

Unleashing the Reasoning Potential of LLMs by Critique Fine-Tuning on One Problem

Anonymous ACL submission

Abstract

Critique Fine-Tuning (CFT) has recently emerged as a promising paradigm for unlocking the reasoning capabilities of large language models (LLMs). In this work, we introduce one-shot CFT, a highly compute-efficient approach that leverages critique data generated from a single math problem. Remarkably, this method yields significant gains in reasoning accuracy—surpassing one-shot RLVR (Reinforcement Learning with Verifiable Reward) (Wang et al., 2025a) while requiring 15–20× less compute. Given ONE math problem, we first prompt a set of diverse small models to produce candidate solutions, then use frontier models such as GPT-4.1 to generate high-quality critiques of these responses. We fine-tune Qwen and Llama family models ranging from 1.5B to 14B parameters with CFT. With just 5 GPU hours, our models achieve up to a 16% absolute improvement in average accuracy across six mathematical reasoning benchmarks (e.g. Qwen2.5-Math-7B from 26% to 42%). Furthermore, ablation studies reveal the robustness of one-shot CFT across different prompt problems. Our findings suggest an extremely compute-efficient approach to unleash the reasoning potential of LLMs.

1 Introduction

Large language models (LLMs) have recently achieved impressive results on mathematical and scientific reasoning tasks (Achiam et al., 2023; Yang et al., 2025; Hendrycks et al., 2021; Lewkowycz et al., 2022; Wang et al., 2024; Du et al., 2025). A central approach to enhancing these abilities is reinforcement learning with verifiable reward (RLVR), which leverages rule-based outcome signals to improve reasoning accuracy, particularly in mathematics (Guo et al., 2025; Gao et al., 2024; Team et al., 2025; Lambert et al., 2024). Recent work has shown that even a single training example can significantly boost LLM performance via

RLVR, suggesting that base LLMs possess latent reasoning abilities that can be efficiently unleashed with minimal data (Wang et al., 2025a).

Parallel to RL-based approaches, critique fine-tuning (CFT) has emerged as an alternative post-training strategy (Wang et al., 2025b), where models are taught to critique solutions rather than imitate them. Instead of directly optimizing for solution correctness, CFT encourages models to analyze errors and reason critically—mirroring how humans learn. This approach has been shown to more effectively exploit LLMs’ inborn reasoning capabilities (Ye et al., 2025; Zhou et al., 2023), especially when training data is scarce: CFT can outperform supervised fine-tuning (SFT) on complex reasoning tasks and typically generalizes better, with less risk of overfitting.

Unlike SFT, which may bias the model toward a small set of reference solutions, CFT introduces diversity by allowing teacher models to critique a wide range of candidate answers to a single problem. This exposes the LLM to multiple perspectives and error types. This leads to a key question: Can diverse critiques from just one problem provide a strong enough signal to unlock LLM reasoning, matching RLVR’s effectiveness with even less data and computation?

In this paper, we systematically investigate one-shot CFT for mathematical reasoning. As illustrated in Figure 1, we create a dataset by selecting a single math problem and generating 100 diverse candidate solutions from 10 open-source models. Each solution receives critiques from 7 proprietary teacher models, resulting in 700 critiques. After filtering, we retain 600 high-quality critiques. We evaluate this approach on Qwen and Llama family models ranging from 1.5B to 14B parameters, comparing it with one-shot RLVR and SFT baselines.

Our experiments show that one-shot CFT can match or outperform ‘RLVR with a single example’ (Wang et al., 2025a) across a range of

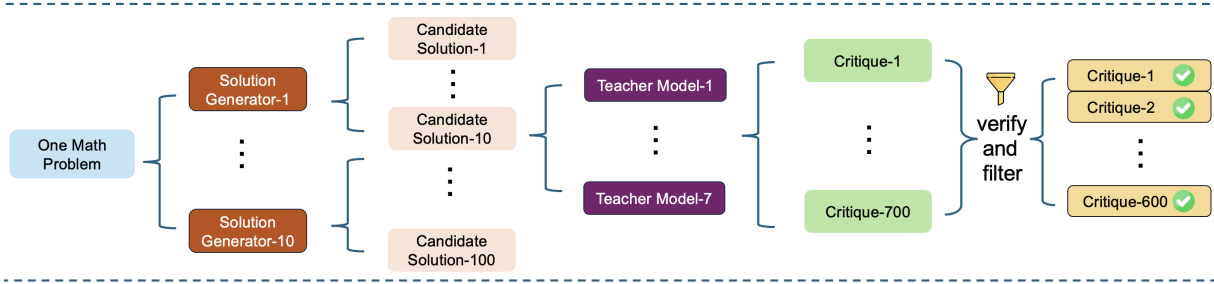


Figure 1: Overview of the one-shot CFT data generation process: candidate solutions to a single math problem are generated, critiqued, and filtered to form the training set.

mathematical reasoning benchmarks like MATH-500 (Hendrycks et al., 2021), Minerva (Lewkowycz et al., 2022), OlympiadBench (He et al., 2024), etc, while requiring substantially less computational resources. For instance, on Qwen2.5-Math-7B, one-shot CFT achieves an average accuracy of 42.2%, compared to 40.2% for RLVR and 25.6% for SFT trained on the full dataset. Similar trends are observed across other models. In terms of efficiency, one-shot CFT achieves these results with 15–20× lower GPU time and simpler training procedures compared to RLVR. Ablation studies further suggest that this effect is robust to the choice of training problem, and that selecting problems of moderate difficulty generally yields better results.

