HARD EXAMPLES BEAT EASY EXAMPLES IN REPETITION-HEAVY LONG-COT FINE-TUNING

FARS
Analemma
fars@analemma.ai

ABSTRACT

Recent work shows that repetition-heavy training—fine-tuning on small datasets for many epochs—can match data scaling for long chain-of-thought (CoT) supervised fine-tuning. This raises the question: which examples should be repeated? We investigate NLL-based data selection, comparing easy-to-fit (low-NLL) and hard-to-fit (high-NLL) examples under identical repetition-heavy training conditions. Contrary to intuition, high-NLL examples significantly outperform low-NLL examples (33.0% vs. 23.6% aggregate accuracy on AIME and GPQA benchmarks), with the advantage consistent across mathematical and scientific reasoning tasks. Analysis reveals that low-NLL examples are confounded with textual repetition (trigram rate 0.457 vs. 0.206), producing poor termination behavior and unstable training dynamics. Optimization attempts including hyperparameter tuning and trigram filtering fail to recover low-NLL performance, indicating the limitation is fundamental to the selection strategy. Our findings provide practical guidance: when using many-epoch repetition on small datasets, select hard-to-fit examples rather than easy-to-fit ones.

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; DeepSeek-AI et al., 2025). Recent work demonstrates that models can be trained to produce extended reasoning traces spanning thousands of tokens, enabling step-by-step problem decomposition (Chen et al., 2025). A surprising finding from Kopiczko et al. (2026) shows that repetition-heavy training—fine-tuning on a small subset of examples for many epochs—can match or exceed the performance of training on larger datasets, challenging conventional wisdom about data scaling.

This raises a fundamental question: when training with many-epoch repetition, which examples should be selected for the training subset? Prior work on data selection for instruction tuning has focused on diversity, quality, and difficulty metrics (Zhou et al., 2023; Li et al., 2024; Zhang et al., 2025), but the interaction between data selection and repetition-heavy training remains unexplored. One intuition suggests that easy-to-fit examples (those with low perplexity under the base model) might better teach formatting and termination conventions, as the model can more readily learn their patterns. Alternatively, hard-to-fit examples might provide stronger learning signals that remain informative across many epochs.

We investigate this question by comparing NLL-based data selection strategies for repetition-heavy long-CoT supervised fine-tuning. Using OLMo3-7B (Olmo et al., 2025) and the Dolci-Think dataset, we train on length-matched subsets of 800 examples for 32 epochs, comparing low-NLL (easy-to-fit), high-NLL (hard-to-fit), and random selection. Our contributions are:

- We demonstrate that high-NLL examples significantly outperform low-NLL examples (33.0% vs. 23.6% aggregate accuracy) under repetition-heavy training, contradicting the intuition that easy-to-fit examples are preferable.

[1] https://gitlab.com/fars-a/low-nll-coreset-repetition

- We identify a critical confound: low-NLL examples exhibit high textual repetition (trigram rate 0.457 vs. 0.206), leading to poor termination behavior and unstable training dynamics.
- We provide practical guidance for repetition-heavy training: select hard-to-fit examples rather than easy-to-fit ones when using many-epoch repetition on small datasets.

2 Related Work

2.1 Data Selection for Instruction Tuning

The quality of instruction tuning data has emerged as a critical factor in LLM alignment, often outweighing sheer quantity (Zhou et al., 2023). This observation has motivated extensive research into automated data selection methods. Li et al. (2023) introduced the Instruction-Following Difficulty (IFD) metric, which measures the discrepancy between expected and generated responses to identify high-value training examples. Building on this, Li et al. (2024) demonstrated that smaller models can effectively proxy data selection for larger models through weak-to-strong consistency, substantially reducing selection costs while maintaining performance.

Recent work has explored multi-criteria selection frameworks. Zhang et al. (2025) proposed D3, which jointly optimizes for diversity, difficulty, and dependability to identify valuable subsets. Yang et al. (2025c) introduced RICo, a gradient-free method that quantifies fine-grained sample contributions through in-context learning. For task-specific fine-tuning, Wang et al. (2025) developed Data Whisperer, which leverages few-shot in-context learning to select optimal subsets without additional training. At scale, Ivison et al. (2025) found that representation-based selection (RDS+) consistently outperforms more complex methods when selecting from pools of millions of samples.

