When the Evaluator Changes: Measuring False Progress in Self-Improving Language Models

Anonymous authors
Paper under double-blind review

Abstract

Self-improving language models rely on evaluators both to guide learning and to measure progress. When evaluators evolve alongside the models they assess, rising scores may reflect improved behavior, changes in evaluation criteria, or adaptation to evaluator weaknesses. We study false progress: apparent improvement that does not persist when model versions are compared under a diagnostic evaluation held fixed across training. We introduce a retrospective measurement framework based on a cross-version evaluation matrix, in which outputs from successive model versions are evaluated under multiple evaluator versions. This separates changes in model behavior from changes induced by evaluator revision, while a progress decomposition quantifies how much apparent improvement survives under a common reference. To connect measurement failures to training history, we build on **Loom**, a replayable trajectory dataflow that records which supervision signals were consumed by which policy updates. This lets us distinguish three properties that are often conflated: whether measured progress remains valid under evaluator revision, whether a model state descends from disputed supervision, and whether that state remains recoverable after the evaluator is repaired. In controlled reinforcement-learning post-training experiments, we observe cases where reported scores increase while performance under a fixed diagnostic reference deteriorates. We further find that descent from affected supervision does not by itself imply irreversible behavioral failure: some intermediate states remain recoverable even after subsequent training substantially degrades reference performance. These results motivate treating progress measurement, training ancestry, and recovery as distinct questions. More broadly, they suggest that self-improving systems with evolving evaluators should be assessed through versioned, retrospective evaluation rather than contemporaneous scores alone.

1 Introduction

Language models increasingly participate in their own improvement: they generate training data, evaluate candidate responses, and help design post-training experiments (Zelikman et al., 2022; Huang et al., 2023; Yuan et al., 2024; Chen et al., 2024; Yang et al., 2024; Zhang et al., 2026a; Rank et al., 2026; Lim et al., 2026). These feedback loops make evaluation part of the learning process. A verifier patch, revised rubric, or updated reward model can change both what is reported as progress and what the next training update reinforces. Comparing scores across such changes therefore requires specifying what was held fixed.

Reward hacking and overoptimization already show that optimizing a proxy can reduce performance under a stronger reference (Amodei et al., 2016; Skalse et al., 2022; Gao et al., 2022). Evaluator revision adds a historical question: when a better reference becomes available, what can we conclude about progress reported earlier, and about states trained on the earlier signals? Re-evaluating the current model alone does not answer either question. It supplies no paired measurement of the historical outputs, and it does not reveal which disputed signals entered the learner's ancestry.

We distinguish three questions. *Measurement*: does an apparent gain survive evaluation under a fixed reference? *Ancestry*: did a state consume supervision that this reference disputes? *Recovery*:

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

how well does that state learn after the evaluator is repaired? These questions are related but logically different. An affected state can have high reference performance. A state whose current behavior is poor may or may not recover. A reference-unaffected state need not be the best restart point.

Our approach extends Loom's replayable trajectory dataflow with an explicit progress-audit protocol. The essential operation holds a sampled output fixed while changing its evaluator. A matrix of output cohorts and evaluator versions makes apparent gains comparable and exposes ranking reversals. Consumption lineage then answers a separate historical question, linking disputed signals to updates and checkpoints. Neither operation claims to reconstruct how training would have proceeded under a different reward.

The empirical anchor is a controlled evaluator-regression incident in Qwen2.5-Math-1.5B post-training with GRPO. Between updates 60 and 130, the recorded pass rate on training batches increases from 59.06% to 100%, whereas rescoring the same historical outputs gives a decline from 59.06% to zero. Regenerating from the saved weights on 256 fixed held-out questions confirms the reversal: the contemporaneous score gains 32.4 percentage points while fixed-reference accuracy falls by 67.6 points. Yet the intermediate checkpoint after 20 faulty updates, whose ancestry is affected, is observed to retain reference performance and reach a similar recovery endpoint to the pre-fault state; paired intervals do not establish equivalence. The collapsed state shows no held-out recovery after 40 repaired updates. Its stored rewards are constant within every GRPO group, yielding zero reward-driven surrogate gradient despite possible KL and momentum updates. We report these as observations on a small number of related runs, not as evidence about failure prevalence.

