Harsher Self-Critique Hurts: An Empirical Study of Critique Stringency, Budget Control, and Effort-Inducing Strategies for LLMs

Anonymous Authors

Abstract

Large language models (LLMs) often produce generic or shallow responses despite being capable of more careful work—a phenomenon known as the "lazy LLM" problem. A popular hypothesis holds that prompting LLMs to be harsher critics of their own work should improve output quality. We systematically test this hypothesis through four controlled experiments on the GSM8K math reasoning benchmark using GPT-4.1-mini. Our findings are striking and counterintuitive: (1) increasing self-critique stringency *monotonically degrades* accuracy from 96.0% to 91.0% ($p < 0.01$), contradicting the popular intuition; (2) token budget constraints maintain or slightly improve accuracy (96.7%) while reducing token usage by 32%; (3) budget constraints partially mitigate critique-induced damage in a factorial design; and (4) no alternative effort-inducing strategy—including high-stakes framing, explicit rubrics, step verification, or self-consistency voting—outperforms simple chain-of-thought prompting, despite using 1.2–5× more tokens. These results demonstrate that the "lazy LLM" problem is not solvable through harder self-critique. Self-critique introduces errors by causing models to second-guess correct answers at a higher rate than it repairs incorrect ones. The most effective intervention is simply constraining response length, which forces concise reasoning without the error amplification of self-evaluation.

1 Introduction

Users of large language models frequently complain that models "take the easy way out"—producing verbose but superficial responses, using brute-force approaches instead of elegant solutions, and failing to demonstrate the careful reasoning they are capable of. This perception is empirically grounded: SELF-REFINE [Madaan et al., 2023] showed that LLM outputs consistently fall short of model capability, while Poddar et al. [2025] found responses are 3–20× longer than necessary for factual questions. A common folk hypothesis holds that being more aggressive in prompting—asking the model to try harder, framing the task as high-stakes, or instructing it to ruthlessly critique its own work—should produce better results.

We test this hypothesis directly. Across four experiments on the GSM8K math reasoning benchmark [Cobbe et al., 2021] using GPT-4.1-mini, we find that **harsher self-critique monotonically degrades accuracy** from 96.0% (no critique) to 91.0% (harsh critique, $p < 0.01$). The model is more likely to "find" errors in correct solutions than to detect and fix real errors—a phenomenon we term *asymmetric error introduction*. In contrast, simply constraining response length to 75 words achieves 96.7% accuracy while using 32% fewer tokens. No other effort-inducing strategy (high-stakes framing, explicit rubrics, step verification, or self-consistency voting) outperforms simple chain-of-thought prompting.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

Gap in existing work. The literature on LLM self-correction is divided. SELF-REFINE [Madaan et al., 2023] demonstrated improvements through iterative self-feedback on subjective tasks but showed near-zero improvement on math reasoning. Huang et al. [2024] found that intrinsic self-correction degrades reasoning performance, and Kamoi et al. [2024] concluded that self-correction on reasoning tasks has never been demonstrated to work under fair conditions. However, no prior work has systematically varied the *stringency* of self-critique from gentle to adversarial, tested the interaction between critique intensity and response budget, or compared critique-based approaches against multiple alternative effort-inducing strategies within a unified experimental framework.

Our contributions. We make the following contributions:

- We conduct the first systematic study of critique stringency, testing five levels from no critique to adversarial self-evaluation, and show that accuracy degrades monotonically with harshness.
- We demonstrate that token budget constraints are a simple, effective intervention that maintains or improves accuracy while reducing computational cost by up to 32%.
- We show through a 3×3 factorial design that budget constraints partially mitigate critique-induced damage, and we identify the mechanism: constraining critique length limits the model's ability to fabricate errors.
- We compare five effort-inducing strategies and find that none outperforms simple chain-of-thought prompting on a near-ceiling benchmark, establishing a negative but practically important result.