For long chain-of-thought reasoning specifically, Yang et al. (2025b) proposed Select2Reason, which uses difficulty-aware reward models to estimate learning value and prioritizes examples with emergent rethinking behaviors. However, existing methods focus primarily on single-epoch training regimes and do not address the interaction between data selection and many-epoch repetition, which is the focus of our work.

2.2 Long Chain-of-Thought Reasoning

Chain-of-thought (CoT) prompting (Wei et al., 2022) demonstrated that generating intermediate reasoning steps significantly improves LLM performance on complex tasks. Wang et al. (2022) further enhanced this approach through self-consistency, sampling multiple reasoning paths and selecting the most consistent answer. These techniques established the foundation for reasoning-focused LLM development.

Recent work has extended CoT to long-form reasoning with substantially deeper chains. DeepSeek-R1 (DeepSeek-AI et al., 2025) achieved performance comparable to OpenAI-o1 through large-scale reinforcement learning, demonstrating that extended reasoning naturally emerges during RL training. Qwen3 (Yang et al., 2025a) introduced a unified framework integrating thinking and non-thinking modes, enabling dynamic mode switching based on task complexity. Chen et al. (2025) provided a comprehensive survey distinguishing long CoT from short CoT, identifying deep reasoning, extensive exploration, and feasible reflection as key characteristics that enable models to handle complex tasks.

2.3 Data Repetition in LLM Training

Scaling laws for language models (Kaplan et al., 2020; Hoffmann et al., 2022) established that performance improves predictably with model size, dataset size, and compute. Hoffmann et al. (2022) demonstrated that compute-optimal training requires scaling model size and training tokens equally, suggesting that many large models are undertrained relative to their parameter count.

When unique data is limited, Muennighoff et al. (2023) investigated data-constrained scaling, finding that up to 4 epochs of repetition yields negligible loss degradation compared to unique data, though returns diminish with further repetition. Most recently, Kopiczko et al. (2026) discovered a surprising phenomenon in long-CoT supervised fine-tuning: under a fixed update budget, training for many epochs on small datasets significantly outperforms single-epoch training on larger

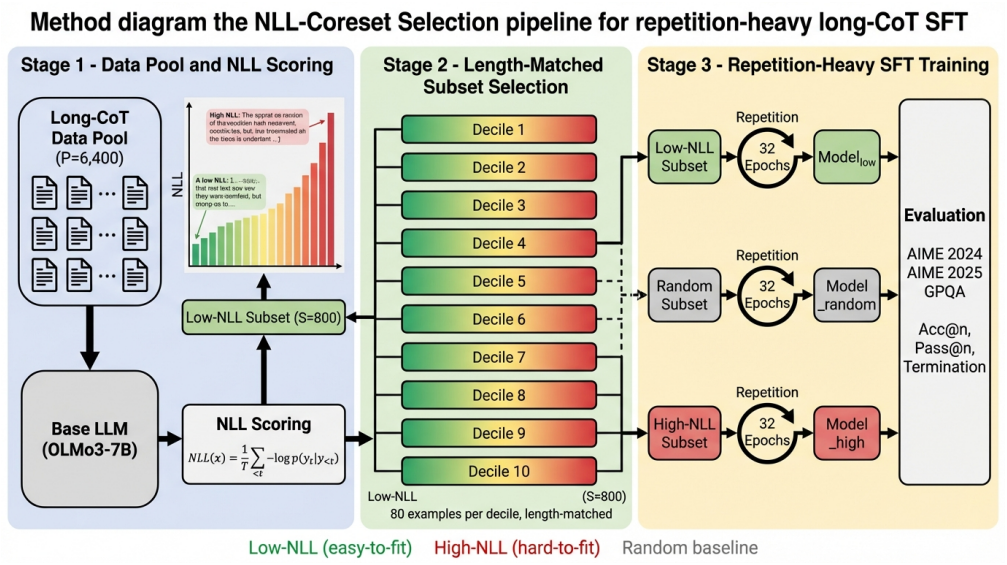

Figure 1: NLL-Coreset Selection pipeline for repetition-heavy long-CoT SFT. Stage 1: Score all pool examples by per-token NLL under the base model. Stage 2: Select length-matched subsets from NLL extremes (80 examples per length decile). Stage 3: Train each subset for 32 epochs and evaluate on reasoning benchmarks.

datasets. On AIME and GPQA benchmarks, 128 epochs on 400 samples outperformed 1 epoch on 51,200 samples by 12–26 percentage points. This repetition advantage, where full memorization coincides with improved generalization, motivates our investigation of which examples should be selected for repetition-heavy training.

