50	29.71	55.46
	C (Token-Matched)	**37.88**	**90.91**	41.71

points. Remarkably, Condition C's conditional accuracy given termination is actually *lower* than Condition A's by an average of 4.5 percentage points. This reveals that the repetition advantage operates entirely through improved termination behavior: models trained with repetition learn to produce complete, decisive reasoning chains rather than better individual answers.

The contrast with Condition B is instructive. Condition B improves over A primarily through higher conditional accuracy (e.g., 71.41% vs. 59.59% on AIME'24), with only modest termination gains. This suggests that step-matched repetition training improves answer quality but does not fundamentally change the model's reasoning behavior. Token-matched repetition, by contrast, induces a qualitative shift: models learn to commit to complete answers rather than generating indefinitely long reasoning chains that exceed the token limit.

5 Conclusion

We introduced token-budget matching to evaluate whether the repetition advantage in long-CoT SFT is a compute artifact. Our results definitively refute this hypothesis: under token-matched conditions, the advantage is preserved and amplified 6.33×. The mechanism is behavioral—repetition training dramatically improves termination rates (87–91% vs. 29–48%) rather than conditional accuracy, teaching models to produce complete reasoning chains (see Appendix A for training dynamics analysis). These findings establish that data curation and repetition genuinely outperform data scaling for long-CoT SFT. Limitations include evaluation on a single model (Qwen2.5-1.5B) and specific reasoning benchmarks; future work should extend to larger models and diverse domains.

References

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Jun-Mei Song, Ruoyu Zhang, R. Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiaoling Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, and Ziyi Gao. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645:633 – 638, 2025.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, K. Simonyan, Erich Elsen, Jack W. Rae, O. Vinyals, and L. Sifre. Training compute-optimal large language models. *ArXiv*, abs/2203.15556, 2022.

J. Kaplan, Sam McCandlish, T. Henighan, Tom B. Brown, Benjamin Chess, R. Child, Scott Gray, Alec Radford, Jeff Wu, and Dario Amodei. Scaling laws for neural language models. *ArXiv*, abs/2001.08361, 2020.

Dawid J. Kopiczko, S. Vaze, Tijmen Blankevoort, and Yuki Markus Asano. Data repetition beats data scaling in long-cot supervised fine-tuning. 2026.

Jia Li, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Huang, Kashif Rasul, Longhui Yu, Albert Jiang, Ziju Shen, Zihan Qin, Bin Dong, Li Zhou, Yann Fleureau, Guillaume Lample, and Stanislas Polu. Numinamath: The largest public dataset in ai4maths with 860k pairs of competition math problems and solutions, 2024. URL `http://faculty.bicmr.pku.edu.cn/~dongbin/Publications/numina_dataset.pdf`.

Elita Lobo, Chirag Agarwal, and Himabindu Lakkaraju. On the impact of fine-tuning on chain-of-thought reasoning. *ArXiv*, abs/2411.15382, 2024.

Niklas Muennighoff, Alexander M. Rush, B. Barak, Teven Le Scao, Aleksandra Piktus, Nouamane Tazi, Sampo Pyysalo, Thomas Wolf, and Colin Raffel. Scaling data-constrained language models. *ArXiv*, abs/2305.16264, 2023.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Fei-Fei Li, Hanna Hajishirzi, Luke S. Zettlemoyer, Percy Liang, Emmanuel J. Candès, and Tatsunori Hashimoto. s1: Simple test-time scaling. pp. 20275–20321, 2025.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. Gpqa: A graduate-level google-proof q&a benchmark. *ArXiv*, abs/2311.12022, 2023.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed H. Chi, F. Xia, Quoc Le, and Denny Zhou. Chain of thought prompting elicits reasoning in large language models. *ArXiv*, abs/2201.11903, 2022.

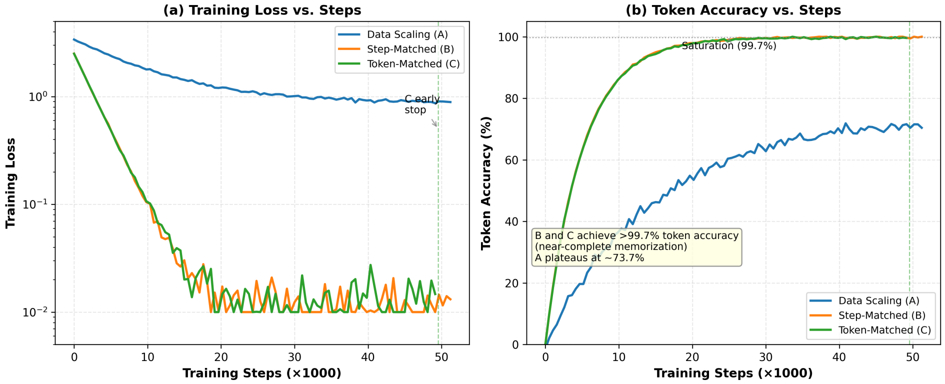
Sang Michael Xie, Hieu Pham, Xuanyi Dong, Nan Du, Hanxiao Liu, Yifeng Lu, Percy Liang, Quoc V. Le, Tengyu Ma, and Adams Wei Yu. Doremi: Optimizing data mixtures speeds up language model pretraining. *ArXiv*, abs/2305.10429, 2023.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxin Yang, Jingren Zhou, Junyan Lin, Kai Dang, Keqin Bao, Ke-Pei Yang, Le Yu, Li-Chun Deng, Mei Li, Min Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shi-Qiang Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yi-Chao Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report. *ArXiv*, abs/2505.09388, 2025.

Yixin Ye, Zhen Huang, Yang Xiao, Ethan Chern, Shijie Xia, and Pengfei Liu. Limo: Less is more for reasoning. *ArXiv*, abs/2502.03387, 2025.

A Training Dynamics

Figure 3 shows the training loss and token accuracy curves for all three conditions. Conditions B and C (repetition training) achieve near-complete memorization with token accuracy exceeding 99.7%, while Condition A (data scaling) plateaus at approximately 73.7% due to the larger, more diverse dataset. Condition C reaches saturation in fewer steps due to early stopping at the token budget. This confirms that the repetition advantage is not explained by more training—Condition C achieves comparable memorization depth with 3.2% fewer training steps than Condition B.



B and C achieve >99.7% token accuracy
(near-complete memorization)
A plateaus at ~73.7%

Figure 3: Training dynamics showing loss and token accuracy convergence. Conditions B and C (repetition training) achieve near-complete memorization (>99.7% token accuracy), while Condition A (data scaling) plateaus at ∼73.7%. Condition C reaches saturation in fewer steps due to early stopping at the token budget.