The remainder of this paper is organized as follows. Section 2 reviews related work on self-correction and budget control. Section 3 describes our experimental design. Section 4 presents results across all four experiments. Section 5 discusses implications, mechanisms, and limitations. Section 6 concludes.

2 Related Work

We organize related work into three themes: self-correction and self-critique in LLMs, response length and budget control, and alternative strategies for improving LLM output quality.

Self-correction in LLMs. SELF-REFINE [Madaan et al., 2023] introduced iterative self-refinement, where a single LLM generates, critiques, and revises its own output. Across seven tasks, SELF-REFINE improved outputs by approximately 20% on average, with the largest gains on subjective tasks such as dialogue response generation (+49.2%) and sentiment reversal (+32.4%). A critical ablation showed that specific feedback dramatically outperformed generic feedback. However, SELF-REFINE showed near-zero improvement on math reasoning, with the model producing "everything looks good" feedback 94% of the time.

Huang et al. [2024] directly challenged the premise of self-correction, showing that on GSM8K with GPT-3.5-Turbo, self-correction changed 12.3% of correct answers to incorrect but only 1.7% of incorrect answers to correct. Kamoi et al. [2024] provided the most comprehensive analysis, concluding that intrinsic self-correction on reasoning tasks has never been demonstrated to work under fair experimental conditions. They identified a key insight: LLMs cannot *find* reasoning errors but *can* correct them given the error location—the bottleneck is error detection, not correction. Xu et al. [2024] revealed that LLMs systematically overestimate the quality of their own output (self-bias), and this bias amplifies during iterative self-refinement.

Our work extends this literature by systematically varying critique *stringency* rather than testing a binary on/off critique condition. We show that the degradation is not merely present but monotonically increases with harshness.

Response length and budget control. Multiple lines of work demonstrate that LLM verbosity is a systematic problem and that constraining output length can improve both efficiency and accuracy. TALE [Han et al., 2024] introduced token-budget-aware reasoning, showing that explicit budgets can improve accuracy on GSM8K while reducing token usage. CCoT [Nayab et al., 2024] demonstrated 12–25% reduction in redundancy with maintained or improved accuracy through simple length-constraining prompts. Wu et al. [2025] provided theoretical evidence that accuracy follows an inverted U-curve with chain-of-thought length, and that more capable models achieve peak performance with shorter reasoning chains. Poddar et al. [2025] benchmarked 12 LLMs and found responses are 3–20$\times$ longer than necessary, with simple brevity prompts achieving 25–60% energy savings while improving quality.

Level	Critique Instruction
L0 (Baseline)	No critique; direct chain-of-thought only
L1 (Gentle)	"Please briefly review your answer. . . Check if there are any errors"
L2 (Moderate)	"Carefully examine each step. . . Identify any logical errors, calculation mistakes, or unjustified leaps"
L3 (Harsh)	"You are an extremely strict math professor. . . Find every flaw, no matter how small. . . Assume there ARE errors"
L4 (Adversarial)	"Your job is to DESTROY this solution. . . This answer is probably wrong—prove it"

Table 1: Five levels of critique stringency used in Experiment 1. L0 serves as the no-critique baseline. L1–L4 implement increasingly aggressive self-evaluation instructions applied after an initial solution is generated.

Our budget control experiments build on this work by testing the *interaction* between budget constraints and self-critique—a combination not previously studied.

Alternative improvement strategies. Self-consistency [Wang et al., 2023] uses majority voting over multiple samples, providing a compute-matched baseline for any refinement strategy. Reflexion [Shinn et al., 2023] uses verbal self-reflection stored in episodic memory, though Kamoi et al. [2024] noted that some experiments used exact-match ground truth as feedback. Tree of Thoughts [Yao et al., 2023] explores multiple reasoning paths through explicit search. These approaches all add computational cost; we compare several of them against simple chain-of-thought under matched or unmatched compute budgets.

3 Methodology

We design four experiments to test the effects of critique stringency, budget control, their interaction, and alternative effort-inducing strategies on LLM reasoning accuracy.