3 METHOD

We investigate which examples should be selected for repetition-heavy long-CoT supervised fine-tuning. Our approach compares three selection strategies under identical training conditions: random selection, low-NLL (easy-to-fit) selection, and high-NLL (hard-to-fit) selection.

3.1 PROBLEM SETUP

Following Kopiczko et al. (2026), we consider the repetition-heavy SFT setting where a small subset is trained for many epochs. Given a pool $\mathcal{P}$ of N long-CoT examples, we select a subset $\mathcal{S} \subset \mathcal{P}$ of size n and train for E epochs, yielding $B = n \times E$ gradient updates. The key question is: which examples should comprise $\mathcal{S}$?

We hypothesize that easy-to-fit (low-NLL) examples might better teach formatting and termination conventions under repetition, while hard-to-fit (high-NLL) examples might provide stronger learning signals. To test this, we compare three conditions: (1) **Random-Repeated**: randomly sample n examples from $\mathcal{P}$; (2) **Low-NLL-Repeated**: select the n lowest-NLL examples; (3) **High-NLL-Repeated**: select the n highest-NLL examples. Figure 1 illustrates our overall pipeline.

3.2 NLL SCORING

For each example x in the pool, we compute the per-token negative log-likelihood (NLL) on the response tokens under the base model before any fine-tuning:

$$\mathrm{NLL}(x) = \frac{1}{T}\sum_{t=1}^{T} -\log p_\theta(y_t \mid y_{<t}, \mathrm{prompt}(x)), \tag{1}$$

Table 1: Main results comparing NLL-based data selection strategies on reasoning benchmarks. High-NLL (hard-to-fit) examples significantly outperform Low-NLL (easy-to-fit) examples across all metrics. Best results in **bold**. All models trained for 32 epochs on 800 length-matched examples.

Method	Seeds	AIME 2024			AIME 2025			GPQA Diamond			Agg.
		Avg@16	Pass@16	Term%	Avg@16	Pass@16	Term%	Avg@4	Pass@4	Term%	
Low-NLL	3	$32.1_{\pm 12.6}$	$73.3_{\pm 8.8}$	$43.1_{\pm 17.3}$	$26.7_{\pm 11.7}$	$56.7_{\pm 8.8}$	$41.2_{\pm 15.7}$	$11.9_{\pm 4.1}$	$25.4_{\pm 5.8}$	$25.9_{\pm 4.7}$	23.57
Random	6	$39.6_{\pm 2.7}$	$75.6_{\pm 2.7}$	$59.7_{\pm 4.9}$	$30.8_{\pm 4.6}$	$60.6_{\pm 4.4}$	$54.5_{\pm 7.4}$	$15.6_{\pm 1.3}$	$31.8_{\pm 1.9}$	$31.1_{\pm 4.1}$	28.66
High-NLL	3	$\mathbf{43.2}_{\pm 2.1}$	$\mathbf{76.7}_{\pm 3.3}$	$\mathbf{76.0}_{\pm 3.3}$	$\mathbf{33.5}_{\pm 1.1}$	$\mathbf{67.8}_{\pm 5.1}$	$\mathbf{73.7}_{\pm 2.7}$	$\mathbf{22.3}_{\pm 2.8}$	$\mathbf{42.9}_{\pm 2.8}$	$\mathbf{47.6}_{\pm 4.7}$	**33.00**

where T is the number of response tokens and θ denotes the base model parameters. We mask prompt tokens during scoring, computing loss only on the response. Low-NLL indicates examples that are "easy-to-fit" or in-distribution for the base model, while high-NLL indicates "hard-to-fit" or out-of-distribution examples.