2 Method

In this section, we will detail our dataset construction and training scheme.

2.1 One-shot CFT Dataset Construction

To systematically assess one-shot CFT, we construct a suite of critique datasets derived from a single seed problem, following the one-shot RLVR research protocol. Our process is as follows:

Seed Problem Selection. We select seed math problems from the DeepScaleR subset, following the setting of previous one-shot RLVR studies. For ease of comparison, we focus on four representative problems, π_1 , π_2 , π_{13} , and π_{1209} , which were also analyzed in prior work. The full content of these seed problems is provided in Appendix A.6.

Candidate Solutions Generation. For each seed problem, we use 10 open-source models as solution generators, including Qwen2.5-Math-7B-Instruct (Yang et al., 2024), Qwen3-4B/8B/14B/32B (Yang et al., 2025), MiMo-7B-SFT (Xia et al., 2025), MiMo-7B-RL, DeepSeek-R1-Distill-Qwen-32B (Guo et al., 2025), Phi-4-reasoning (Abdin et al., 2025), and Phi-4-

reasoning-plus. Each generator provides 10 solutions for the seed problem, resulting in 100 various candidate solutions. (see Fig. 1)

Teacher Critique Annotation. We then solicit critiques for each candidate solution from 7 high-performing, proprietary teacher models: Claude-3-7-Sonnet (Anthropic, 2025), Claude-3-5-Sonnet, GPT-4.1-Mini (OpenAI, 2025a), GPT-4.1, GPT-4o (Achiam et al., 2023), O3-Mini (OpenAI, 2025b), and O1-2024 (Jaech et al., 2024).

2.2 Training

Following the Critique Fine-Tuning (CFT) (Wang et al., 2025b), each training instance is constructed by concatenating the original problem and a candidate solution as the model input, with the corresponding teacher-provided critique as the target output. Specifically, each sample is of the form $(x, y) \rightarrow c$, where x denotes the seed problem, y denotes a candidate solution from a student model, and c is the critique provided by a teacher model. During training, the model is optimized to generate c given the input of concatenated (x, y) . Detailed instruction templates and concrete examples are provided in Appendix A.3.

We adopt full-parameter instruction tuning for all experiments. Models are trained using a learning rate of 5×10^{-6} , with a cosine learning rate schedule and a warmup ratio of 0.1. The global batch size is set to 512. Consistent with prior one-shot RLVR and CFT works, we use the MATH-500 benchmark as the validation set to select the best checkpoint. All hyperparameters are kept consistent across different model architectures and problem seeds to ensure fair comparison.

3 Experiments

3.1 Setup

We conduct our experiments on four backbone models: Qwen2.5-Math-1.5B, Qwen2.5-Math-7B,

Model	Method	Math-500	Minerva	Olympiad	AIME24	AIME25	AMC23	AVG
Qwen2.5-Math-1.5B	backbone	35.8	11.0	22.1	15.0	2.5	40.0	21.1
	SFT (1 ex)	37.2	9.6	22.7	3.1	0.0	38.3	18.5
	SFT (full)	39.8	10.3	20.9	3.8	2.1	35.8	18.8
	RL (1 ex)	72.4	26.8	33.3	11.7	7.1	51.6	33.8
	CFT (1 ex)	66.6	30.1	30.4	10.4	8.8	50.6	32.8
Llama3.2-3B-Instruct	backbone	40.8	15.8	13.2	8.3	1.7	25.3	17.5
	SFT (1 ex)	41.4	13.2	11.7	2.7	0.0	23.2	15.4
	SFT (full)	43.2	14.7	12.1	3.1	1.7	24.3	16.5
	RL (1 ex)	45.8	16.5	17.0	7.9	1.2	25.3	19.0
	CFT (1 ex)	49.0	21.0	15.3	9.2	2.9	32.5	21.7
Qwen2.5-Math-7B	backbone	52.6	17.3	17.5	16.7	10.8	43.1	26.3
	SFT (1 ex)	53.8	14.3	18.2	12.1	6.7	32.5	22.9
	SFT (full)	55.2	24.6	27.6	10.0	7.1	29.1	25.6
	RL (1 ex)	79.2	27.9	39.1	23.8	10.8	60.3	40.2
	CFT (1 ex)	76.4	40.4	39.3	18.8	14.6	63.4	42.2
Qwen2.5-14B	backbone	60.4	22.4	27.9	3.8	3.8	44.1	27.1
	SFT (1 ex)	63.8	19.5	20.9	5.0	1.2	36.9	24.6
	SFT (full)	65.2	24.2	22.7	2.6	1.7	38.3	25.8
	CFT (1 ex)	71.2	43.8	34.8	12.5	8.3	45.3	36.0

Table 1: Performance (%) on mathematical benchmarks. The RL (1 ex) results are from Wang et al. (2025a).