3.1 Benchmark and Model

Benchmark. We use GSM8K [Cobbe et al., 2021], a dataset of 1,319 grade-school math word problems with numerical answers. We select random subsets of 150–200 problems (seed=42) for each experiment. GSM8K provides unambiguous ground truth, enabling precise accuracy measurement, and is the most widely used benchmark in the self-correction literature [Madaan et al., 2023, Huang et al., 2024, Kamoi et al., 2024, Han et al., 2024, Nayab et al., 2024].

Model. We use GPT-4.1-mini via the OpenAI API with temperature $T = 0$ for deterministic outputs (except where noted for self-consistency). We extract answers using regex patterns for the `#### [number]` format, with fallback patterns for "the answer is" and LaTeX `\boxed{}` notation. Correctness is determined by numerical comparison with tolerance 10^{-6}.

3.2 Experiment 1: Critique Stringency Spectrum

We test five levels of critique stringency using a generate-then-critique protocol on $n = 200$ problems. Table 1 describes each level. For L0 (baseline), the model produces a single chain-of-thought response. For L1–L4, the model first generates a solution, then receives the critique prompt along with its own solution and produces a revised answer.

3.3 Experiment 2: Budget Control

We test four budget levels with no self-critique on $n = 150$ problems: no word limit, 300 words, 150 words, and 75 words. Budget instructions are appended to the standard chain-of-thought prompt (e.g., "Keep your solution to at most 75 words. Be very concise—only essential steps.").

Level	**Acc. (%)**	**95% CI**	**Changed**	**C→W**	**Tokens**
L0 (Baseline)	**96.0**	[92.3, 98.0]	0.0%	0.0%	305
L1 (Gentle)	94.5	[90.4, 96.9]	2.5%	2.1%	776
L2 (Moderate)	93.5	[89.2, 96.2]	4.5%	3.6%	885
L3 (Harsh)	91.0	[86.2, 94.2]	6.5%	5.7%	1,218
L4 (Adversarial)	91.5	[86.8, 94.6]	6.5%	5.7%	1,232

Table 2: Experiment 1: Critique stringency results ($n = 200$). **Acc.** = accuracy. **Changed** = percentage of answers modified after critique. **C→W** = percentage of originally correct answers changed to incorrect. **Tokens** = average tokens per problem. The L0 vs. L3 comparison is significant at $p = 0.009$ (McNemar's test).

3.4 Experiment 3: Critique $\times$ Budget Factorial

We use a 3×3 factorial design crossing three critique levels (none, moderate, harsh) with three budget levels (no limit, 150 words, 75 words), yielding nine conditions on $n = 150$ problems. This design tests whether budget constraints interact with critique stringency.

3.5 Experiment 4: Alternative Effort-Inducing Strategies

We compare five strategies on $n = 150$ problems:

- **Baseline Chain-of-Thought**: standard chain-of-thought prompting.
- **High Stakes**: emotional urgency framing ("This is an extremely important exam that will determine your entire career...").
- **Explicit Rubric**: generate a solution, evaluate against a 4-criterion rubric, then revise.
- **Step Verification**: generate a solution, then re-solve the problem independently and reconcile.
- **Self-Consistency** ($k = 5$): five samples at temperature $T = 0.7$, majority vote.

3.6 Statistical Analysis

We use McNemar's test with continuity correction for paired binary outcomes (correct/incorrect), as all conditions share the same problem set. We report Cohen's h for effect size, 95% Wilson score confidence intervals for proportions, and set significance at $\alpha = 0.05$.

3.7 Reproducibility

All API calls are cached by SHA-256 hash of request parameters, ensuring exact reproducibility. Code, data, and results are publicly available.[1]

4 Results

4.1 Experiment 1: Harsher Critique Monotonically Degrades Accuracy

Table 2 presents accuracy across five critique stringency levels. The baseline (L0, no critique) achieves 96.0% accuracy. Each increase in critique stringency reduces accuracy, with the decline reaching statistical significance at L3 (harsh, $p = 0.009$, Cohen's $h = -0.207$) and L4 (adversarial, $p = 0.027$, Cohen's $h = -0.189$). The relationship is monotonically negative—there is no inverted U-curve, **rejecting H1**.

