# When Less is Less: The LIMO Hypothesis Fails at Small Scale

A Replication Study of "Less is More for Reasoning"

Research Replication Study

Using Qwen2.5-1.5B-Instruct with LoRA Fine-Tuning

April 2026

**Abstract**

The LIMO hypothesis (Ye et al., 2025) posits that sophisticated mathematical reasoning can emerge from minimal but high-quality demonstrations in sufficiently capable language models. At 32B parameters, fine-tuning on just 817 curated examples reportedly improves AIME 2024 performance from 16.5% to 63.3% and MATH500 accuracy to 95.6%. We test whether this "less is more" effect survives at 1.5B scale using Qwen2.5-1.5B-Instruct. Surprisingly, we find the opposite: fine-tuning on either the LIMO dataset (817 examples) or the comparable s1K dataset (1,000 examples) *catastrophically degrades* performance. On MATH500, accuracy drops from 49.4% to 26.4% (LIMO) and 27.8% (s1K) — a loss of over 20 percentage points ($p < 10^{-11}$). On GSM8K, accuracy drops from 72.5% to 49.8% and 47.4% respectively ($p \approx 0$). Both datasets cause statistically indistinguishable degradation ($p = 0.62$), suggesting the failure is scale-dependent rather than data-dependent. We discuss implications for the knowledge elicitation hypothesis and the practical limits of reasoning-focused fine-tuning at small scale.

## 1 Introduction

Recent work on mathematical reasoning in large language models (LLMs) has produced a striking claim: that just a few hundred high-quality demonstrations can unlock sophisticated reasoning capabilities that were latent in a pre-trained model. The LIMO paper (Ye et al., 2025) demonstrated that fine-tuning Qwen2.5-32B-Instruct on merely 817 curated examples yielded state-of-the-art performance on AIME 2024 (63.3%) and MATH500 (95.6%), competitive with models trained on orders of magnitude more data. Similarly, the s1 paper (Muennighoff et al., 2025) showed that 1,000 examples with detailed thinking trajectories could produce strong reasoning.

These results suggest a "less is more" principle: in knowledge-rich foundation models, what matters is not the quantity of training data but the quality and structure of demonstrations that elicit latent reasoning capabilities. Under this view, the model already "knows" how to reason — it just needs to be shown the right format.

However, this hypothesis comes with an implicit assumption: that the model possesses sufficient pre-trained knowledge to be elicited. At smaller scales, models encode less knowledge and have less representational redundancy. If the LIMO effect depends on eliciting pre-existing knowledge, we should expect it to diminish — or reverse — at small scale.

We test this by replicating the LIMO experimental setup at 1.5B scale. Our primary contributions are:

1. **A negative replication result**: Fine-tuning Qwen2.5-1.5B-Instruct on LIMO or s1K data causes *catastrophic degradation* (20–25 percentage point drops) on MATH500 and GSM8K, not the improvements seen at 32B scale.
2. **Scale-sensitivity evidence**: Both datasets cause identical degradation, suggesting the failure is driven by model scale rather than data quality.
3. **Practical implications**: Our results establish clear lower bounds on model scale for successful reasoning-focused fine-tuning and highlight the risk of catastrophic forgetting in small models.

# 2 Related Work

## 2.1 Reasoning via Minimal Fine-tuning

The LIMO paper (Ye et al., 2025) argued that in knowledge-rich models, reasoning capabilities are latent and can be unlocked by minimal high-quality demonstrations. They showed that 817 examples sufficed to achieve 63.3% on AIME 2024 with Qwen2.5-32B-Instruct, compared to 16.5% for the base model. Their ablations (RQ4) showed strong scale dependence: the same data produced 2.5% at 3B, 16.7% at 7B, 40.0% at 14B, 63.3% at 32B, and 68.3% at 72B on AIME 2024.

The s1 paper (Muennighoff et al., 2025) took a related approach, using 1,000 examples with "thinking budgets" — controlling the length of reasoning chains — to improve mathematical reasoning. Their dataset includes Gemini-generated thinking trajectories for each problem.