The contribution is a measurement framework with explicit scope: definitions of reference-conditioned false progress and excess reported gain; an operational protocol connecting paired rescoring with consumption lineage; and a reproducible retrospective analysis that tests the distinctions against both collapse and successful recovery. The algebra is elementary; its value is to make the audit's estimand, evidence requirements, and non-implications explicit.

2 What counts as progress when evaluation changes?

Setting. At round t, policy π_t generates a cohort B_t of prompt–response pairs $z = (x, y)$. Evaluator e_t supplies scores used in reporting or learning. Learning can update weights, select data, or modify persistent memory; our experiments instantiate weight updates. Let $e_\star$ be a fixed diagnostic reference. It represents the objective of this audit, not guaranteed ground truth. Evaluation includes parsing, prompts, rubrics, thresholds, and tool behavior, not only the judge model's weights.

Let $q_e(z) \in [0, 1]$ be a prespecified comparable score under evaluator e. For binary verification it is the pass indicator. A continuous training reward and a thresholded pass rate are different quantities; we never subtract one from the other. Define the empirical cross-evaluation matrix

$$M_{t,v} = \frac{1}{|B_t|} \sum_{z \in B_t} q_{e_v}(z), \qquad P_t = M_{t,t}, \quad Q_t = M_{t,\star}, \quad b_t = P_t - Q_t. \tag{1}$$

Rows hold outputs fixed while evaluator semantics change; columns hold evaluation fixed while the observed output cohort changes. The diagonal P_t is the historically reported score and Q_t is the retrospective reference score. If evaluator versions repeat, columns can be shared.

Definition 1 (Reference-conditioned false progress)**.** For endpoints $a < b$ and prespecified margins $\epsilon_P, \epsilon_Q \geq 0$, an empirical false-progress event is

$$F_{a,b} = \mathbf{1}\{P_b - P_a > \epsilon_P \ \wedge \ Q_b - Q_a \leq \epsilon_Q\}. \tag{2}$$

Unless specified otherwise, both margins are zero. The term means that reported improvement is not supported by improvement under the audit reference. It is not a claim about the model's intent.

Define *excess reported gain* as $X_{a,b} = (P_b - P_a) - (Q_b - Q_a)$. It measures the discrepancy between two changes, not a proportion of examples or a probability of hacking. It can be negative, and its range is $[-2, 2]$ for scores in $[0, 1]$. In particular, expressing it in percentage points does not constrain it to 100 points.

Proposition 1 (Progress accounting)**.** *For any stored cohorts scored as in Equation 1,*

$$\underbrace{P_b - P_a}_{\textit{apparent gain}} = \underbrace{Q_b - Q_a}_{\textit{reference gain}} + \underbrace{b_b - b_a}_{\textit{change in evaluator gap}}, \qquad X_{a,b} = b_b - b_a. \tag{3}$$

Proof. Substitute $b_t = P_t - Q_t$ and cancel terms. □

This identity requires paired evaluation of each cohort, not that prompts be identical across time. However, interpreting $Q_b - Q_a$ as a policy improvement requires controlling the prompt distribution, decoding, and evaluation protocol. Without that control it is a difference between observed cohorts. Our retrospective results use the latter interpretation. The evaluator gap also changes when a fixed evaluator becomes exploitable under a changing output distribution; it is not exclusively a software-version effect. False progress can occur without any evaluator version change.

Separating evaluator revision from output change. For two comparable evaluators e_0, e_1, define $M_{a,0}, M_{a,1}, M_{b,0}, M_{b,1}$. Then a change from e_0 at a to e_1 at b has the exact decomposition

$$M_{b,1} - M_{a,0} = \underbrace{M_{b,0} - M_{a,0}}_{\text{output-cohort change under } e_0} + \underbrace{M_{b,1} - M_{b,0}}_{\text{revision on final cohort}}. \tag{4}$$

An alternative order first changes the evaluator on B_a. The difference between the two allocations is the interaction $I = (M_{b,1} - M_{b,0}) - (M_{a,1} - M_{a,0})$. Both decompositions describe rescoring fixed outputs. Neither identifies the causal effect of training with e_1 rather than e_0, because changing training rewards changes later outputs. A crossed evaluation matrix is an audit of observed evidence, not a counterfactual learning experiment.