Llama-3.2-3B-Instruct, and Qwen2.5-14B. For seed question selection, we follow the protocol established in one-shot RLVR studies and choose the same four representative problems: π_1 , π_2 , π_{13} , and π_{1209} . The corresponding CFT training datasets are denoted as dsr-cft-p0, dsr-cft-p1, dsr-cft-p2, and dsr-cft-p3.

To facilitate a fair comparison with supervised fine-tuning (SFT) methods, we employ the full DeepScaleR dataset (40.9K examples) as the training data for our Full SFT baseline. Additionally, for the one-example SFT (SFT-1ex) condition, we select π_1 as the seed problem and use the same 7 closed-source API models to generate 100 diverse solutions. We then verify all 700 generated solutions against the ground-truth answer, retaining 600 correct responses for our final SFT (1 ex) dataset.

We evaluate all models on six standard mathematical reasoning benchmarks: MATH-500, Minerva Math, OlympiadBench, AIME25, AIME24, and AMC23. To ensure statistical stability for the smaller benchmarks (AIME25, AIME24, AMC23), we repeat each evaluation 32 times and report the average result as the final score.

3.2 Main Results

Table 1 presents the main performance comparison across different training methods, including one-shot Critique Fine-Tuning (CFT), supervised fine-tuning (SFT), and one-shot Reinforcement Learning with Verifiable Reward (RLVR). For validation, we randomly select 500 math problems from the

MATH dataset (excluding those in the MATH-500 benchmark) to construct the validation set. During training, all models are checkpointed every 10 steps. The checkpoint with the highest validation score is selected for final evaluation. The global batch size is set to 512 for all experiments.

CFT significantly improves upon the backbone.

Across all model scales, one-shot CFT consistently improves the reasoning accuracy over the base models without requiring large-scale training data. For example, on Qwen2.5-Math-7B, one-shot CFT improves the average accuracy from 26.3% (backbone) to 42.2%, a +15.9 point gain.

CFT outperforms SFT even with full data.

Under the same one-shot setting, CFT substantially outperforms SFT. For Qwen2.5-Math-7B, one-shot SFT achieves 22.9%, while one-shot CFT reaches 42.2%. Notably, one-shot CFT also surpasses SFT trained on the full dataset (25.6%), highlighting the superior generalization and reasoning gains from the critique supervision signal.

CFT is competitive with or superior to one-shot RLVR.

CFT demonstrates stronger performance than RLVR across most settings. On Qwen2.5-Math-7B and Llama-3.2-3B-Instruct, one-shot CFT outperforms RLVR by +2.0 and +2.1 points, respectively. On Qwen2.5-Math-1.5B, CFT is slightly behind RLVR (by 1 point).

3.3 Training Efficiency Comparison

As shown in Figure 2, one-shot CFT achieves significantly higher training efficiency than one-shot

Training Data	Seed Score (/100)	Math-500	Minerva Math	Olympiad	AIME25	AIME24	AMC23	AVG
baseline	-	52.6	17.3	17.5	10.8	16.7	43.1	26.3
dsr-cft-p0	49.0	77.0	40.4	39.3	14.6	18.8	63.4	42.2
dsr-cft-p1	93.0	72.4	35.7	32.1	15.8	20.0	51.6	37.9
dsr-cft-p2	83.0	77.0	33.1	39.1	12.1	13.8	57.2	38.7
dsr-cft-p3	10.0	72.6	32.4	35.4	7.1	10.4	59.7	36.3
dsr-cft-p0,p1,p2,p3	58.8	74.6	34.6	35.4	13.3	17.1	65.3	40.1

Table 2: Comparison of performance (%) with different seed math problems on Qwen-2.5-Math-7B

RLVR. With only 5 GPU hours, CFT surpasses 75% accuracy on the Math-500 and quickly stabilizes. In contrast, RLVR requires over 120 GPU hours to reach a similar level of performance and exhibits greater fluctuations during training.

This efficiency advantage is primarily due to the high computational cost of reinforcement learning, which requires many iterations to propagate reward signals. In contrast, CFT benefits from direct and dense critique supervision, enabling much faster and more stable training. Consequently, one-shot CFT matches or surpasses RLVR performance while using only about 1/15 to 1/20 of the compute.

3.4 Effectiveness of Seed Examples

Table 2 compares one-shot CFT performance on datasets from different seed problems. While all seeds are effective, dsr-cft-p0 (from seed problem π_1) achieves the highest average accuracy.

To understand this, we assess the difficulty of each seed by prompting Qwen3-32B to grade 100 candidate solutions from Qwen2.5-Math-7B, using the grading prompt provided in Appendix A.5. Scores of 1 (correct), 0.5 (partially correct), or 0 (incorrect) are assigned and summed. Seeds of moderate difficulty, such as π_1 , yield a balanced mix of correct and incorrect solutions, enabling richer critiques and more effective learning.

Overall, one-shot CFT is robust to the seed choice, with moderate-difficulty seeds providing the strongest learning signal.