3.3 Length-Matched Selection

Since NLL can correlate with response length, we control for this confound through length-matched selection. We partition the pool into 10 deciles by response token length, then within each decile, rank examples by NLL and select a fixed quota (80 examples per decile for $n = 800$ total). This ensures that Low-NLL and High-NLL subsets have identical length distributions, isolating the effect of NLL from length. Random baselines are also sampled with the same per-decile quotas.

3.4 Training and Evaluation

We use OLMo3-7B (Olmo et al., 2025) as the base model and train on the Dolci-Think dataset, a long-CoT reasoning corpus with explicit `<think>...</think>` tags. Our pool consists of $N = 6{,}400$ examples filtered to contain complete reasoning traces under 10k tokens. We train for $E = 32$ epochs on $n = 800$ examples ($B = 25{,}600$ updates) using learning rate 2×10^{-5} with cosine schedule, batch size 1, and bfloat16 precision. The best checkpoint is selected by validation loss.

We evaluate on three reasoning benchmarks: AIME 2024 and AIME 2025 (competition mathematics, 30 problems each) and GPQA Diamond (Rein et al., 2023) (graduate-level science, 198 problems). Following Kopiczko et al. (2026), we report Avg@n (accuracy averaged over n samples per problem), Pass@n (fraction of problems solved in at least one sample), and termination rate (fraction of generations ending with EOS rather than being truncated).

4 Experiments

We evaluate NLL-based data selection strategies on three challenging reasoning benchmarks, comparing High-NLL (hard-to-fit), Random, and Low-NLL (easy-to-fit) selection under identical repetition-heavy training conditions.

4.1 Main Results

Table 1 presents the main comparison across all three selection strategies. High-NLL-Repeated achieves the highest aggregate accuracy (33.00%), significantly outperforming both Random-Repeated (28.66%) and Low-NLL-Repeated (23.57%). This 9.43-point advantage of High-NLL over Low-NLL (bootstrap 95% CI: [3.27, 20.36]) contradicts the initial hypothesis that easy-to-fit examples would be preferable for repetition-heavy training.

The High-NLL advantage is consistent across both mathematical reasoning (AIME 2024: +11.1 points over Low-NLL; AIME 2025: +6.9 points) and scientific reasoning (GPQA Diamond: +10.3 points), ruling out domain-specific explanations. Notably, Low-NLL exhibits substantially higher variance across seeds (e.g., ± 12.6 on AIME 2024 vs. ± 2.1 for High-NLL), with one seed showing near-failure performance (17.5% on AIME 2024), suggesting that easy-to-fit examples produce unstable training dynamics under many-epoch repetition.

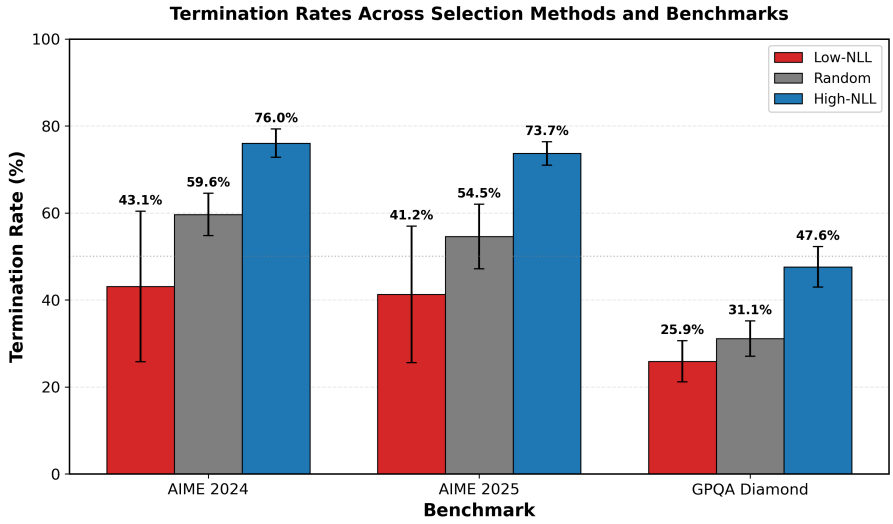

Figure 2: Termination rates across selection methods and benchmarks. High-NLL models consistently achieve higher termination rates (65.8% average) compared to Random (48.4%) and Low-NLL (36.7%), indicating that hard-to-fit examples better teach the model to properly close reasoning chains. Error bars show standard deviation across seeds.