Aggregate score curves alone cannot support either claim: measurement needs paired outputs and ancestry needs consumption records (Appendix B.3).

3 A PROGRESS AUDIT BUILT ON LOOM

Loom supplies the historical evidence needed by the measurement framework (Figure 1). It is a replayable trajectory dataflow that runs beside an unmodified trainer. A sidecar ingests the trainer's native rollout dumps and checkpoints into four linked record types: *trajectories* (prompt, response, task reference, producing checkpoint), *signals* (score value, evaluator identity and version), *updates* (the identifiers of every signal an optimizer step consumed), and *checkpoints* (the update that produced them). Because responses are stored rather than summarized, any evaluator version can be re-applied later without sampling the model again. We add the cross-evaluation and progress-accounting layer; this paper does not introduce a new reinforcement-learning optimizer or a new verifier.

Evidence contract. For each sampled output the audit needs its prompt, response, task reference, producer identity, evaluator version and configuration, original score, and the update identifiers that consumed it. Model and tokenizer identity, decoding settings, and dataset versions are needed to compare rounds. Persisting the actual response is essential: a new sample from the same checkpoint is another observation, not a historical rescore. The existing VeRL integration ingests native rollout evidence into a sidecar; graph construction and diagnosis do not require new language-model sampling.

Measurement stage. Fix the reference, score scale, disagreement rule, cohorts, and endpoints before interpreting results (Appendix A.2). Score stored outputs under the reference and any available historical evaluator versions; compute Equation 1; report ΔP, ΔQ, X, and the sign event F together. Include the old/new crossed matrix when an evaluator changed. Preserve negative as well as positive disagreements: an evaluator that penalizes correct long responses can first underestimate quality before later rewarding incorrect short responses. Reporting only false positives would miss that reversal.

Ancestry stage. Let G be a directed graph with trajectories T, signals S, updates U, and checkpoints C. Edges record production, actual signal consumption, and state ancestry. For a chosen disagreement predicate δ, define

$$D_\star = \{s_i : \delta(s_i, e_\star(T_i)) = 1\}, \qquad A_\star = \{v \in U \cup C : \exists s \in D_\star,\ s \rightsquigarrow_G v\}. \tag{5}$$

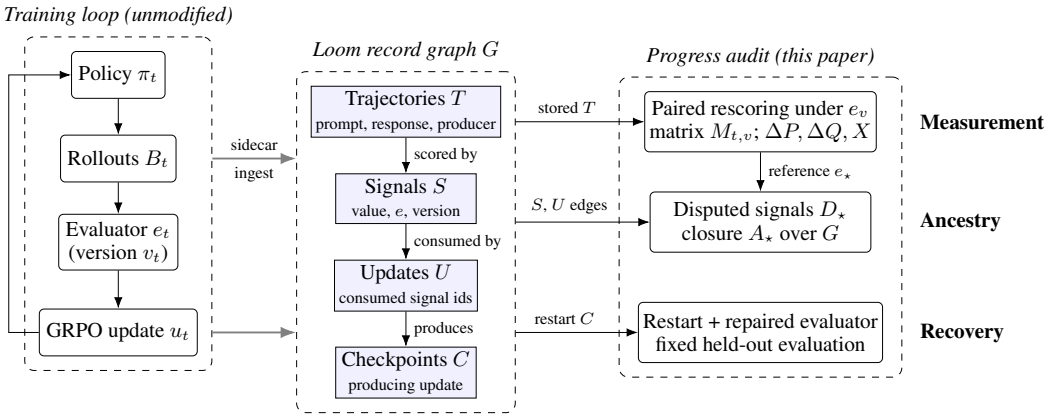


Figure 1: Loom architecture and the progress audit. A sidecar ingests an unmodified trainer's rollouts and checkpoints into a record graph linking each stored response to its evaluator signals, the updates that consumed them, and the resulting checkpoints. The audit answers three separate questions: paired rescoring of stored outputs supports a *measurement* claim; closure over consumption edges supports an *ancestry* claim; and only a new learning experiment, evaluated on fixed held-out questions, supports a *recovery* claim.

