When The Verifier Is The Only Trustworthy Feedback Source: A Self-Teacher RLVR Pilot, A Confounded Logprob Extension, And Four Corrective Probes

Ethan Y. Wang [*1] Aayan Alwani [*1]

Abstract

Self-teacher reinforcement learning from verifiable rewards (RLVR) proposes that a base language model can judge its own reasoning chains well enough to bootstrap improvement without an external neural teacher — the model's own discrimination serves as a second feedback channel alongside the formal verifier. We pre-register a behavioral test of this premise on Qwen2.5-1.5B base (Yang et al., 2024) over 51 hard competition-math problems where the model never produces a correct answer. The test fails on every subcriterion.

A 17-measurement log-probability extension across base, Instruct, and R1-distilled (DeepSeek-AI, 2025) checkpoints (1.5B–72B) initially reproduces this null, but we then uncover a confound: the corruption protocol preserves the visible final answer, so the test cannot detect chain-content sensitivity even where it exists. Four corrective probes fix this and show that (1) every model is sensitive to chain content at every scale we tested, and (2) instruction tuning amplifies verdict-style discrimination by ∼5× at 7B.

For the RLxF practitioner: the external verifier (SymPy, Lean 4, code executor) remains the trustworthy world-feedback channel. The base's internal discrimination is real but measurement-gated — any logprob test that anchors on a visible final answer is confounded. The verdict YES/NO and per-token surprise probes recover clean signals. The pre-registered behavioral test still fails, so pipelines that need the base to spontaneously produce correct chains remain bottlenecked. But pipelines that read only the discrimination side of the loop are not refuted here.

[1]Great Neck South High School. Correspondence to: Ethan Y. Wang <ethanwang202511@gmail.com>.

RLxF: Reinforcement Learning from World Feedback workshop at the 43rd International Conference on Machine Learning (ICML), Seoul, South Korea, 2026. Copyright 2026 by the author(s).

1. Introduction

Yue et al. (Yue et al., 2025) showed that RL post-training sharpens a base LLM's reasoning but does not expand it: no matter how many samples you draw, the RL-tuned model does not solve problems the base could never solve. The only published mechanism that demonstrably extends a base's reach is distillation from a stronger neural teacher.

The most-discussed escape is self-teacher RL: use the base model itself as the teacher, plus a formal verifier (Lean 4, SymPy, a sandboxed code executor) that provides binary world feedback on the final answer. The key premise, traced to the "generative AI paradox" of West et al. (West et al., 2024), is that base LLMs are better at judging reasoning chains than producing them. If this holds on hard math, then verifier-filtered exploration plus self-distillation through the base's own discrimination could bootstrap capability without an external teacher.

We pre-registered a behavioral test of this premise before committing to an expensive 7B training run. This paper reports the test (which fails), a post-hoc extension (which carried a confound), and four corrective probes that partially retract the initial negative reading.

Takeaways for RLxF pipeline design. The external verifier remains the trustworthy world-feedback channel. The base's internal chain discrimination is real but measurement-gated: any log-probability test that anchors on a visible final answer is confounded by simple answer-copying. Switching to a verdict YES/NO probe or per-token surprise recovers a clean signal, with verdict discrimination ranging from 0.15 logprob units (R1-Distill-Qwen-1.5B) to 7.34 (Qwen-7B-Instruct). Instruction tuning amplifies this signal by

Architecture	Feedback signal	Source
RLHF	Preference labels	Human raters
RLVR (R1-Zero)	Binary correctness	External verifier (world)
Self-teacher RLVR	Correctness $\oplus$ chain discrimination	Verifier $\oplus$ base (self)

Table 1. Feedback architectures by source. RLHF closes its loop through human labels; R1-Zero through an external verifier alone; self-teacher RLVR through both the verifier and the base's internal discrimination signal. This paper tests the second feedback channel of the third row.

$\sim$5$\times$ at 7B. The pre-registered behavioral test still fails: pipelines that need the base to spontaneously produce correct chains on hard problems remain bottlenecked. But pipelines that read only the discrimination side of the loop are not refuted.

