The Self-Verification Cliff: Generation Outpaces Self-Selection in Frontier LLMs, and It Widens With Capability

Arihant Jain [1 2]

Abstract

Self-evolving scientific agents must do more than *generate* candidate solutions: they must *select* or *verify* which of their own outputs to trust, because every self-play, self-rewarding, and reinforcement-learning loop that omits an external checker implicitly assumes the model can grade itself. We test that assumption on competition mathematics, where final answers are integers and grading is therefore an exact, noise-free oracle. Across two frontier model families on 86 AIME problems, we find a large and systematic *self-verification cliff*: a model contains a correct answer among $k{=}8$ samples far more often than it can pick that answer using its own judgment. GPT-5.4-mini reaches a correct solution on 54.7% of problems but self-selects correctly on only 41.9%, recovering just 36% of its sampling headroom; Gemini-3.5-flash recovers 43%. Critically, the cliff *widens* with generation ability rather than closing: the stronger generator has the larger absolute gap. We further show that an external verifier helps only when it is at least as capable as the generator—a stronger cross-family verifier closes part of the cliff ($+3.5$ pts), while a weaker one makes selection *worse* than self-selection (-2.3 pts)—and that majority voting fails to rescue selection on these diverse-answer problems. The cliff is a structural ceiling on same-model self-improvement and a concrete argument for external (e.g. formal) verification in self-evolving math agents.

[1]University of Pennsylvania, Philadelphia, Pennsylvania, USA
[2]Arc Institute, Palo Alto, California, USA. Correspondence to: Arihant Jain <arihant.jain@arcinstitute.org>.

The 3rd *AI for Math Workshop* at the 43rd International Conference on Machine Learning (ICML), Seoul, South Korea, 2026.
Copyright 2026 by the author(s).

1. Introduction

The 2026 theme of the AI for Math community—*self-evolving scientific agents*—turns on a single load-bearing capability:[1] a system that improves itself must decide *which of its own outputs are correct*. Self-play theorem provers keep proofs their own checker accepts; self-rewarding pipelines fine-tune on solutions the model judges good; reinforcement learning with verifiable rewards (RLVR) optimizes against a reward signal that, outside narrow string-matchable domains, is increasingly the model's own judgment. In each case the verifier is the load-bearing member of the loop.

A natural and influential cautionary result, *The Limit of RLVR* (Yue et al., 2025), showed that for mathematical reasoning the base model already *contains* correct solutions in its sampling distribution at large k, and that RLVR largely redistributes probability mass within that distribution rather than expanding it. That finding is about *generation*. We ask the complementary question about *selection*: given that the correct answer is somewhere in the model's k samples, how often can the model *find* it using only its own judgment?

We answer this on competition mathematics, where the final answer is an integer in $[0, 999]$ and correctness is therefore an exact, deterministic oracle—no LLM judge, no human grader, no proof checker, and hence no grading noise to confound the measurement. On 86 AIME problems and two frontier model families, we document a large *self-verification cliff*: the gap between what a model can generate (pass@k) and what it can self-select (self-best@k). Our contributions:

1. **The cliff is large.** GPT-5.4-mini recovers only 36% of its $k{=}8$ sampling headroom by self-selection; Gemini-3.5-flash recovers 43% (§4).
2. **The cliff widens with capability.** The stronger generator has the *larger* absolute cliff (0.128 vs. 0.058), mirroring—now across frontier families—the within-family generation/verification asymmetry reported by Chen et al. (2026).
3. **A verifier helps only if it is at least as strong.** A

[1]Code and data: https://github.com/arihantjain4/Self-Verification-Cliff

stronger cross-family verifier closes part of the cliff ($+3.5$ pts); a weaker one *hurts* relative to self-selection (-2.3 pts), refining the cross-family picture of Lu et al. (2025).

4. **Majority voting does not rescue selection** on diverse-answer problems, sitting at or below self-selection and sometimes degrading with k.