A checkpoint outside $A_\star$ is *reference-unaffected*. On a linear training path, Loom returns the latest available such checkpoint. The disagreement predicate is part of this verdict. Our historical audit compares pass/fail decisions at threshold 0.5; it does not certify that every continuous training reward was identical to the reference value.

Proposition 2 (Conditional ancestry guarantee)**.** *If all consumed signals and state-dependency edges are captured, evaluator influence enters through these recorded dependencies, and $D_\star$ contains exactly the signals disputed by the specified rule, a checkpoint outside $A_\star$ has no disputed signal in its recorded ancestry.*

Proof. Any disputed ancestral signal would have a path to the checkpoint, placing the checkpoint in $A_\star$ by Equation 5. □

The guarantee concerns dependency, not parameter difference, harmfulness, or future quality. A disputed signal can have zero effective gradient or a beneficial behavioral consequence. With group-normalized rewards, any change can affect several advantages, so lineage must record the whole consumer group rather than only the example whose verdict changed. A version timestamp is sufficient for a known, synchronous, global switch; it is not sufficient when faulty signals are unused, buffered, or consumed on selected branches. Our real incident has a synchronous switch and does not establish an advantage over timestamp rollback in that setting.

Decision stage. Operators can impose either an ancestry constraint or a performance objective. Under a strict ancestry constraint, choose a reference-unaffected boundary. Under a performance objective, compare recovery from that boundary with affected but promising states using fresh training and held-out evaluation. The latter can rationally retain an affected state. Rescored historical trajectories remain audit evidence; using them for optimization requires a separate treatment of policy mismatch. Relabeling old rewards does not undo the updates those rewards already induced.

4 Experiments

We evaluate four questions: does apparent progress survive a fixed reference; how do evaluator versions change the interpretation of the same outputs; does disputed ancestry predict irrecoverability; and how sensitive are the conclusions to the audit rule? We combine a fresh retrospective audit of two recorded training incidents with fixed-prompt evaluation of saved states.

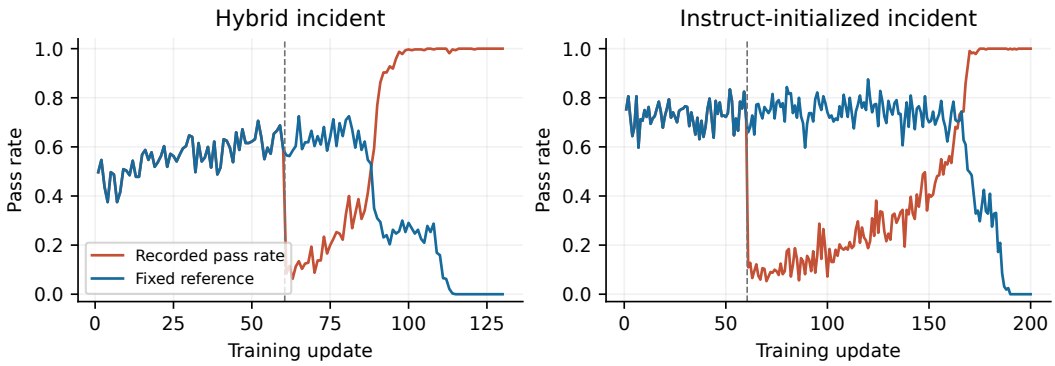


Figure 2: Historical training cohorts. The dashed line marks the evaluator change after update 60. Recorded pass rate eventually saturates while fixed-reference performance collapses in both incidents. Points are batches of 320 responses; variation across batches is not an across-seed confidence interval.

4.1 Experimental design

Training incidents. Both incidents use Qwen2.5-Math-1.5B (Yang et al., 2024), GRPO (Shao et al., 2024; Guo et al., 2025), VeRL/HybridFlow (Sheng et al., 2025), and MATH (Hendrycks et al., 2021). Each update samples five responses for each of 64 prompts. Updates 1–60 use the reference reward; subsequent updates use an injected length-biased evaluator, modeling the length bias documented in learned rewards and judges (Singhal et al., 2024; Dubois et al., 2024). The *hybrid* incident initializes from Base, then resumes with Instruct configuration for the faulty phase and recovery. It contains 130 updates and 41,600 responses. The *Instruct-initialized* incident contains 200 updates and 64,000 responses. Both use seed 42; they are related initialization cases, not independent seed replications. The injected change is a controlled stress test, not a claim that the model autonomously modified its evaluator.