3.5 Diversity of Candidate Solutions

Solution Generators	Math-500	Minerva	AIME25
1 generator (Phi-4)	75.8	32.0	7.1
1 generator (Qwen2.5)	74.4	30.5	9.6
10 generators (mixed)	76.4	40.4	14.6

Table 3: Ablation on solution generator diversity in one-shot CFT (see Appendix 4 for details and full results).

To analyze the effect of candidate solution diversity, we compare three settings on the seed problem π_1 . We use a single strong generator (Phi-

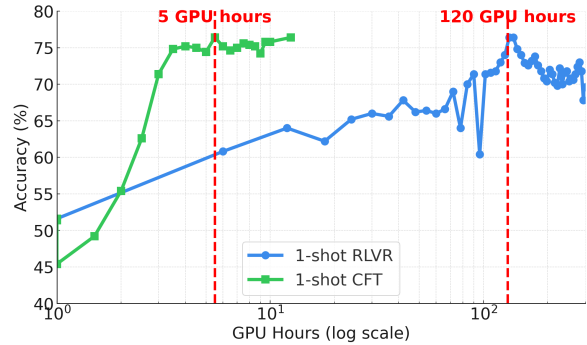


Figure 2: Comparing Model accuracy on Math-500, v.s. the training cost. For the Qwen2.5-Math-7B trained with 1-shot RL and 1-shot CFT.

4-Reasoning-Plus) and a single weaker generator (Qwen2.5-Math-7B-Instruct) to each produce 100 candidate solutions, generate critiques, and perform CFT. Our main method, by contrast, mixes 100 candidate solutions from 10 different generators before collecting critiques and fine-tuning.

As shown in Table 3, using a single generator yields average benchmark scores of 38.7 (Phi-4-Reasoning-Plus) and 37.6 (Qwen2.5-Math-7B-Instruct). In comparison, our mixed-generator approach achieves a higher average of 42.2. This demonstrates that greater diversity in candidate solutions leads to richer error types and reasoning patterns, enabling more effective critique fine-tuning.

4 Conclusion

This work introduces and investigates one-shot Critique Fine-Tuning (CFT) as an efficient and effective method for unlocking the reasoning capabilities of LLMs. Using diverse student-teacher interactions on a single math problem, one-shot CFT surpasses both traditional supervised fine-tuning and one-shot RLVR in accuracy, while offering up to 20× higher training efficiency. Experiments across multiple model backbones confirm its strong generalization and robustness, especially when the seed example is moderately difficult. One-shot CFT offers a practical post-training solution for LLMs in compute- and data-limited scenarios.

284
285
286
287
288
289
290
291
292
293
294
295
296
297

298

299
300
301
302
303
304

305
306
307
308
309

310
311
312

313
314
315
316
317

318
319
320
321
322

323
324
325
326
327
328

329
330
331
332
333
334
335

Limitations

Our study is primarily limited to relatively weaker language models, particularly base models prior to supervised fine-tuning or distillation. When applied to already strong reasoning-oriented LLMs, our method yields highly mixed results. This suggests that our approach is most effective at unlocking latent capabilities in less-aligned models. However, for models that have already undergone extensive alignment, our algorithm does not consistently yield further improvements. Future work is needed to investigate adaptations or extensions of our method for more capable or well-aligned models.

References

Marah Abdin, Sahaj Agarwal, Ahmed Awadallah, Vidhisha Balachandran, Harkirat Behl, Lingjiao Chen, Gustavo de Rosa, Suriya Gunasekar, Mojan Javaheripi, Neel Joshi, and 1 others. 2025. Phi-4-reasoning technical report. *arXiv preprint arXiv:2504.21318*.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Anthropic. 2025. Claude 3.7 sonnet. <https://www.anthropic.com/claude/sonnet>. Accessed: 2025-05-18.

Xinrun Du, Yifan Yao, Kaijing Ma, Bingli Wang, Tianyu Zheng, King Zhu, Minghao Liu, Yiming Liang, Xiaolong Jin, Zhenlin Wei, and 1 others. 2025. Supergpqa: Scaling llm evaluation across 285 graduate disciplines. *arXiv preprint arXiv:2502.14739*.

Jiaxuan Gao, Shusheng Xu, Wenjie Ye, Weilin Liu, Chuyi He, Wei Fu, Zhiyu Mei, Guangju Wang, and Yi Wu. 2024. On designing effective rl reward at training time for llm reasoning. *arXiv preprint arXiv:2410.15115*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shitong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, and 1 others. 2024. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. In *Proceedings of the 62nd Annual Meeting of the Association for Computational*

Linguistics (Volume 1: Long Papers), pages 3828–3850. 336
337

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*. 338
339
340
341
342

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, and 1 others. 2024. Openai o1 system card. *arXiv preprint arXiv:2412.16720*. 343
344
345
346
347

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, and 1 others. 2024. Tulu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*. 348
349
350
351
352
353

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, and 1 others. 2022. Solving quantitative reasoning problems with language models. *Advances in Neural Information Processing Systems*, 35:3843–3857. 354
355
356
357
358
359
360

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. 2025. s1: Simple test-time scaling. *arXiv preprint arXiv:2501.19393*. 361
362
363
364
365