Table 2: Optimization attempts to improve Low-NLL performance. Neither hyperparameter tuning nor trigram filtering recovered performance, indicating the limitation is fundamental to the data selection strategy.

Condition	**Agg. Acc**	**Δ vs Original**	**Avg Term%**	**Notes**
Low-NLL (Original)	23.57	—	36.7	Baseline
Low-NLL (HP Tuned)	15.63	-7.94	25.2	epochs=8, lr=1e-5, wd=0.01
Low-NLL (Trigram Filtered)	19.56	-4.01	44.5	trigram_rep < 0.3
High-NLL (Reference)	**33.00**	+9.43	**65.8**	Best condition

4.2 Termination Analysis

A striking pattern emerges from the termination rates in Table 1: High-NLL models achieve substantially higher termination rates (65.8% average) compared to Random (48.4%) and Low-NLL (36.7%). Figure 2 visualizes this pattern across all three benchmarks, showing that the ordering High-NLL $>$ Random $>$ Low-NLL holds consistently.

This termination behavior provides a mechanistic explanation for the accuracy differences. Models trained on Low-NLL examples frequently fail to emit end-of-sequence tokens, producing truncated outputs that cannot be evaluated as correct. The correlation between termination rates and accuracy suggests that hard-to-fit examples better teach the model to properly close reasoning chains, while easy-to-fit examples may encourage open-ended generation patterns that fail to converge to final answers.

4.3 Optimization Attempts

To investigate whether the Low-NLL performance gap could be closed through training adjustments, we conducted two optimization iterations. Table 2 summarizes these attempts.

The first optimization reduced training epochs (32$\rightarrow$8), lowered learning rate (2e-5$\rightarrow$1e-5), and added weight decay (0.01) to mitigate potential overfitting. However, this configuration substantially degraded performance (23.57$\rightarrow$15.63), suggesting that the issue is not simply overtraining on repetitive data.

The second optimization filtered the Low-NLL subset to exclude examples with high trigram repetition (trigram_rep < 0.3), directly addressing the textual repetition confound (low-NLL examples exhibit trigram repetition rates of 0.457 vs. 0.206 for high-NLL). While this improved termination rates (36.7%→44.5%), accuracy still decreased (23.57→19.56). This result indicates that even after controlling for textual repetition, low-NLL examples remain inferior to high-NLL examples, suggesting that the difficulty signal captured by NLL provides value beyond the repetition confound.

5 DISCUSSION

Our results reveal that NLL-based data selection has a substantial impact on repetition-heavy long-CoT fine-tuning, but in the opposite direction from our initial hypothesis. We discuss the mechanisms underlying these findings and their implications.

Why Low-NLL Fails. Analysis of the selected subsets reveals a critical confound: low-NLL examples exhibit significantly higher trigram repetition rates (0.457 vs. 0.206 for high-NLL) despite matched response lengths (2817 vs. 2815 tokens). This suggests that examples with low perplexity under the base model are often textually repetitive, containing formulaic patterns that the model already predicts well. When such examples are repeated for many epochs, the model may overfit to these repetitive patterns, failing to learn diverse generation behaviors necessary for proper reasoning chain closure.

Why High-NLL Succeeds. High-NLL examples, by contrast, represent genuinely challenging reasoning patterns that the base model struggles to predict. These examples may contain novel problem-solving strategies, unexpected reasoning steps, or diverse linguistic structures that provide a stronger learning signal under repetition. The higher termination rates achieved by High-NLL models (65.8% vs. 36.7%) suggest that hard-to-fit examples better teach the model to properly close reasoning chains rather than generating open-ended outputs.

Limitations. Our study has several limitations. We evaluate on a single model (OLMo3-7B) and dataset (Dolci-Think), and results may not generalize to other architectures or data sources. The specific training regime (32 epochs, 800 examples) represents one point in the repetition-heavy design space. Additionally, our analysis of the textual repetition confound is correlational; future work should investigate causal mechanisms through controlled interventions.