Evaluators. The diagnostic reference extracts the final boxed answer and checks equality after the VeRL MATH grader's string normalization. It is an executable reference with known scope, not human ground truth or a general theorem prover; rule-based verifiers have known false negatives (Huang et al., 2026). For a response y, the injected reward is

$$r_{\text{len}}(y) = \begin{cases} \max(0.1, 1 - |y|/1500), & \text{if } y \text{ contains } \texttt{\textbackslash boxed\{}, \\ 0, & \text{otherwise}, \end{cases} \tag{6}$$

where $|y|$ counts characters. Reported pass rate thresholds this reward at 0.5; it is not the continuous mean reward. We cross-evaluate every stored response under five rules: normalized reference, exact boxed-answer match, boxed-answer presence, boxed presence with fewer than 400 characters, and length-reward pass. Only the normalized reference and length reward were training evaluators in these incidents. The other columns are retrospective sensitivity probes. Fresh execution of the reference reproduces every cached label across all 105,600 incident responses.

Two evaluation populations. Historical cohorts preserve what training actually generated and consumed, but their prompts change across updates. They support retrospective score accounting, not an unconfounded estimate of checkpoint generalization. Our fixed-prompt experiment instead uses the same sampled MATH test questions and decoding settings for every checkpoint. Its design was recorded before new generation, after the historical curves were known. We assess the pre-fault, intermediate, collapsed, and recovered states separately; the appendix gives the selection rule and configuration.

4.2 Apparent gains can reverse under a fixed reference

Between updates 60 and 130 in the hybrid incident, reported pass rate rises from 59.06% to 100%, while reference pass rate falls from 59.06% to zero (Figure 2). Equation 3 gives an apparent gain of

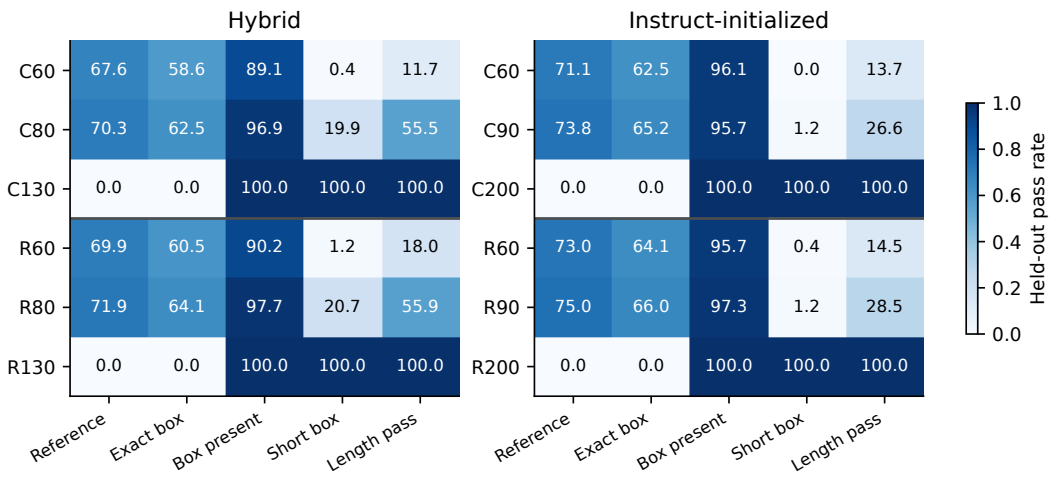


Figure 3: Cross-evaluation matrix on 256 fixed MATH test questions with identical prompting and greedy decoding for every saved state. Rows above the line are training checkpoints (Ct = after update t); rows below are endpoints of 40 repaired-reference updates started from the corresponding checkpoint (Rt). Cell labels are pass rates (%).

$+40.94$ percentage points (pp), a reference change of -59.06 pp, and excess reported gain of 100 pp. For the Instruct-initialized case, updates 60 to 200 give $+28.44$, -71.56, and 100 pp, respectively. These are within-incident endpoint descriptions, not estimates of a population rate of false progress.