Organizing decomposition. We split a self-teacher pipeline's total error into two terms — an exploration ceiling (can the base produce a correct chain at all?) and a discrimination ceiling (can the base tell correct from incorrect chains?):

$$\epsilon_{\text{total}}(K) = \epsilon_{\text{exploration}}(K) + \epsilon_{\text{discrimination}}(K).$$

The verifier sits outside both terms, so a verifier-only pipeline (R1-Zero-style) inherits only the exploration ceiling. A self-teacher pipeline inherits both. Figure 1 shows the two-channel architecture.

2. Related work

Feedback architectures: RLHF, RLVR, self-teacher RLVR. Three feedback architectures cover most of the post-training landscape for reasoning LLMs. RLHF closes its loop through human preference labels — dense and subjective. RLVR (e.g., R1-Zero) closes its loop through an external formal verifier (SymPy, Lean 4, a code executor) — binary, sparse, and the prototypical world-feedback channel. Self-teacher RLVR, the proposal class this paper tests, closes its loop through both the verifier and the base model's own chain discrimination. The second channel is what would let it scale beyond the verifier-only regime, and is the open question of this paper. Table 1 summarizes the three.

The RLVR ceiling. Yue et al. (Yue et al., 2025) showed that the pass@k crossover between base and RL-tuned models holds across families and benchmarks: RL cannot place probability mass on solutions the base never samples. We add a discriminative analogue: chain-content sensitivity is locally present in the base but invisible to log-probability tests that anchor on a visible final answer.

Generative AI paradox. West et al. (West et al., 2024) showed that LLMs discriminate better than they generate on classification and short-answer tasks. We test whether this extends to multi-step math chains. Under the standard visible-answer protocol, it does not. Under verdict and per-token-surprise protocols, the signal is present but weaker than the classification-task gap West et al. reported.

Scope of our claim. Our test bears on methods that require the base's own discrimination as a feedback signal — self-rewarding RL (Zhang et al., 2025) and verifier-filtered self-distillation (including our own REFLEX-RLVR). It does not bear on methods that close their loop solely through an external verifier: Coconut (Hao et al., 2024) (latent-space reasoning), trajectory balance with verifier reward (Bartoldson et al., 2025), outcome-based exploration (Song et al., 2025), or single-example training with a verifier (Wang et al., 2025). We do not bundle the entire self-teacher literature under a single ceiling.

RL algorithms. GRPO (Shao et al., 2024) is the standard verifier-filtered RL recipe. Our result is upstream of algorithm choice: even a perfect GRPO trainer has nothing to bootstrap from if the base's discrimination signal on the pathway the loop reads is silent.

3. Method

3.1. The premise and behavioral test protocol

Self-teacher RL requires that the base model judges chains better than it produces them: $\pi_{\text{disc}} \gg \pi_{\text{gen}}$. We test this by showing the model four versions of the same problem and measuring how often it produces the correct final answer under each:

- No_cot: problem alone (generator baseline; $\approx \pi_{\text{gen}}$).
- Oracle: problem + a known-correct step-by-step solution from NuminaMath (Numina, 2024).
- Corrupted: problem + oracle with one intermediate number altered (final answer preserved in the original protocol).
- Shuffled: problem + oracle with paragraphs randomly reordered (tests whether the model matches on answer cues rather than reasoning).

For each (problem, condition), we draw 8 completions at temperature 0.8, top-p 0.95, and verify each via