Together these give self-evolving-agent designers a concrete rule: a same-model self-improvement loop is bounded by its self-verification accuracy, that bound *tightens* as the generator improves, and escaping it requires a verifier at least as capable as the generator—of which a formal checker is the limiting, capability-independent case.

2. Related Work

Limits of self-improvement and RLVR. Yue et al. (2025) argue RLVR operates within the base model's reasoning support; Shao et al. (2025) document *misevolution*, where self-evolving agents drift in unintended directions. We contribute a mechanism upstream of both: the selection step itself is weak, and weakest where it is most needed.

Generation vs. verification. Chen et al. (2026) show, within the Qwen family, that self-verification does not improve with scale or generation strength, and can regress under generation-focused fine-tuning. We extend this from within-family to *cross-frontier-family*, and frame it as a scaling phenomenon in k. Lu et al. (2025) study 37 models as verifiers on short-answer tasks and report a distribution-similarity bias; we corroborate the directionality (self/similar verifiers over-accept) on a clean integer-answer substrate and tie it to a capability-ordering rule. Naik et al. (2026) find proof verifiers are inconsistent and prompt-sensitive; DeepSeek-AI (2025) motivate their system explicitly by the "generation–verification gap."

Self-consistency and generative verifiers. Majority voting (Wang et al., 2023) and generative verifiers (Zhang et al., 2024) are the standard "free" self-verification methods. We include majority voting as a baseline and find it ineffective on diverse-answer competition problems.

Proof-level evaluation. That answer-level success overstates reasoning quality is established by *Proof or Bluff* (Petrov et al., 2025) and the Open Proof Corpus (Dekoninck et al., 2025). We deliberately restrict to answer-level correctness precisely because it yields a noise-free oracle, isolating the selection deficit from grading uncertainty; the proof-level extension (with a formal Lean oracle) is the natural follow-up (§6).

	GEMINI-3.5-FLASH	GPT-5.4-MINI
pass@1	0.305	0.347
pass@8 (oracle)	0.407	0.547
maj@8	0.314	0.407
self-best@8	0.349	0.419
cliff (pass@8−self)	**0.058**	**0.128**
recovered fraction	43%	36%
cross-select@8 (other family)	0.384	0.395
Δ vs. self	+0.035	−0.023

Table 1. All metrics on the common set (n=86). The stronger generator has the larger cliff and recovers a smaller fraction of its headroom.

3. Method

Data. We use 86 AIME problems (2023–2025) with integer answers, the common set on which all conditions below completed. Correctness is exact integer match against the gold answer.

Models. Two frontier families accessed by API: GEMINI-3.5-FLASH and GPT-5.4-MINI. We chose two capability levels within reach of a small budget to expose the capability dependence of the cliff; the protocol is provider-agnostic.

Generation. For each (model, problem) we draw k=8 chain-of-thought samples at temperature 0.8, each terminating in `\boxed{answer}`.

Metrics. (i) *pass@k*: probability a correct answer appears among k samples, using the unbiased estimator of Chen et al. (2021). (ii) *self-best@k*: the model is shown its own k candidates and selects the one it judges most likely correct (temperature 0); we score the chosen candidate. (iii) *maj@k*: the most frequent boxed answer (self-consistency). (iv) *cross-select@k*: a *different* family selects among the generator's k candidates—an external verifier at matched compute. The *self-verification cliff* is pass@k − self-best@k; the *recovered fraction* is (self-best@k − pass@1)/(pass@k − pass@1), where 1.0 denotes a perfect selector and 0.0 a selector no better than a single sample.

Grading is the oracle. Because answers are integers, every metric above is computed against ground truth with zero grading noise; the only model-internal judgment we measure is the *selection* step.