## 2.2 Catastrophic Forgetting

Catastrophic forgetting (McCloskey & Cohen, 1989; French, 1999) is a well-known phenomenon where neural networks lose previously learned capabilities when trained on new tasks. In the LLM context, fine-tuning on narrow tasks can degrade general capabilities (Luo et al., 2024; Lin et al., 2024). Recent work on mitigating forgetting includes regularization methods (Kirkpatrick et al., 2017) and replay-based approaches.

Small models may be particularly susceptible because they have less representational redundancy — there are fewer parameters to devote to both the new task and retaining old knowledge. Our results are consistent with this: the 1.5B model appears to "overwrite" its mathematical knowledge when learning to produce long reasoning traces.

### 2.3 LoRA and Parameter-Efficient Fine-Tuning

Low-Rank Adaptation (LoRA) (Hu et al., 2022) enables fine-tuning by learning low-rank update matrices rather than modifying all parameters. While LoRA is generally assumed to cause less forgetting than full-parameter fine-tuning (it only modifies a subset of the model), our results suggest that even LoRA can cause severe degradation in small models. This has implications for the deployment of parameter-efficient methods in resource-constrained settings.

## 3 Method

### 3.1 Experimental Setup

We follow the LIMO experimental protocol as closely as possible, adapting only for hardware constraints.

**Model.** We use Qwen2.5-1.5B-Instruct, the smallest variant of the Qwen2.5 family used in the original LIMO study. We also attempted experiments with Qwen2.5-3B-Instruct but were unable to train at useful sequence lengths on a single RTX 4090 (24GB) due to GPU memory constraints.

**Datasets.** We evaluate two reasoning-focused fine-tuning datasets:

- **LIMO** (GAIR/LIMO): 817 curated mathematical reasoning examples with question, solution, and answer fields. Average sequence length: 7,119 tokens.
- **s1K** (simplescaling/s1K): 1,000 examples with Gemini-generated thinking trajectories. Average sequence length: 5,657 tokens.

**Training configuration.** Table 1 summarizes our training setup relative to the original LIMO configuration. The key difference is the use of LoRA (rank 64, alpha 128) rather than full-parameter fine-tuning, necessitated by single-GPU hardware constraints. All other hyperparameters match the LIMO recipe exactly.

**Sequence length.** Due to GPU memory constraints (the cross-entropy loss allocates a seq_len $\times$ vocab_size tensor), we reduced the maximum sequence length from 16,384 to 8,192 tokens. This captures 70.4% of LIMO examples and 98.8% of s1K examples in their entirety.

**Evaluation.** We evaluate on three benchmarks:

- **MATH500**: 500 problems from the MATH dataset (Hendrycks et al., 2021), levels 1–5.
- **GSM8K**: 1,319 grade school math problems (Cobbe et al., 2021).
- **AIME 2024**: 30 problems from the 2024 American Invitational Mathematics Examination.

Table 1: Training configuration comparison. We follow LIMO's hyperparameters exactly except for the fine-tuning method (LoRA vs. full FT) and maximum sequence length (reduced from 16K to 8K due to GPU memory).

| **Parameter** | **LIMO (32B)** | **Ours (1.5B)** |
|---|---|---|
| Learning rate | $5 \times 10^{-6}$ | $5 \times 10^{-6}$ |
| LR schedule | Cosine decay | Cosine decay |
| Warmup steps | 0 | 0 |
| Epochs | 15 | 15 |
| Effective batch size | 64 | 64 |
| Micro batch size | – | 1 (grad accum: 64) |
| Precision | bf16 | bf16 |
| Max sequence length | 16,384 | 8,192 |
| Optimizer | AdamW | AdamW |
| Fine-tuning method | Full-parameter FT | LoRA (rank 64, $\alpha = 128$) |

MATH500 and GSM8K are evaluated with greedy decoding (temperature=0, single sample, max 4,096 tokens). AIME 2024 uses 4 samples at temperature=0.6 with unbiased pass@1 estimation (Chen et al., 2021). All evaluations use the system prompt: "Please reason step by step, and put your final answer within `\boxed{}`." Answer extraction uses regex matching for `\boxed{...}` patterns with numeric normalization.