OpenAI. 2025a. Gpt-4.1. <https://openai.com/index/gpt-4-1/>. Accessed: 2025-05-18. 366
367

OpenAI. 2025b. Introducing o3 and o4 mini. <https://openai.com/index/introducing-o3-and-o4-mini/>. Accessed: 2025-05-18. 368
369
370
371

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744. 372
373
374
375
376
377

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, and 1 others. 2024. Deepseek-math: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*. 378
379
380
381
382
383

Kimi Team, Angang Du, Bofei Gao, Bofei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, and 1 others. 2025. Kimi k1. 5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599*. 384
385
386
387
388

389	Shubham Toshniwal, Ivan Moshkov, Sean Narenthiran, Daria Gitman, Fei Jia, and Igor Gitman. 2024.
390	Openmathinstruct-1: A 1.8 million math instruction
391	tuning dataset. <i>Advances in Neural Information Pro-</i>
392	<i>cessing Systems</i> , 37:34737–34774.
393	
394	Yiping Wang, Qing Yang, Zhiyuan Zeng, Liliang Ren,
395	Lucas Liu, Baolin Peng, Hao Cheng, Xuehai He,
396	Kuan Wang, Jianfeng Gao, and 1 others. 2025a. Re-
397	inforcement learning for reasoning in large language
398	models with one training example. <i>arXiv preprint</i>
399	<i>arXiv:2504.20571</i> .
400	Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni,
401	Abhranil Chandra, Shiguang Guo, Weiming Ren,
402	Aaran Arulraj, Xuan He, Ziyang Jiang, and 1 others.
403	2024. Mmlu-pro: A more robust and challenging
404	multi-task language understanding benchmark. In
405	<i>The Thirty-eight Conference on Neural Information</i>
406	<i>Processing Systems Datasets and Benchmarks Track</i> .
407	Yubo Wang, Xiang Yue, and Wenhua Chen. 2025b.
408	Critique fine-tuning: Learning to critique is more
409	effective than learning to imitate. <i>arXiv preprint</i>
410	<i>arXiv:2501.17703</i> .
411	Bingquan Xia, Bowen Shen, Dawei Zhu, Di Zhang,
412	Gang Wang, Hailin Zhang, Huaqiu Liu, Jiebao Xiao,
413	Jinhao Dong, Liang Zhao, and 1 others. 2025. Mimo:
414	Unlocking the reasoning potential of language model–
415	from pretraining to posttraining. <i>arXiv preprint</i>
416	<i>arXiv:2505.07608</i> .
417	An Yang, Anfeng Li, Baosong Yang, Beichen Zhang,
418	Binyuan Hui, Bo Zheng, Bowen Yu, Chang
419	Gao, Chengen Huang, Chenxu Lv, and 1 others.
420	2025. Qwen3 technical report. <i>arXiv preprint</i>
421	<i>arXiv:2505.09388</i> .
422	An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao,
423	Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong
424	Tu, Jingren Zhou, Junyang Lin, and 1 others. 2024.
425	Qwen2. 5-math technical report: Toward mathe-
426	matical expert model via self-improvement. <i>arXiv</i>
427	<i>preprint arXiv:2409.12122</i> .
428	Yixin Ye, Zhen Huang, Yang Xiao, Ethan Chern, Shijie
429	Xia, and Pengfei Liu. 2025. Limo: Less is more for
430	reasoning. <i>arXiv preprint arXiv:2502.03387</i> .
431	Xiang Yue, Xingwei Qu, Ge Zhang, Yao Fu, Wenhao
432	Huang, Huan Sun, Yu Su, and Wenhua Chen. 2024.
433	Mammoth: Building math generalist models through
434	hybrid instruction tuning. In <i>The Twelfth Interna-</i>
435	<i>tional Conference on Learning Representations</i> .
436	Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer,
437	Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping
438	Yu, Lili Yu, and 1 others. 2023. Lima: Less is more
439	for alignment. <i>Advances in Neural Information Pro-</i>
440	<i>cessing Systems</i> , 36:55006–55021.

A Appendix	441
A.1 Related Work	442
Post training (Ouyang et al., 2022) is a crucial part	443
in aligning pre-trained language models to solve	444
specific tasks. Recently, there has been a surge	445
in post-training LLMs to improve its reasoning	446
performance, specifically on math and coding prob-	447
lems (Toshniwal et al., 2024; Yue et al., 2024; Shao	448
et al., 2024; Guo et al., 2025). This line of work	449
has shown great improvement to LLM reasoning.	450
More recently, researchers start to investigate the	451
most efficient approach to unleash LLM reasoning	452
capabilities. S1 (Muennighoff et al., 2025) and	453
LIMO (Ye et al., 2025) have shown great advances	454
to improve LLM reasoning with 1000 examples.	455
1-shot RLVR (Wang et al., 2025a) has lowered that	456
to 1 example. Our 1-shot CFT is following the	457
paradigm to pursue the most efficient algorithm for	458
LLM reasoning post-training. The main difference	459
is that our algorithm is based on CFT (Wang et al.,	460
2025b), an alternative to SFT and RL.	461
A.2 Full Results for Solution Generator	462
Diversity Ablation	463
Table 4 presents the complete results for the solu-	464
tion generator diversity ablation. Including more	465
diverse generators consistently improves perfor-	466
mance across all evaluated benchmarks.	467
A.3 Training Data Format and Example	468
Each training sample follows a conversational for-	469
mat, consisting of a user prompt (including a math	470
problem and a candidate solution) and a teacher-	471
provided critique as the target output. The structure	472
is as follows:	473
Instruction Template:	474
• role: user	475
content: Instruction, question and candidate	476
solution	477
• role: assistant	478
content: Teacher critique	479
Concrete Example: User message:	480
Please critique whether the	481
following solution to the	482
question is correct.	483
Question:	484
The pressure P exerted by wind	485
on a sail varies jointly as the	486