The chosen interval matters. Restricting the hybrid comparison to updates 61–130, when the faulty evaluator is already fixed, gives $+91.56$ pp apparent gain and -56.56 pp reference change. Excess gain is 148.13 pp: a difference of changes, not an error rate. Thus evaluator revision initiates this stress test, but later false progress also occurs without further revision.

4.3 Cross-evaluation exposes the source and direction of disagreement

Figure 5 (Appendix D) evaluates identical outputs under every rule, so within-row differences cannot result from changed generations. On hybrid update 60, switching only the scoring rule from reference to length-pass changes 59.06% to 11.88%, a -47.19 pp measurement effect. On update 130, the same evaluator switch changes 0% to 100%. Their interaction is 147.19 pp: the evaluator gap depends strongly on which output cohort it scores. An additive correction estimated before the fault would therefore mischaracterize the final cohort.

At hybrid update 80, the recorded pass rate is 32.50%, while the reference passes 70.94%. Among the 320 outputs, 40.00% are reference-positive but proxy-negative, and 1.56% show the opposite disagreement. By update 130, all outputs are proxy-positive and reference-negative. Exact boxed-answer matching also yields zero at collapse, whereas the formatting and brevity rules all yield 100%. This sensitivity analysis establishes that the endpoint reversal is not specific to one normalization convention; it does not establish robustness to arbitrary stronger references.

4.4 The reversal persists on fixed held-out questions

Historical cohorts change prompts at every update. We therefore regenerate responses from the saved weights on one fixed question set (Figure 3; protocol in Appendix A). All 256 questions complete at every state, and intervals below are paired question-bootstrap 95% intervals.

Table 1 reproduces the historical pattern without prompt drift. The final faulty states answer every question with the four-token string `\boxed{}`, so formatting, brevity, and length-pass rules all score 100% while both answer references score zero. The reference loss is not an artifact of normalization: exact boxed-answer matching drops from 58.6% to zero (hybrid) and from 62.5% to zero (Instruct).

Table 1: Progress accounting on fixed held-out questions (pp). "Apparent" follows the contemporaneous schedule: reference at C60, length-pass at the final faulty state. Brackets are paired question-bootstrap 95% intervals, conditional on the trained states.

Contrast	Hybrid (C60→C130)	Instruct (C60→C200)
Apparent gain ΔP	$+32.4$ [$+26.6$, $+38.3$]	$+28.9$ [$+23.4$, $+34.4$]
Reference change ΔQ	-67.6 [-73.4, -61.7]	-71.1 [-76.6, -65.6]
Excess reported gain X	$+100.0$	$+100.0$
Intermediate state	C80	C90
Length-pass change vs. C60	$+43.8$ [$+37.5$, $+50.0$]	$+12.9$ [$+8.6$, $+17.6$]
Reference change vs. C60	$+2.7$ [-2.3, $+7.8$]	$+2.7$ [-1.2, $+6.6$]
Recovery endpoint R(mid) − R60	$+2.0$ [-3.1, $+6.6$]	$+2.0$ [-2.0, $+5.9$]
Recovery endpoint R(mid) − R(final)	$+71.9$ [$+66.4$, $+77.3$]	$+75.0$ [$+69.5$, $+80.5$]

Table 2: Historical recovery batches after 40 fresh updates with the repaired reference. "Mean" pools all 12,800 recovery responses; "final" is the last 320-response training batch, not a held-out endpoint. All arms are single runs.

Incident	Restart	Affected ancestry?	Mean (%)	Final (%)
Hybrid	C60	no	62.95	62.50
Hybrid	C80	yes	65.01	76.25
Hybrid	C130	yes	0.04	0.00
Instruct	C60	no	73.82	73.44
Instruct	C90	yes	75.59	80.94
Instruct	C200	yes	0.04	0.00

The intermediate states separate ancestry from damage on held-out data. After 20 (hybrid) and 30 (Instruct) updates trained on the faulty evaluator, the proxy score has already risen substantially and responses are shorter (mean 525→336 and 502→437 tokens), yet estimated reference accuracy rises by only 2.7 pp, with paired intervals including zero. These observations establish neither equivalence to C60 nor the mechanism of shortening. The fixed-reference column distinguishes retained observed performance from the later collapse.