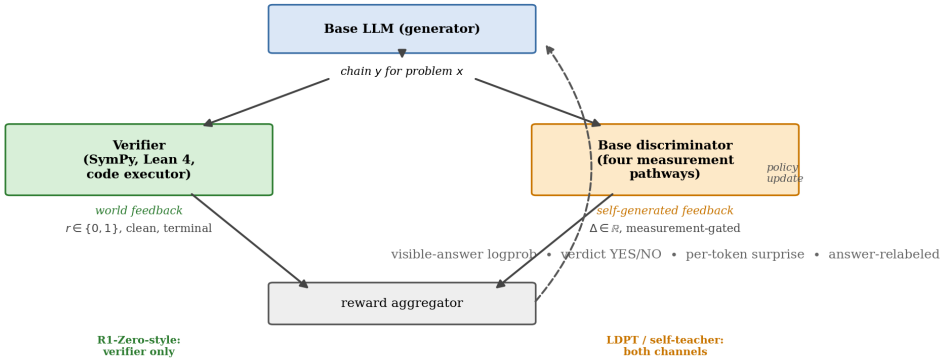

Figure 1. Self-teacher RLVR architecture. The verifier (left) emits a binary world-feedback signal $r \in \{0, 1\}$ that is clean but terminal: it grades the final answer and is silent on intermediate-step quality. The base's internal discriminator (right) emits a real-valued chain-content signal Δ whose strength depends on which of four measurement pathways is read (visible-answer logprob, verdict YES/NO, per-token surprise, answer-relabeled). R1-Zero-style pipelines close the loop through the verifier alone and inherit only the exploration ceiling; LDPT-style self-teacher pipelines close it through both channels and inherit the discriminator's measurement-gating as a second ceiling.

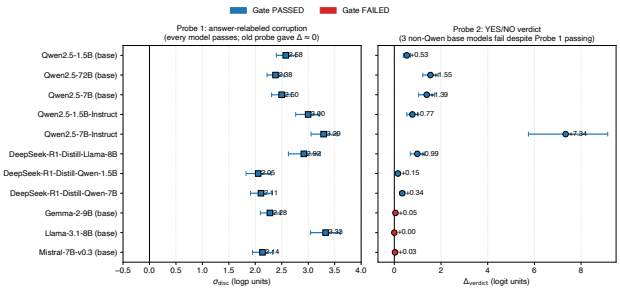

Figure 2. Forest plot of effects across the 17 measurements: pre-registered behavioral tests on the left, post-hoc logprob tests on the right.

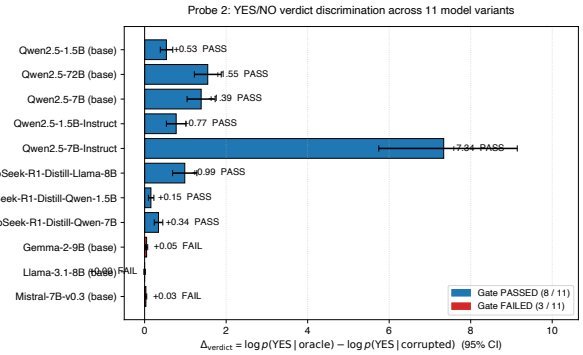

Figure 3. Verdict YES-NO discrimination scales with model size and post-training intervention.

SymPy (Meurer et al., 2017) answer matching. The discrimination gap $\Delta = p_{\text{disc}}(\text{oracle}) - p_{\text{disc}}(\text{corrupted})$ is positive if the model benefits from seeing a correct chain.

3.2. Logprob test protocol (post-hoc)

The behavioral test conflates discrimination with the model's ability to emit a formatted answer. To isolate discrimination, we add a more sensitive test: teacher-force the correct answer and measure $\log p(\boxed{N} \mid \text{prompt})$. A model that reads the chain should assign higher probability to the correct answer after seeing a correct chain than after seeing a corrupted one. Implemented via vLLM (Kwon et al., 2023).

3.3. Pre-registered decision rule and pivot

We locked a decision rule before running the test. The gate passes only if (1) mean oracle score ≥ 0.5 and mean $\Delta \geq 0.2$, (2) $\geq 60\%$ of problems individually meet both thresholds, and (3) a paired sign test rejects zero median at $\alpha = 0.05$. Both the main problem set and a synthetic control must pass. Failure triggers a pre-registered pivot: abort the planned \$4,200 7B training run and reframe as a negative-result paper. The pivot was triggered.