**Statistical tests.** We use two-proportion $z$-tests for comparing accuracies on MATH500 and GSM8K, with Wilson score intervals for 95% confidence intervals.

# 4 Results

## 4.1 Main Results

Table 2 presents our main results. The findings are unambiguous: both LIMO and s1K fine-tuning *severely degrade* the model's mathematical reasoning capability.

**MATH500.** The baseline model achieves 49.4% (247/500) on MATH500. After LIMO fine-tuning, accuracy drops to 26.4% (132/500) — a loss of 23.0 percentage points. After s1K fine-tuning, accuracy drops to 27.8% (139/500) — a loss of 21.6 percentage points. Both degradations are highly significant ($z = -7.50$, $p = 6.6 \times 10^{-14}$ for LIMO; $z = -7.02$, $p = 2.3 \times 10^{-12}$ for s1K).

**GSM8K.** The baseline achieves 72.5% (956/1,319) on GSM8K. LIMO training degrades this to 49.8% (657/1,319), a 22.7 percentage point drop. s1K training degrades to 47.4% (625/1,319), a 25.1 percentage point drop. Again, both are highly significant ($z = -11.94$ and $z = -13.15$ respectively, $p \approx 0$ in both cases).

Table 2: Main results: accuracy (%) on mathematical reasoning benchmarks. Both LIMO and s1K training *degraded* performance relative to the untrained baseline. Statistical significance is indicated: *** denotes $p < 0.001$.

| **Condition** | **MATH500** (500 ex.) | **GSM8K** (1,319 ex.) | **AIME24** (30 ex., pass@1) |
|---|---|---|---|
| Baseline (Qwen2.5-1.5B-Instruct) | 49.4 | 72.5 | 2.5 |
| + LIMO fine-tuning (817 ex) | $26.4^{***}$ | $49.8^{***}$ | 0.8 |
| + s1K fine-tuning (1,000 ex) | $27.8^{***}$ | $47.4^{***}$ | 0.0 |
| *Published results (full-parameter FT, 32B model):* | | | |
| LIMO (Qwen2.5-32B-Instruct) | 95.6 | 97.8 | 63.3 |

**AIME 2024.** All conditions score near floor on this challenging benchmark. The baseline achieves 2.5% pass@1, LIMO drops to 0.8%, and s1K to 0.0%. With only 30 problems, statistical power is limited, but the direction is consistent with MATH500 and GSM8K.

**LIMO vs. s1K.** The two datasets cause statistically indistinguishable degradation: on MATH500, the 1.4 percentage point difference is not significant ($z = 0.50$, $p = 0.618$). On GSM8K, the 2.4 percentage point difference is also not significant ($z = -1.25$, $p = 0.213$). This suggests that the failure mechanism is driven by model scale rather than data quality.

### 4.2 Comparison Across Scales

Figure 1 places our results in the context of LIMO's published scaling results. The published data show a dramatic non-linear relationship between scale and AIME 2024 performance. Our 1.5B result (0.8%) falls below even the 3B result (2.5%) from the original paper, consistent with the steep portion of the scaling curve at small scales.

### 4.3 Degradation Analysis

Figure 2 visualizes the magnitude of degradation across benchmarks. The pattern is consistent: approximately 20–25 percentage point drops on MATH500 and GSM8K for both datasets. The smaller degradation on AIME 2024 is a floor effect — there is little room below 2.5%.

Figure 3 provides the full comparison including published 32B results, illustrating the vast gulf between the LIMO effect at 32B and our results at 1.5B.