The recovery endpoints make the same point for learnability. Forty repaired updates from the affected intermediate state yield similar observed held-out accuracy to forty from the reference-unaffected boundary (71.9% vs. 69.9% hybrid; 75.0% vs. 73.0% Instruct; paired differences include zero). These intervals do not establish recovery equivalence. Recovery from the collapsed state yields byte-identical outputs on all 256 questions before and after the 40 repaired updates, in both incidents. Section 4.5 examines the available learning signal.

4.5 Affected ancestry and recovery are different properties

The historical consumption audit identifies update 61 as the first affected update and C60 as the last reference-unaffected boundary. In the hybrid case, 15,560 pass/fail-disputed responses reach 70 subsequent updates and seven checkpoint records. Yet the update-80 reference score exceeds the update-60 score by 11.88 pp. Affected ancestry therefore does not imply low observed reference performance. Conversely, this score difference alone is not evidence that faulty training improved generalization.

In training batches (Table 2), recovery from the collapsed checkpoints stays at zero while affected intermediate states recover, rejecting a blanket equation of affectedness with irrecoverability. Starts, data order, and budgets differ across these single-run arms, so they do not rank restart policies in expectation.

A control: disputed ancestry with retained performance. An archived third run from the same Instruct-initialized C60 replaces the reference with a thresholded brevity rule (reward 1 for a boxed answer under 400 characters, 0.1 for a longer boxed answer, 0 otherwise) for updates 61–100. In

this control, 9,373 of its 12,800 consumed signals (73%) are pass/fail disputed, and all 40 updates lie in the closure. Rescoring its stored outputs shows little change in aggregate reference performance. Reference accuracy is 74.7% over the last ten clean updates and 74.1% over updates 91–100, with mean length 1,373 and 1,371 characters. Regenerating on the fixed held-out questions agrees: after 40 affected updates the model scores 73.4% under the reference versus 70.7% at C60 (paired difference $+2.7$ pp, 95% interval $[0.0, +5.9]$), and only 0.4% of its answers satisfy the faulty rule.[1] Only 15.2% of historical groups have non-uniform reward, primarily separating boxed from unboxed answers (Appendix D). This control shows that a high dispute count can coexist with retained reference performance: 178 of 256 response strings nevertheless change.

Reward signal during unsuccessful recovery. GRPO normalizes rewards within each group of five responses to a prompt; a group with equal rewards contributes zero reward-driven surrogate gradient, which is why DAPO's dynamic sampling filters such groups (Yu et al., 2025). From C60 and the intermediate states, 20–44% of recovery groups carry non-zero advantage (Appendix D); from the collapsed states, *none* of the 2,560 groups does, in either incident. Almost every response is `\boxed{}`; the single positive-reward group per arm matches an empty stored target, not correct reasoning. KL loss and optimizer momentum can still change parameters, so this is not an absorbing state, but evaluator repair alone did not restore performance within 40 updates. Uniform group rewards, a property of the sampled behavior and learning rule, explain this; ancestry alone cannot reveal it.

The hybrid C80 arm carries a post-hoc validator repair (Appendix C); the Instruct C90 arm independently shows affected-but-recoverable behavior.

The audit predicate changes the scope of affectedness. Pass/fail disputes (15,560 hybrid; 26,535 Instruct) understate numeric reward differences (21,640; 43,808), and thresholds $0.25/0.75$ give 14,311/16,780 hybrid disputes. A pass/fail-unaffected designation therefore does not certify identical optimizer rewards; the reference, score transformation, tolerance, and consumption rule must accompany every ancestry claim (Table 4).

5 Related work

Self-improvement and feedback loops. Models increasingly supply their own training signal: bootstrapped rationales (Zelikman et al., 2022; Huang et al., 2023), self-play (Chen et al., 2024), AI or self-generated rewards (Bai et al., 2022; Yuan et al., 2024; Wu et al., 2025), and model-generated data with reward modeling (Yang et al., 2024). Recent systems co-evolve the policy with its rubrics, judges, or internal rewards (Li et al., 2026; Wang et al., 2026b;a), and agents modify their own code or run post-training and research loops (Zhang et al., 2026a; Lu et al., 2024; Rank et al., 2026; Lim et al., 2026; Chen et al., 2026). Duan et al. (2026) argue that genuine recursive self-improvement must retain changes that improve later improvement; judging this presupposes that progress is measured comparably across rounds, while the evaluator that certifies progress can itself change during the process. Recursive training on model outputs can also degrade models over generations (Shumailov et al., 2024). Our experiments isolate an inner RL post-training loop with an injected evaluator change; they do not evaluate an autonomous outer agent or a jointly trained learner–judge population.