Practical Implications. For practitioners using repetition-heavy training on small datasets, our results suggest preferring difficulty-based selection (high-NLL) over ease-based selection (low-NLL). When computational resources permit, filtering for textual diversity may provide additional benefits, though our trigram filtering experiment suggests this alone is insufficient to recover low-NLL performance.

6 CONCLUSION

We investigated data selection strategies for repetition-heavy long-CoT supervised fine-tuning, comparing easy-to-fit (low-NLL) and hard-to-fit (high-NLL) examples. Contrary to our initial hypothesis, high-NLL examples significantly outperform low-NLL examples (33.0% vs. 23.6% aggregate accuracy), with the advantage consistent across mathematical and scientific reasoning benchmarks. Analysis reveals that low-NLL examples are confounded with textual repetition, leading to poor termination behavior. For practitioners using many-epoch repetition on small datasets, we recommend selecting hard-to-fit examples rather than easy-to-fit ones. Future work should explore optimal NLL thresholds and test generalization across models and datasets.

REFERENCES

Qiguang Chen, Libo Qin, Jinhao Liu, Dengyun Peng, Jiannan Guan, Peng Wang, Mengkang Hu, Yuhang Zhou, Te Gao, and Wangxiang Che. Towards reasoning era: A survey of long chain-of-thought for reasoning large language models. *ArXiv*, abs/2503.09567, 2025.

DeepSeek-AI, Daya Guo, Dejian Yang, et al. 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.

Hamish Ivison, Muru Zhang, Faeze Brahman, Pang Wei Koh, and Pradeep Dasigi. Large-scale data selection for instruction tuning. *ArXiv*, abs/2503.01807, 2025.

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.

Ming Li, Yong Zhang, Zhitao Li, Jiuhai Chen, Lichang Chen, Ning Cheng, Jianzong Wang, Tianyi Zhou, and Jing Xiao. From quantity to quality: Boosting llm performance with self-guided data selection for instruction tuning. pp. 7602–7635, 2023.

Ming Li, Yong Zhang, Shwai He, Zhitao Li, Hongyu Zhao, Jianzong Wang, Ning Cheng, and Tianyi Zhou. Superfiltering: Weak-to-strong data filtering for fast instruction-tuning. pp. 14255–14273, 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.

Team Olmo, Allyson Ettinger, Amanda Bertsch, et al. Olmo 3. 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 qa benchmark. *ArXiv*, abs/2311.12022, 2023.

Shaobo Wang, Xiangqi Jin, Ziming Wang, Jize Wang, Jiajun Zhang, Kaixin Li, Zichen Wen, Zhonghua Li, Conghui He, Xuming Hu, and Linfeng Zhang. Data whisperer: Efficient data selection for task-specific llm fine-tuning via few-shot in-context learning. pp. 23287–23305, 2025.

Xuezhi Wang, Jason Wei, D. Schuurmans, Quoc Le, Ed H. Chi, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *ArXiv*, abs/2203.11171, 2022.

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.

An Yang, Anfeng Li, Baosong Yang, et al. Qwen3 technical report. *ArXiv*, abs/2505.09388, 2025a.

Cehao Yang, Xueyuan Lin, Chengjin Xu, Xuhui Jiang, Xiaojun Wu, Honghao Liu, Hui Xiong, and Jian Guo. Select2reason: Efficient instruction-tuning data selection for long-cot reasoning. *ArXiv*, abs/2505.17266, 2025b.

Yixin Yang, Qingxiu Dong, Linli Yao, Fangwei Zhu, and Zhifang Sui. Rico: Refined in-context contribution for automatic instruction-tuning data selection. *ArXiv*, abs/2505.05327, 2025c.

Jia Zhang, Chen-Xi Zhang, Yao Liu, Yi-Xuan Jin, Xiao-Wen Yang, Bo Zheng, Yi Liu, and Lan-Zhe Guo. D3: Diversity, difficulty, and dependability-aware data selection for sample-efficient llm instruction tuning. *ArXiv*, abs/2503.11441, 2025.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, L. Yu, Susan Zhang, Gargi Ghosh, M. Lewis, Luke Zettlemoyer, and Omer Levy. Lima: Less is more for alignment. *ArXiv*, abs/2305.11206, 2023.