4. Results

The cliff is large and capability-dependent. Table 1 and Figure 1(a) tell the central story. GPT-5.4-mini's sampling places a correct answer within reach on 54.7% of problems, but its own selector lands on the correct answer only 41.9% of the time—a 12.8-point cliff, recovering only 36% of the headroom between one sample and eight. Gemini-3.5-

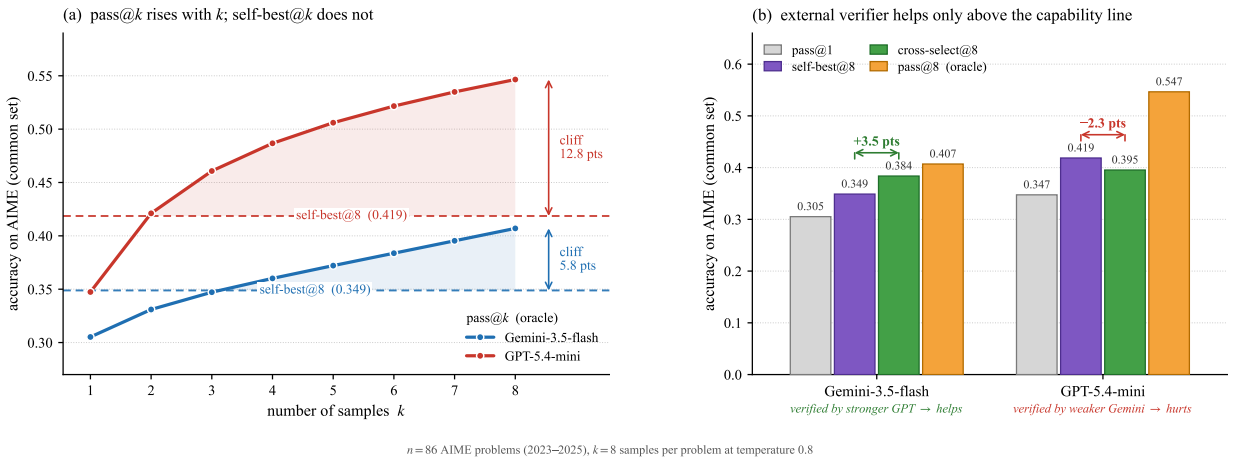

Figure 1. **The self-verification cliff.** *(a)* pass@k (solid) rises with k, but self-best@8 (dashed) lags far behind; the shaded region is the cliff. The stronger generator (GPT-5.4-mini) has the *larger* cliff. *(b)* pass@1, self-best@8, cross-select@8, and the pass@8 oracle ceiling. A *stronger* cross-family verifier helps Gemini (+3.5 pts); a *weaker* one hurts GPT (−2.3 pts).

flash shows a smaller absolute cliff (5.8 pts) and recovers 43%. The model with the *better* generator therefore has the *worse* self-verification efficiency: generation scaled up, self-selection did not keep pace. This is the cross-family analog of the within-family asymmetry of Chen et al. (2026), and it is the opposite of what a designer hoping that "better models verify themselves better" would predict.

A vivid instance. On AIME 2023 problem 20, GPT-5.4-mini produced the correct answer (928) in *five* of its eight samples—so even a plain majority vote would have succeeded—yet its own self-selection chose 64, a minority *wrong* answer. The model overruled its own consensus and discarded a correct solution it had generated repeatedly.

External verification helps only above the capability line. Figure 1(b) and the last rows of Table 1: when the *stronger* GPT verifies the *weaker* Gemini's candidates, selection improves over Gemini judging itself ($0.349 \rightarrow 0.384$, $+3.5$ pts). But when the *weaker* Gemini verifies the *stronger* GPT's candidates, it does *worse* than GPT judging itself ($0.419 \rightarrow 0.395$, -2.3 pts). External verification is not a panacea; it pays off only when the verifier is at least as capable as the generator. This refines the cross-family account of Lu et al. (2025): the relevant axis is verifier capability relative to the generator, not merely distributional dissimilarity.

Majority voting does not rescue selection. maj@8 sits at or below self-best@8 for both models (0.314 vs. 0.349; 0.407 vs. 0.419) and, for Gemini, *degrades* as k grows past five (Figure 1(a), not shown for maj). On AIME's diverse integer answers the sample distribution does not concentrate enough for voting to help, so the common "free" self-verification baseline offers little.