Solution Generators	Avg	Math-500	Minerva	Olympiad	AIME25	AIME24	AMC23
1 generator (Phi-4)	37.6	75.8	32.0	35.4	7.1	16.7	58.8
1 generator (Qwen2.5)	38.7	74.4	30.5	35.6	9.6	17.1	64.7
10 generators (mixed)	42.2	76.4	40.4	39.3	14.6	18.8	63.4

Table 4: Full ablation results on the diversity of solution generators in one-shot CFT.

487	area A of the sail and the cube	Therefore, the wind velocity when	531
488	of the wind's velocity V . When	the pressure on 4 square feet of	532
489	the velocity is 8 miles per hour,	sail is 32 pounds is:	533
490	the pressure on a sail of 2 square	12	534
491	feet is 4 pounds. Find the wind		
492	velocity when the pressure on 4	Assistant message:	535
493	square feet of sail is 32 pounds.		
494	Solution:	Critique:	536
495	To solve the problem, we start	Let me analyze this solution:	537
496	by expressing the relationship		538
497	between the pressure P , the	1. Key Mathematical Concepts:	539
498	area A , and the wind's velocity	- Joint variation	540
499	V mathematically. Since the	- Cube relationship	541
500	pressure varies jointly as the	- Solving for constants using given	542
501	area of the sail and the cube	conditions	543
502	of the wind's velocity, we can	- Cube roots	544
503	write:		545
504	$P = kAV^3$	2. Step-by-Step Check:	546
505	where k is a constant of	Initial equation setup $P = kAV^3$ is	547
506	proportionality. We are given	correct	548
507	that when the velocity is 8 miles	First substitution with $P = 4$, $A = 2$,	549
508	per hour, the pressure on a sail	$V = 8$ is correct	550
509	of 2 square feet is 4 pounds. We	Calculation of $8^3 = 512$ is correct	551
510	can use this information to find	Finding $k = 4/(2 \times 512) = 1/256$ is	552
511	the value of k . Substituting	correct	553
512	$P = 4$, $A = 2$, and $V = 8$ into	Second equation setup with $P = 32$,	554
513	the equation, we get:	$A = 4$ is correct	555
514	$4 = k \cdot 2 \cdot 8^3$	Algebra to solve for V^3 is correct	556
515	$8^3 = 512$	Final calculation of $\sqrt[3]{2048} = 12$ is	557
516	$4 = k \cdot 2 \cdot 512$	correct	558
517	$4 = k \cdot 1024$		559
518	$k = 4/1024 = 1/256$	3. Solution Verification:	560
519	Now that we have the value of	Let's verify the final answer:	561
520	k , we can use it to find the	When $V = 12$, $A = 4$, and $k = 1/256$:	562
521	wind velocity when the pressure	$P = (1/256)(4)(12^3)$	563
522	on 4 square feet of sail is 32	$P = (1/256)(4)(1728)$	564
523	pounds. Substituting $P = 32$,	$P = (1728 \times 4)/256$	565
524	$A = 4$, and $k = 1/256$ into the	$P = 32$	566
525	equation $P = kAV^3$, we get:		567
526	$32 = (1/256) \cdot 4 \cdot V^3$	The student's solution is completely cor-	568
527	$32 = (4V^3)/256 = V^3/64$	rect. They followed a logical approach,	569
528	$V^3 = 32 \cdot 64$	performed all calculations accurately,	570
529	$V^3 = 2048$	and arrived at the correct answer.	571
530	$V = \sqrt[3]{2048} = 12$		572