The mechanism is clear from the transition rates. At L3 and L4, the model changes 6.5% of its answers after self-critique. Of originally correct answers, 5.7% are changed to incorrect (Correct→Wrong), while only 12.5–25% of originally incorrect answers are salvaged (Wrong→Correct). Since the baseline is at 96% accuracy, there are far more correct answers to damage than incorrect answers to fix. Self-critique costs 2.5–4$\times$ more tokens than the baseline while degrading accuracy.

[1]Repository details withheld for anonymous review.

Budget	**Acc. (%)**	**95% CI**	**Avg. Words**	**Avg. Tokens**
No limit	95.3	[90.7, 97.7]	123	290
300 words	96.0	[91.5, 98.2]	134	318
150 words	**96.7**	[92.4, 98.6]	87	253
75 words	**96.7**	[92.4, 98.6]	45	198

Table 3: Experiment 2: Budget control results ($n = 150$). Tighter budgets maintain or slightly improve accuracy while reducing token usage. The 75-word condition uses 32% fewer tokens than the unconstrained condition.

	No Limit	**150 words**	**75 words**
No Critique	95.3%	**96.7%**	**96.7%**
Moderate Critique	92.7%	94.7%	94.7%
Harsh Critique	88.0%	91.3%	92.0%

Table 4: Experiment 3: Critique $\times$ budget factorial ($n = 150$). Accuracy (%) across nine conditions. Budget constraints partially rescue accuracy under harsh critique (88.0% $\rightarrow$ 92.0%), but even with tight budgets, critique still degrades performance relative to no critique at the same budget level.

4.2 Experiment 2: Budget Constraints Maintain Accuracy at Lower Cost

Table 3 presents results for four budget levels. Budget constraints do not hurt accuracy at any level tested. The tightest constraint (75 words) achieves the highest accuracy (96.7%) while using 32% fewer tokens than the unconstrained condition. While no individual comparison reaches statistical significance (the sample size of 150 limits power for small effects), the trend is consistent: **H2 is partially supported**.

4.3 Experiment 3: Budget Constraints Partially Mitigate Critique Damage

Table 4 presents the 3 $\times$ 3 factorial results. Two main effects emerge:

Critique effect (negative). Within each budget level, adding critique reduces accuracy. The worst condition (harsh critique, no limit: 88.0%) is 8.7 percentage points below the best (no critique, tight budget: 96.7%).

Budget effect (positive). Within each critique level, tighter budgets improve accuracy. This is most pronounced for harsh critique, where the 75-word budget (92.0%) partially recovers the loss from unconstrained harsh critique (88.0%). Budget constraints appear to limit the model's ability to "overthink" during self-evaluation: when forced to be concise in its critique, the model is less likely to fabricate errors in correct solutions. **H3 is partially supported**, though in the unexpected direction that budget helps primarily by mitigating critique damage rather than by synergistically combining with critique.

Table 5 shows that budget constraints also reduce the token overhead of critique.

4.4 Experiment 4: No Alternative Strategy Outperforms Baseline

Table 6 presents results for five effort-inducing strategies. No strategy achieves even marginally significant improvement over the baseline. **H4 is rejected.**

High-stakes emotional framing has zero measurable effect on mathematical reasoning—the model produces identical results with slightly more verbose phrasing. Self-consistency voting over five samples uses 5$\times$ the compute for no accuracy gain; at 95.3% per-sample accuracy, the expected disagreement rate is too low to benefit from consensus. The explicit rubric strategy shows a slight (non-significant) accuracy *decrease* to 94.7%, consistent with our finding that critique-like interventions tend to harm performance.

	No Limit	**150 words**	**75 words**
No Critique	290	252	202
Moderate Critique	850	770	731
Harsh Critique	1,092	813	759

Table 5: Average tokens per problem across factorial conditions. Budget constraints substantially reduce the token overhead of self-critique.