5. Implications for Self-Evolving Agents

Consider an idealized self-improvement loop that, at each round, samples candidates and retains those its own selector accepts. The fraction of retained items that are actually correct is bounded by the selector's accuracy, and the *compounding* of selection error over rounds is governed by 1 − self-verification accuracy. Our measurements imply three things.

First, the loop's per-round yield is capped well below the model's generative reach: GPT-5.4-mini leaves 36% of within-reach problems unselected even with oracle-graded candidates available.

Second—and most consequential for the field's trajectory—this ceiling does *not* relax as base models improve. Because the cliff *widens* with generation ability, scaling the generator without separately scaling verification *increases* the fraction of capability that self-selection wastes. "The model will be smart enough to check itself" is exactly backwards on our data.

Third, the escape route is an external verifier above the capability line, of which a *formal* checker (e.g. a Lean kernel) is the limiting case: it is sound independent of model capability, so it never falls below the line. This is a concrete, measurement-grounded argument for why autoformalization and formal verification remain essential infrastructure for self-evolving math agents—even as natural-language reasoning improves.

6. Limitations

We measure *answer-level* selection, chosen for its noise-free oracle; the selection deficit on *proofs* is plausibly larger and

is the priority follow-up, with Lean providing the oracle (Tsoukalas et al., 2024). We study k=8 and two families due to budget; the qualitative trends (cliff, capability dependence, verifier ordering) are consistent across both families and all k, but absolute numbers will shift with k and model choice. Our self-improvement implications come from a static analysis of selection accuracy rather than a trained loop; instantiating the loop with and without an external verifier is a natural confirmation. Finally, selection is single-turn; allowing the verifier to recompute or use tools may shrink the cliff and is worth measuring.

7. Conclusion

Frontier LLMs can generate correct mathematics they cannot then recognize as correct. This self-verification cliff is large, it widens as generation improves, it is not rescued by majority voting, and it is closed by external verification only when that verifier is at least as capable as the generator. For self-evolving scientific agents, the practical consequence is that verification must be scaled as deliberately as generation—and that capability-independent formal checkers are the safest way to do so.

Impact Statement

This paper studies when language models can verify their own mathematical outputs. By quantifying a limit of self-verification, it cautions against deploying self-improving systems that rely on a model grading itself, and argues for external and formal verification. We see no negative societal impacts beyond those general to improving automated reasoning; if anything, the work encourages more verifiable, auditable agent designs.

References

M. Chen et al. Evaluating large language models trained on code. *arXiv:2107.03374*, 2021.

Y. Chen et al. Learning to self-verify makes language models better reasoners. *arXiv:2602.07594*, 2026.

DeepSeek-AI. DeepSeekMath-V2: Towards self-verifiable mathematical reasoning. *arXiv:2511.22570*, 2025.

J. Dekoninck et al. The Open Proof Corpus: A large-scale study of LLM-generated mathematical proofs. *arXiv:2506.21621*, 2025.

J. Lu et al. When does verification pay off? A closer look at LLMs as solution verifiers. *arXiv:2512.02304*, 2025.

A. Naik et al. Do we need frontier models to verify mathematical proofs? *arXiv:2604.02450*, 2026.

I. Petrov et al. Proof or bluff? Evaluating LLMs on 2025 USA Math Olympiad. *arXiv:2503.21934*, 2025.

S. Shao et al. Your agent may misevolve: Emergent risks in self-evolving LLM agents. *arXiv:2509.26354*, 2025.

G. Tsoukalas et al. PutnamBench: Evaluating neural theorem-provers on the Putnam Mathematical Competition. *NeurIPS Datasets & Benchmarks*, 2024.

X. Wang et al. Self-consistency improves chain of thought reasoning in language models. *ICLR*, 2023.

Y. Yue et al. Does reinforcement learning really incentivize reasoning capacity in LLMs beyond the base model? (The Limit of RLVR). *arXiv:2504.13837*, 2025.

L. Zhang et al. Generative verifiers: Reward modeling as next-token prediction. *arXiv:2408.15240*, 2024.