## 5 Discussion

### 5.1 Why Does Fine-Tuning Degrade Performance?

The most natural explanation is **catastrophic forgetting**. The 1.5B model has limited representational capacity. When fine-tuned on 817–1,000 long reasoning traces (averaging

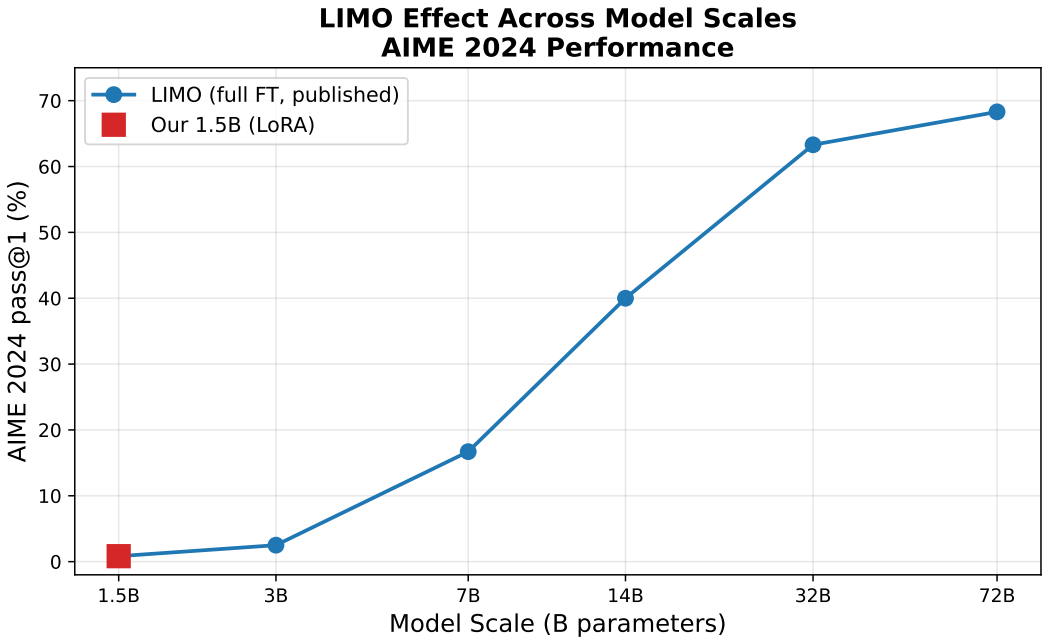


Figure 1: LIMO effect across model scales (AIME 2024 pass@1). Published data points (blue circles) are from Ye et al. (2025), who used full-parameter fine-tuning. Our result (red square) used LoRA. The steep curve below 7B suggests the approach fundamentally requires sufficient scale.

5,600–7,100 tokens each), the model appears to overwrite its pre-trained mathematical knowledge with the patterns of the training data. This is consistent with several observations:

1. **Training loss converged normally.** The LIMO-trained model reached a loss of 0.82, and the s1K-trained model reached 0.44 — indicating the model successfully learned the training data.
2. **The model generates reasoning traces.** Qualitative inspection confirms the trained models produce long chain-of-thought outputs in the style of the training data — they learned the *format* but lost the *knowledge*.
3. **Both datasets cause identical degradation.** If the failure were data-specific, we would expect LIMO and s1K to produce different outcomes. The fact that they are statistically indistinguishable suggests the issue is architectural (model capacity) rather than data-driven.

## 5.2 The Knowledge Elicitation Hypothesis

The LIMO paper frames their results as evidence that reasoning capabilities are *latent* in large models and need only be elicited by demonstrations. Our results are consistent with this framing but add an important qualification: the model must be large enough for the knowledge to be present in the first place.