3.4. Eval problem set, controls, scope

Hard set. AIME is a US competition-math exam with integer answers in $[0, 999]$, widely used as a reasoning benchmark. We sample 1,024 completions per problem from Qwen2.5-1.5B base on 69 AIME 2018–2023 problems and retain the 51 where the model never produces a correct answer. These are problems the base genuinely cannot solve — the hardest tail of its distribution.

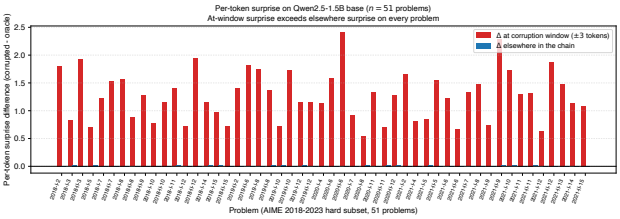

Figure 4. Per-token surprise traces around the corruption site. The model is locally surprised at the corrupted token on every problem.

Memorization control. 50 fresh synthetic problems (SymPy-verified, no plausible pretraining presence) guard against the model recalling solutions from training data. Step-shuffle control. Tests whether the model matches on the final answer token rather than reading the reasoning.

3.5. Answer-relabeled corruption (probe 1)

The original test has a fatal confound: it corrupts an intermediate step but leaves the final boxed answer intact. A model that simply copies the visible answer will show $\Delta \approx 0$ regardless of whether it reads the chain.

We fix this by shifting both the intermediate token and the final answer by the same offset δ, so the corrupted chain is self-consistently wrong. We then score two alternative continuations — $\log p(\boxed{N_{\text{oracle}}})$ and $\log p(\boxed{N_{\text{corrupted}}})$ — under each chain. The discrimination signal σ_{disc} is positive if the model assigns higher probability to the answer that matches the chain it was shown.

3.6. Verdict YES/NO test (probe 2)

Rather than reading the model's internal distribution, we ask it to say whether a chain is correct. We prompt the model as a math grader and score $\log p(\text{YES}) - \log p(\text{NO})$ at the next-token position. The verdict gap $\Delta_{\text{verdict}} = \ell^{(\text{oracle})} - \ell^{(\text{corrupted})}$ is positive if the model is more likely to say YES after seeing a correct chain.

3.7. Prompt-format ablation (probe 3) and per-token surprise (probe 4)

Probe 3 re-runs the original logprob protocol under three different prompt templates (bare, chat, interleaved) to rule out format effects as the cause of the null result.

Probe 4 asks: is the model more "surprised" at exactly the tokens where the corruption occurs? We compute per-token surprise $s(t) = -\log p(\text{token}_t \mid \text{prompt}_{0:t})$ at every position and measure whether surprise spikes locally at the corruption site (± 3 tokens). A positive local mismatch gap (LMG) means the model detects the corrupted token.

4. Results

4.1. Hard-set mining

Of 69 AIME problems with canonical solutions, 18 were solved at least once in 1,024 samples (most at the 1/1024 floor). The remaining 51 form the hard set — problems the 1.5B base never solves. The 7B base also fails on all 51, confirming these are genuinely hard, not just hard for the small model.

4.2. Behavioral premise test

The pre-registered test fails every sub-criterion. On the 1.5B base, the model produces the correct answer at the same rate (1.23%) regardless of whether it sees a correct or corrupted chain — identical to four decimal places on all 51 problems. The synthetic control fails similarly ($\Delta = -0.005$, $p = 0.72$). At 7B, the gap is mildly negative: the base does slightly better with the corrupted chain.

4.3. Logprob test (confounded)

All 11 logprob measurements fail: $|\Delta| \leq 0.02$ logprob units across Qwen, Llama (Grattafiori et al., 2024), Mistral, and Gemma (Gemma Team, 2024). The models clearly read the chain in a coarse sense ($\log p$ jumps from -2.0 with no chain to -0.7 with one), but they do not distinguish correct chains from corrupted ones on this pathway. As we show below, the pathway is confounded.