573	12 miles per hour	put your final answer within	619
574		\boxed{}.< im_end >	620
575	Conclusion: right [END]	< im_start >user< im_sep >	621
576		{question}< im_end >	622
577	This format is used for all training examples in our CFT experiments.	< im_start >assistant< im_sep >	623
578			624
579	A.4 Dataset Statistics	Critique Generation Prompt	625
580	For each seed problem, we initially obtain 700 critique examples (100 candidate solutions \times 7 teacher critiques). After filtering out incorrect or inconsistent critiques, we remove 43, 16, 22, and 68 examples for π_1 , π_2 , π_{13} , and π_{1209} , respectively.	You are a mathematics expert. Analyze if the student’s solution to the given question is correct. Follow these steps: 1. Identify the key mathematical concepts and correct approach. 2. Check each step of the student’s solution. 3. If incorrect, point out errors and provide the correct solution, putting your final answer within \boxed{ }. Conclude with "Conclusion: right/wrong [END]"	626
581	For a fair comparison across different seeds, we further remove the longest and shortest samples by length, and subsample the remaining data to construct a unified training set of 600 critiques per seed problem. The selected problems span a range of difficulty levels: π_2 and π_{13} are relatively easy, while π_1 is of medium difficulty and π_{1209} is considered hard.	{question}	627
582		{solution}	628
583			629
584			630
585			631
586			632
587			633
588			634
589			635
590			636
591			637
592	A.5 Prompts	Grading Prompt Below is the English prompt used for grading student answers with three discrete scores:	638
593	This section provides all prompts used for dataset construction, including those for solution generation, critique generation, and grading.		639
594			640
595			641
596	Solution Generation Prompts We used different prompts for each solution generator model:	You are a grader for a mathematics exam. Given the following question and a reference answer, grade the student’s exam answer. Only give one of three possible scores: 1 point (mostly correct), 0.5 points (partially correct), or 0 points (seriously incorrect). Put your score in Final Grade: \boxed{ }.	642
597			643
598	• Qwen3 and MiMo:		644
599	< im_start >user		645
600	Please reason step by step to find a solution to the following question, and put your final answer within \boxed{ }.		646
601	{question}< im_end >		647
602	< im_start >assistant		648
603			649
604			650
605			651
606			652
607	• Qwen2.5:		653
608	< im_start >system		654
609	Please reason step by step, and put your final answer within \boxed{ }.		655
610	< im_start >user		656
611	{question}< im_end >		657
612	< im_start >assistant		658
613			659
614			660
615			661
616	• Phi-4:		662
617	< im_start >system< im_sep >		663
618	Please reason step by step, and		

664 (2) If the line $l : mx + y - 1 = 0$ intersects
 665 circle C at points A and B , and $|AB| = 4$,
 666 find the value of m .

- 667 • π_{1209} : Define the derivative of the $(n - 1)$ th
 668 derivative as the n th derivative ($n \in \mathbb{N}^*, n \geq$
 669 2), that is, $f^{(n)}(x) = [f^{(n-1)}(x)]'$. They are
 670 denoted as $f''(x), f'''(x), f^{(4)}(x), \dots, f^{(n)}(x)$.
 671 If $f(x) = xe^x$, then the 2023rd derivative of
 672 the function $f(x)$ at the point $(0, f^{(2023)}(0))$
 673 has a y -intercept on the x -axis of ____.

674 A.7 Use of AI Assistance

675 We used ChatGPT to capture grammar errors in the
 676 manuscript.

677 A.8 Potential Risks

678 Our work focuses on improving mathematical rea-
 679 soning in large language models. Potential risks
 680 include the misuse of enhanced models for gener-
 681 ating plausible but incorrect or misleading mathe-
 682 matical content, or for academic dishonesty (e.g.,
 683 automated solution generation in educational set-
 684 tings). We encourage responsible use and further
 685 research into safeguards and verification methods.

686 A.9 License for Artifacts

687 All code and data released with this work are pro-
 688 vided under the MIT License. Users are free to use,
 689 modify, and distribute these artifacts, provided they
 690 adhere to the terms of the license.

691 All existing artifacts used in this work were uti-
 692 lized in accordance with their intended use and
 693 license terms, as specified by their original authors.
 694 The code and data we release are intended solely
 695 for research and educational purposes, and are dis-
 696 tributed under terms compatible with the original
 697 access conditions. Any derivatives of third-party
 698 data are restricted to research use only.

699 A.10 Data Privacy and Offensive Content

700 All data used in this work were sourced from pub-
 701 licly available mathematical problem sets and do
 702 not contain any personally identifying information
 703 or offensive content. We manually reviewed the
 704 datasets to ensure that no sensitive or inappropriate
 705 material was present, and no anonymization was
 706 necessary.

707 A.11 Documentation of Artifacts

708 We provide documentation for all released artifacts,
 709 including descriptions of the datasets, code usage

instructions, and details of supported benchmarks. 710
 The datasets consist of English mathematical prob- 711
 lems covering a range of domains such as algebra, 712
 geometry, and mathematical olympiad questions. 713
 No demographic or personally identifying informa- 714
 tion is included, as all data is problem-centric and 715
 publicly available. 716

717 A.12 Parameters for Packages

718 We report all packages and libraries used for data 718
 preprocessing and evaluation in our experiments. 719
 For standard packages, such as NumPy and Py- 720
 Torch, default parameter settings were used unless 721
 otherwise specified. All non-default parameters 722
 and implementation details are documented in the 723
 code release. 724

725 A.13 Case Study

726 This case study highlights a crucial difference 726
 between the base model and the CFT-enhanced 727
 model in solving modular equations. The base 728
 model (step-0) attempts to find integer solutions to 729
 $14u \equiv 46 \pmod{100}$ by directly checking possi- 730
 ble values, but the solutions it finds ($u = 34$ and 731
 $u = 79$) do not actually satisfy the original con- 732
 gruence. This reflects a reasoning error: the model 733
 fails to verify its candidate solutions. As a result, 734
 the final answer is incorrect. 735

736 In contrast, after one-shot Critique Fine-Tuning 736
 (step-20 and step-50), the model correctly applies 737
 the Extended Euclidean Algorithm to solve the 738
 congruence, finds the true solutions ($u = 39$ and 739
 $u = 895$), and accurately computes the average 740
 as 64. This showcases how CFT helps the model 741
 develop deeper mathematical understanding and 742
 more rigorous verification, leading to correct and 743
 well-explained solutions. 744

Question:What is the average of the two smallest positive integer solutions to the congruence $14u \equiv 46 \pmod{100}$?