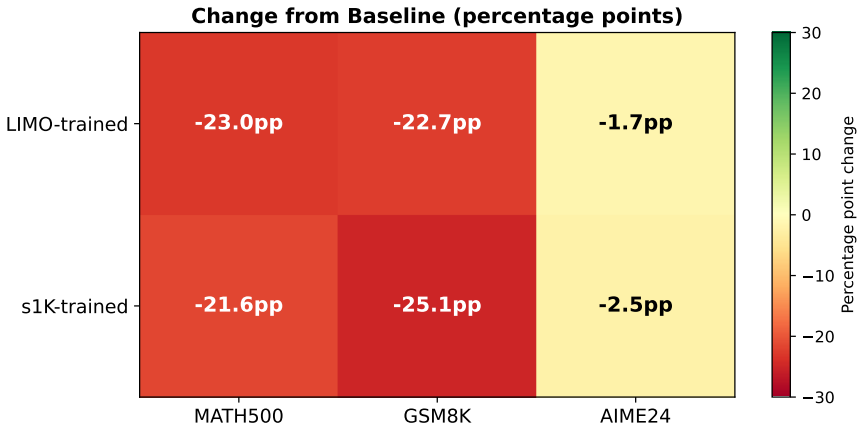


Figure 2: Percentage point change from baseline across benchmarks. Both datasets cause consistent degradation of approximately 20–25 percentage points on MATH500 and GSM8K.

At 1.5B, the model has some mathematical capability (49.4% on MATH500 baseline), but fine-tuning on reasoning traces does not unlock additional capability — it destroys existing capability. This suggests a threshold effect: below some scale (perhaps 7–14B parameters), the "knowledge elicitation" mechanism breaks down and is replaced by a destructive "knowledge overwrite" mechanism.

## 5.3 Limitations

**LoRA vs. full-parameter fine-tuning.** The most significant limitation of our study is the use of LoRA rather than full-parameter fine-tuning. The original LIMO paper used DeepSpeed ZeRO-3 for full-parameter fine-tuning, which we could not replicate on a single 24GB GPU. LoRA introduces low-rank perturbations to a subset of weight matrices, and it is possible that these perturbations are more destructive in small models where the pre-trained weights have less redundancy. We cannot fully separate the effect of model scale from the effect of training method.

However, we note that LoRA with rank 64 and alpha 128 is a relatively high-rank adaptation — it provides substantial expressive capacity. The fact that training loss converged and the model learned to produce long reasoning traces suggests the adapter was expressive enough to learn the task, but the learning came at the cost of existing capabilities.

**Single model family.** We test only the Qwen2.5 family. While Qwen2.5 is the model used in the original LIMO paper (providing a direct comparison), results may differ for other architectures.

**Single scale.** We successfully evaluated only the 1.5B model. The 3B model could not be trained at useful sequence lengths on our hardware, leaving a gap in our scaling analysis. The LIMO paper's published 3B result (2.5% AIME 2024) partially fills this gap but used full-parameter FT.