Strategy	**Acc. (%)**	**95% CI**	**Tokens**	**Cost**
Baseline Chain-of-Thought	**95.3**	[90.7, 97.7]	290	1.0×
High Stakes	95.3	[90.7, 97.7]	351	1.2×
Explicit Rubric	94.7	[89.8, 97.3]	950	3.3×
Step Verification	95.3	[90.7, 97.7]	815	2.8×
Self-Consistency (k=5)	95.3	[90.7, 97.7]	1,454	5.0×

Table 6: Experiment 4: Alternative strategies ($n = 150$). No strategy improves over baseline chain-of-thought prompting. **Cost** = token multiplier relative to baseline. All McNemar p-values ≥ 0.480.

5 Discussion

5.1 Why Does Self-Critique Hurt?

Our results extend the findings of Huang et al. [2024] by showing that the degradation from self-critique is *monotonically related to critique stringency*: the harsher the instruction, the worse the performance. The mechanism is asymmetric error introduction. At the adversarial level, the model changes 6.5% of its answers. Of originally correct answers, 5.7% are changed to incorrect, while only a small fraction of incorrect answers are fixed. Since the baseline is at 96% accuracy, the base rate asymmetry—far more correct answers to damage than incorrect answers to fix—makes the net effect strongly negative.

This connects to the key insight of Kamoi et al. [2024]: LLMs cannot find reasoning errors but can correct them given the error location. When instructed to "find errors" with increasing urgency, the model does not become better at detecting real errors—it becomes more willing to *fabricate* errors where none exist. Harsher instructions essentially lower the model's threshold for declaring something an error, increasing false positives without meaningfully improving true positives.

5.2 Why Do Budget Constraints Help?

Our budget control results are consistent with three lines of prior work: TALE [Han et al., 2024] showed explicit budgets improve efficiency and accuracy; CCoT [Nayab et al., 2024] demonstrated redundancy reduction without accuracy loss; and Wu et al. [2025] provided theoretical evidence for an optimal chain-of-thought length shorter than what models naturally produce.

The mechanism is that budget constraints force the model to focus on essential reasoning steps, eliminating verbose filler that can introduce confusion or error. At 75 words, the model must identify the most direct solution path, which for GSM8K problems is often straightforward arithmetic that does not benefit from elaborate exposition.

5.3 The Budget–Critique Interaction

The factorial experiment reveals that budget constraints partially rescue accuracy under harsh critique (88.0% $\rightarrow$ 92.0%). This suggests that critique-induced damage is partly mediated by verbosity—when the model has unlimited space to critique, it generates more spurious objections. Constraining the critique response limits this error amplification. However, even with tight budget constraints, critique still degrades performance relative to no critique at the same budget level (92.0% vs. 96.7%). Budget control is a partial but not complete antidote to critique-induced harm.

5.4 Why Do Alternative Strategies Fail?

The failure of all alternative strategies reveals that GPT-4.1-mini is already near its capability ceiling on GSM8K. At 95.3% baseline accuracy, there is little room for improvement, and any intervention that introduces additional processing risks degrading rather than improving performance.

High-stakes emotional framing has zero effect on mathematical reasoning—the model does not "try harder" in response to emotional pressure. Self-consistency voting over five samples at $T = 0.7$ uses $5\times$ the compute for no gain, because the per-sample accuracy is already so high that disagreements are rare. The explicit rubric creates a critique-like intervention, and consistent with our main finding, shows a slight accuracy decrease.

5.5 Limitations

Single model. We test only GPT-4.1-mini. Results may differ for weaker models (where the baseline is lower and there is more room for improvement) or for reasoning-specialized models such as o1/o3. On weaker models, self-critique might be more beneficial because there are more genuine errors to detect and correct.