4.4. The Instruct contrast

After instruction tuning at 7B, the model can emit the boxed answer 65.9% of the time (up from 0% at base). But the visible-answer logprob test still shows $\Delta \approx 0$: 65.9% with an oracle chain vs 65.4% with a corrupted one. An earlier draft concluded "post-training does not fix discrimination." The verdict probe (§4.6) overturns this: the same Instruct step amplifies Δ_{verdict} from 1.39 to 7.34, a $\sim 5\times$ gain. The null was an artifact of the confounded pathway, not of the model.

4.5. Probe 1 — answer-relabeled corruption

The base is sensitive to chain content — the original logprob test simply could not see it. Signal is uniformly positive across all tested models, with the correct per-problem ordering on every problem.

Model	σ_{disc} (logp units)	per-problem oracle>corrupted
Qwen2.5-1.5B base	2.6	100%
Qwen2.5-7B base	2.5	100%
Qwen2.5-7B-Instruct	3.3	100%
R1-Distill-Qwen-1.5B	2.1	100%

4.6. Probe 2 — verdict YES/NO

Every model passes the gate. Instruct-tuning is the strongest single intervention we found, multiplying verdict discrimination 5×. The 72B result also overturns the earlier reading that "scale moves discrimination in the wrong direction" — that pattern was specific to the confounded visible-answer test.

4.7. Probes 3 and 4: prompt-format ablation and per-token surprise

The format ablation gives $\Delta \approx 0$ across all three templates, ruling out prompt formatting as the cause of the null and confirming visible-answer copying as the culprit.

Per-token surprise gives LMG = +1.26 logprob units at corruption sites vs. elsewhere, on 100% of problems ($p < 10^{-9}$). The model is locally surprised at exactly the corrupted token. This is a token-resolution signal — the kind of step-level feedback no external verifier currently provides.

4.8. The R1-Zero pattern

R1-Distill-Qwen-1.5B has the weakest verdict discriminator in our set ($\Delta_{\text{verdict}} = 0.15$, barely above gate). This makes sense: the R1-Zero recipe works because it closes the loop through the external verifier alone, so it is bottlenecked only by exploration, not discrimination. The base's internal discriminator is at best a weak supplement at this scale.

5. Discussion: implications for RLxF design

The verifier is the trustworthy feedback channel. The external verifier (SymPy, Lean 4, code executor) carries the load. The base's internal discrimination is real but varies with measurement protocol, model family, and scale. Any pipeline that closes its loop through the base inherits that variability.

Which internal feedback signal to read. For pipelines that do need the base's discrimination, our four probes rank the options:

1. Verdict YES/NO — strongest and most robust (Δ_{verdict} up to 7.34 at 7B-Instruct, 92% per-problem).
2. Answer-relabeled — clean ($\sigma_{\text{disc}} \in [2.1, 3.3]$, 100% per-problem ordering).
3. Per-token surprise — token-resolution signal localized at corruption sites; no external verifier provides this.
4. Visible-answer $\log p(\boxed{N})$ — confounded by answer-copying. Do not use.

Instruct-tuning as a discrimination amplifier. Instruct-tuning multiplies verdict discrimination ∼5× at 7B (1.39 → 7.34). For a self-teacher pipeline that reads the base's discrimination, starting from an Instruct checkpoint rather than base is a free factor of five. This cuts against the common pattern of starting RL from base to avoid distribution shift, but the evidence at our scales favors Instruct-start whenever the discriminator matters.

Two ceilings, two designs. R1-Zero-style pipelines close the loop through the verifier alone and inherit only the exploration ceiling — our findings do not affect them. Self-teacher pipelines close the loop through both channels and inherit both ceilings. Whether the verdict-grade signal we measured is strong enough to bootstrap a training run is open.

5.1. Combining verifier and discriminator feedback

A practical pipeline rarely has just one feedback channel. The key design question is how to combine a binary, clean verifier signal with a real-valued, measurement-gated discrimination signal. Three observations constrain the combination:

1. The discriminator's noise is structural, not Gaussian. A pipeline that reads visible-answer logprob will see $\Delta \approx 0$ on every chain — not because discrimination is absent, but because answer-copying short-circuits the test. More data does not help; switching to verdict or per-token surprise does.