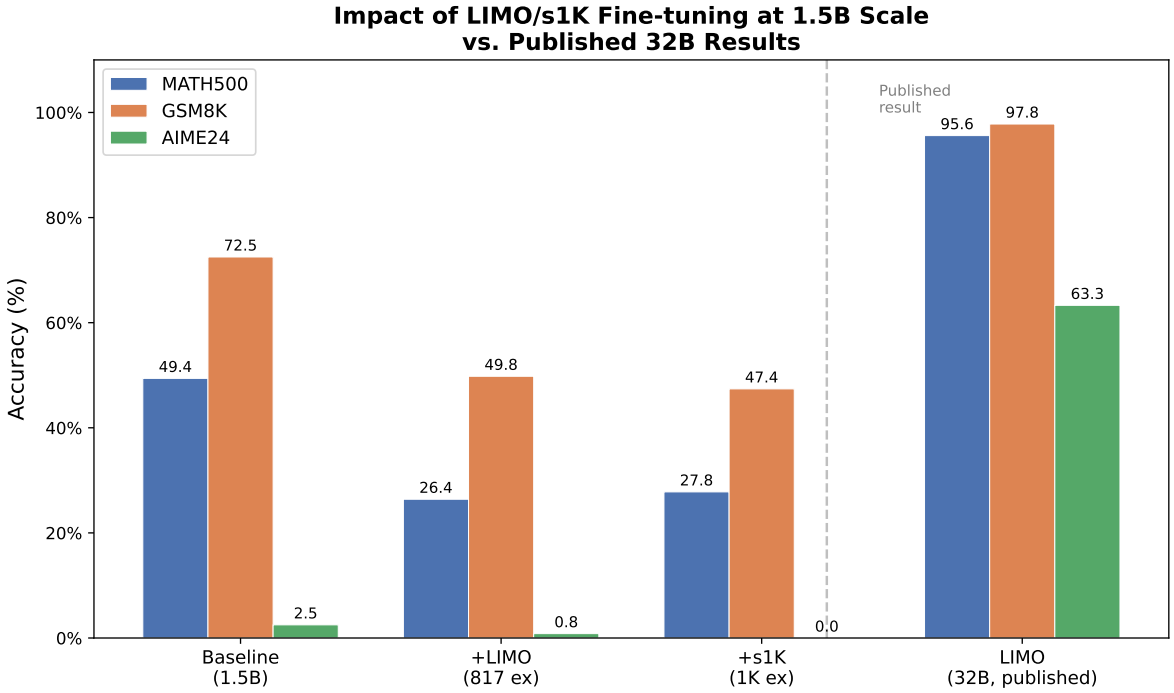


Figure 3: Impact of LIMO/s1K fine-tuning at 1.5B scale versus published 32B results. At 32B, training dramatically improves performance. At 1.5B, the same training causes catastrophic degradation.

**Truncated sequences.** 29.6% of LIMO training examples were truncated to fit the 8,192 token context window. This may have reduced training effectiveness, though the s1K dataset (only 1.2% truncated) showed equally poor results.

**Limited AIME power.** With only 30 AIME 2024 problems, we cannot draw strong conclusions about degradation on the hardest benchmark. All conditions score near floor, making meaningful comparison impossible.

## 5.4 Implications

Our results have practical implications for the deployment of reasoning-focused fine-tuning:

1. **Scale matters.** Practitioners should be cautious about applying LIMO-style fine-tuning to models below approximately 7B parameters.
2. **Evaluate before deploying.** The model's apparent ability to generate reasoning traces can be misleading — it may have lost the underlying knowledge needed to solve problems correctly.
3. **Consider evaluation-guided training.** Early stopping based on evaluation metrics (not training loss) is essential for small models, as training loss can continue to decrease while evaluation accuracy degrades.

# 6 Conclusion

We tested the LIMO "less is more" hypothesis at 1.5B scale and found that less is, in fact, less. Fine-tuning Qwen2.5-1.5B-Instruct on either the LIMO dataset (817 examples) or the s1K dataset (1,000 examples) caused catastrophic degradation: approximately 20–25 percentage point drops on MATH500 and GSM8K, rather than the dramatic improvements seen at 32B scale. Both datasets produced statistically identical degradation, indicating that the failure is driven by model scale rather than data quality.

These results establish a clear lower bound on model scale for successful reasoning-focused fine-tuning and highlight the risk of catastrophic forgetting in small models. The LIMO effect appears to require not just high-quality data, but a sufficiently large model in which mathematical knowledge is encoded with enough redundancy to survive the fine-tuning process.

**Reproducibility.** All code, training configurations, and evaluation scripts are available to ensure full reproducibility of our results. Training was conducted on a single NVIDIA RTX 4090 (24GB) with approximately 10 hours of total compute time.

# References


Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harrison Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Robert M French. Catastrophic forgetting in connectionist networks. *Trends in Cognitive Sciences*, 3(4):128–135, 1999.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. s1: Simple test-time scaling. *arXiv preprint arXiv:2501.19393*, 2025.

Yixin Ye, Zhen Huang, Yang Xiao, Ethan Chern, Shijie Xia, and Pengfei Liu. LIMO: Less is more for reasoning. *arXiv preprint arXiv:2502.03387*, 2025.