Single benchmark. GSM8K represents a specific task type (grade-school math). On tasks where quality is more subjective (dialogue, creative writing, code review), self-critique may behave differently. SELF-REFINE showed improvements on such tasks even when math reasoning did not benefit.

Near-ceiling baseline. At 95–96% accuracy, there is limited headroom. On harder benchmarks (MATH, GPQA), self-critique might be more beneficial because the error rate is higher, providing more opportunities for genuine error detection.

Sample size. With 150–200 problems per experiment, we have limited statistical power for detecting small effects (<3 percentage points). Our significant results ($p < 0.01$ for harsh critique) are robust, but smaller effects may go undetected.

Fixed critique structure. We use single-round generate-then-critique. Multi-round iterative refinement, as in SELF-REFINE, was not tested. It is possible that multiple rounds of gentle critique could accumulate improvements, though Xu et al. [2024] found that self-bias amplifies over iterations.

6 Conclusion

We conducted the first systematic study of critique stringency in LLM self-evaluation, testing five levels from no critique to adversarial across 200 GSM8K problems. Our key findings are:

1. **Harsher self-critique is counterproductive.** Accuracy drops monotonically from 96.0% (no critique) to 91.0% (harsh critique), with the decline significant at $p < 0.01$. The model is more likely to break correct answers than fix incorrect ones.
2. **Budget constraints are the simplest effective intervention.** Asking the model to be concise (75 words) maintains 96.7% accuracy while using 32% fewer tokens.
3. **Budget partially mitigates critique damage.** Tight budget constraints under harsh critique (92.0%) partially recover the accuracy lost from unconstrained harsh critique (88.0%).
4. **No effort-inducing strategy beats the baseline.** High-stakes framing, explicit rubrics, step verification, and self-consistency all fail to improve upon simple chain-of-thought, while using 1.2–$5\times$ more tokens.

The popular intuition that "being harsh with LLMs" improves results is wrong—at least for reasoning tasks. The "lazy LLM" problem is better addressed by constraining output (forcing conciseness) rather than by demanding self-evaluation (which introduces errors). For future work, we recommend investigating these findings on harder benchmarks where the baseline is further from ceiling, on weaker models where there is more room for improvement, and on subjective evaluation tasks where the dynamics of self-critique may differ.

References

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

Tingxu Han, Chunrong Ge, Yuanyuan Fu, Jiawei Jia, Wenqiang Shi, Yiping Zhang, Jian Shi, Deyao Wang, Hai Xiong, and Zheng Li. Token-budget-aware LLM reasoning. *arXiv preprint arXiv:2412.18547*, 2024.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. Large language models cannot self-correct reasoning yet. *arXiv preprint arXiv:2310.01798*, 2024.

Ryo Kamoi, Yusen Sharma, Sarkar Snigdha Sarathi Bhat, Yuqing Quan, Nuo Ding, Yixin Nan, Cornelia Caragea, and Dan Roth. When can LLMs actually correct their own mistakes? a critical survey of self-correction of LLMs. *arXiv preprint arXiv:2406.01297*, 2024.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 36, 2023.

Sania Nayab, Muhammad Nouman Amin, Muhammad Uzair Ghani, Muhammad Awais Aslam, and Qurat Ul Ain. Concise thoughts: Impact of output length on LLM reasoning and cost. *arXiv preprint arXiv:2407.19825*, 2024.

Shaurya Poddar, Shubhanshu Mishra, and Magnus Sahlgren. Brevity is the soul of sustainability. *arXiv preprint arXiv:2506.08686*, 2025.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36, 2023.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2023.

Yuyang Wu, Yifei Xie, Tianyi Liu, Jiaxin Zhang, and Yucheng Wang. When more is less: Understanding chain-of-thought length in LLMs. *arXiv preprint arXiv:2502.07266*, 2025.

Wenda Xu, Guanglei Jiang, Sishuo Mahajan, and Dragomir Radev. Pride and prejudice: LLM amplifies self-bias in self-refinement. *arXiv preprint arXiv:2402.11436*, 2024.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems*, 36, 2023.