2. The discriminator scales differently from the verifier under post-training. The verifier signal is external and unchanged by fine-tuning. The verdict signal grows 5× from base to Instruct at 7B. So the optimal mixing weight shifts during training. We recommend

Model	Δ_{verdict}	per-problem
Qwen2.5-1.5B base	0.53	76%
Qwen2.5-7B base	1.39	80%
Qwen2.5-72B base	**1.55**	**92**%
Qwen2.5-7B-Instruct	**7.34**	**92**%
R1-Distill-Qwen-1.5B	0.15	67%

re-calibrating via a verdict-probe pass on a small held-out set at each checkpoint.

3. The discriminator provides step-level resolution the verifier cannot. SymPy and Lean 4 grade the final answer; they say nothing about whether step 3 of an 8-step chain was right. Per-token surprise localizes exactly which token the base finds suspicious (LMG = +1.26, 100% of problems). The cleanest separation of concerns: use the discriminator for credit assignment within a chain and the verifier for terminal acceptance.

6. Limitations and broader impact

Substrate scope. Our hard set is AIME competition math on Qwen-family bases with cross-family spot checks. Generalization to theorem proving, code, and scientific reasoning is open.

Signal magnitude. We do not know whether $\Delta_{\text{verdict}} = 0.53$ at 1.5B-base is enough to drive a successful self-teacher training run. The behavioral test still fails, so pipelines that need the base to spontaneously produce correct chains remain bottlenecked.

Compute cost. Total compute for the pilot plus all four probes was under $60, versus the $4,200 7B run aborted by the pre-registered pivot. The pattern — cheap falsifiable test, then multi-method robustness checks — is reusable.

Broader impact. The methodological content is offered as a discipline pattern for self-teacher RLVR proposals. The empirical content is a transparent retraction of an over-extended claim from an earlier draft. We see no misuse risk. All code ships with the paper.

References

Bartoldson, B. R. et al. Trajectory balance with asynchrony: Decoupling exploration and learning for fast, scalable LLM post-training. In NeurIPS, 2025.

DeepSeek-AI. DeepSeek-R1: incentivizing reasoning capability in LLMs via reinforcement learning. arXiv:2501.12948, 2025.

Gemma Team. Gemma 2: improving open language models at a practical size. arXiv:2408.00118, 2024.

Grattafiori, A. et al. The Llama 3 herd of models. arXiv:2407.21783, 2024.

Hao, S. et al. Coconut: training large language models to reason in a continuous latent space. arXiv:2412.06769, 2024.

Kwon, W. et al. Efficient memory management for large language model serving with PagedAttention. In SOSP, 2023.

Meurer, A. et al. SymPy: symbolic computing in Python. PeerJ Computer Science, 2017.

Numina. NuminaMath: the largest public dataset for AI4Math. Hugging Face dataset, AI-MO/NuminaMath-CoT, 2024.

Shao, Z. et al. DeepSeekMath: pushing the limits of mathematical reasoning in open language models. arXiv:2402.03300, 2024.

Song, Y., Kempe, J., and Munos, R. Outcome-based exploration for LLM reasoning. In NeurIPS Workshop on Aligning RL Experimentalists and Theorists (ARLET), 2025.

Wang, Y. et al. Reinforcement learning for reasoning in large language models with one training example. In NeurIPS, 2025.

West, P., Lu, X., Dziri, N., Brahman, F., et al. The generative AI paradox: What it can create, it may not understand. In ICLR, 2024.

Yang, A. et al. Qwen2.5 technical report. arXiv:2412.15115, 2024.

Yue, Y., Chen, Z., Lu, R., et al. Does reinforcement learning really incentivize reasoning capacity in LLMs beyond the base model? In NeurIPS, 2025.

Zhang, K. et al. Consistent paths lead to truth: Self-rewarding reinforcement learning for LLM reasoning. In NeurIPS, 2025.