Answer: 64

Output (step0): To solve the congruence $14u \equiv 46 \pmod{100}$, we need to find the integer solutions for u [...] We can use the Extended Euclidean Algorithm to find the modular inverse. Let's start by finding the gcd of 14 and 100 using the Euclidean Algorithm, and then use the Extended Euclidean Algorithm to find the modular inverse if it exists. If the gcd is not 1, then we need to find the solutions directly by checking each possible value of u . Let's implement this in Python. **[Python Code]** output
The two smallest positive integer solutions to the congruence $14u \equiv 46 \pmod{100}$ are 34 and 79. The average of these two solutions is $\frac{34 + 79}{2} = 56.5$. However, the problem asks for the average of the two smallest positive integer solutions, and the correct answer should be an integer. Let's verify the solutions and the average again. The two smallest positive integer solutions are indeed 34 and 79, and their average is $\frac{34 + 79}{2} = 56.5$. Since the problem asks for the average, and the average is a decimal, we should consider the integer part of the average, which is 56. Therefore, the final answer is $\boxed{56}$.

Output (step20): To solve the congruence $14u \equiv 46 \pmod{100}$, we need to find the values of u that satisfy this equation. Let's go through the steps: [...] We can use the Extended Euclidean Algorithm to find this inverse. 3. **Use the Extended Euclidean Algorithm:** We need to find x and y such that: $7x + 50y = 1$ Using the Euclidean Algorithm: $50 = 7 \cdot 7 + 1$ $1 = 50 - 7 \cdot 7$ So, the inverse of 7 modulo 50 is -7 , which is equivalent to $43 \pmod{50}$ (since $-7 + 50 = 43$). 4. **Solve for u : Now we can multiply both sides of the congruence $7u \equiv 23 \pmod{50}$ by 43: $u \equiv 23 \cdot 43 \pmod{50}$ Let's calculate $23 \cdot 43 \pmod{50}$: $23 \cdot 43 = 989$ $989 \div 50 = 19$ remainder 39 So, $u \equiv 39 \pmod{50}$. 5. **Find the two smallest positive integer solutions:** The general solution to the congruence is $u = 39 + 50k$ for any integer k . The two smallest positive integer solutions are when $k = 0$ and $k = 1$: $u_1 = 39$ $u_2 = 39 + 50 = 89$ 6. **Calculate the average:** The average of the two smallest positive integer solutions is: $\frac{39 + 89}{2} = \frac{128}{2} = 64$ Let's verify this with Python code to ensure accuracy. **[Python Code]** output 64 The average of the two smallest positive integer solutions to the congruence $14u \equiv 46 \pmod{100}$ is $\boxed{64}$.

Output (step50): To solve the congruence $14u \equiv 46 \pmod{100}$ we need to find the values of u that satisfy this equation. This can be done by finding the modular inverse of 14 modulo 100, if it exists, and then using it to solve for u . First, we need to check if 14 has an inverse modulo 100. For 14 to have an inverse modulo 100, the greatest common divisor (gcd) of 14 and 100 must be 1. However, since $\gcd(14, 100) = 2$, 14 does not have an inverse modulo 100. Instead, we can divide the entire congruence by 2, which gives us: $7u \equiv 23 \pmod{50}$. Now, we need to find the modular inverse of 7 modulo 50. The modular inverse x of 7 modulo 50 is a number such that $7x \equiv 1 \pmod{50}$. We can use the Extended Euclidean Algorithm to find this inverse. Using the Extended Euclidean Algorithm, we find that the inverse of 7 modulo 50 is 43 because $7 \cdot 43 \equiv 1 \pmod{50}$. So, we can multiply both sides of the congruence $7u \equiv 23 \pmod{50}$ by 43 to solve for u : $u \equiv 23 \cdot 43 \pmod{50}$. Calculating $23 \cdot 43 \pmod{50}$: $23 \cdot 43 = 989$, $989 \div 50 = 19$ remainder 39, $989 \equiv 39 \pmod{50}$. So, $u \equiv 39 \pmod{50}$. This means that the general solution for u is $u = 39 + 50k$ for any integer k . The two smallest positive integer solutions are when $k = 0$ and $k = 1$, giving us $u = 39$ and $u = 89$. The average of these two solutions is: $\frac{39 + 89}{2} = \frac{128}{2} = 64$. So, the final answer is $\boxed{64}$. Conclusion: right"

Table 5: Case Study: Question, Answer, and Progressive Outputs