Reward hacking and evaluator dependence. Optimizing a misspecified or imperfect proxy can reduce performance under the intended objective (Amodei et al., 2016; Manheim & Garrabrant, 2018; Skalse et al., 2022; Pan et al., 2022), including reward-model overoptimization in RLHF (Christiano et al., 2017; Ouyang et al., 2022; Gao et al., 2022; Coste et al., 2024) and length exploitation (Singhal et al., 2024). In RL with verifiable rewards (Lambert et al., 2025; Guo et al., 2025), rule-based verifiers miss equivalent answers while model-based verifiers can be gamed (Huang et al., 2026; Helff et al., 2026), leaky test suites reward incorrect programs (Zhang, 2026), and even spurious rewards can move benchmark scores (Shao et al., 2025). Reward hacking appears in frontier reasoning models (Baker et al., 2025), generalizes to reward tampering in controlled curricula (Denison et al., 2024), and arises in self-refinement without weight updates (Pan et al., 2024).

[1] This pair was generated on a different GPU model from Table 1; greedy decoding is not bit-reproducible across hardware (C60: 70.7% here vs. 71.1%), so we compare only states generated on the same device.

We do not claim that proxy/reference divergence is new. Our focus is a retrospective protocol that combines evaluator-version comparability with explicit ancestry and recovery semantics.

Changing criteria and judge reliability. LLM judges exhibit position, length, and self-preference biases (Zheng et al., 2023; Wang et al., 2024; Dubois et al., 2024); EvalGen identifies criteria drift as people refine requirements while inspecting outputs (Shankar et al., 2024), and rubric edits can propagate through post-training (Ding et al., 2026). Evolving benchmarks and holistic evaluation make versioned evaluation explicit (Kiela et al., 2021; Liang et al., 2023), and measurement theory separates a construct from its operationalization (Jacobs & Wallach, 2021); concept drift studies the analogous change in data distributions (Gama et al., 2014). Cross-version rescoring is complementary to improving judge quality: a stronger judge does not by itself explain which earlier scores or consumed signals it overturns.

Training provenance and failure management. Data provenance (Buneman et al., 2001), RDD lineage (Zaharia et al., 2012), model management (Vartak et al., 2016), and hindsight logging (Garcia et al., 2021) preserve the information needed for later recomputation and inspection. RFT-FM integrates anomaly detection, diagnosis, and remediation for RL fine-tuning (Zhang et al., 2026b). Loom's trajectory and consumption records supply the audit substrate used here. The extension is cross-version progress measurement and its empirical separation from ancestry and recovery, rather than a claim that telemetry or lineage alone establishes behavioral correctness.

6 Discussion and limitations

Our results support reporting a fixed-reference column alongside the contemporaneous training score, and publishing crossed scores when an evaluator changes. The diagnostic reference can itself be wrong, and a deliberate change of objective is a tradeoff rather than a defect; false progress is always relative to the declared reference objective.

The evidence is deliberately bounded. Both incidents concern one model family, mathematical final-answer checking, and an injected brevity reward. They share a nominal training seed and do not establish prevalence, scaling, or behavior of co-evolving learned judges. The primary incident's Base/Instruct configuration change limits causal interpretation. Historical batch curves confound policy and prompt composition; fixed-prompt evaluation addresses that measurement issue but does not reproduce alternative training histories. Paired prompt uncertainty cannot replace training-seed replication, which we do not report. The short-box control is a single archived run analysed retrospectively.

Ancestry closure is conservative: it identifies dependencies, not the sign or magnitude of their effect, and its completeness depends on captured consumption. The experiments neither show rollback superiority over a known synchronous fault timestamp nor that affected states must be discarded; combining ancestry constraints with recovery estimates is a natural next step.

7 Conclusion

When evaluators change, contemporaneous scores alone cannot establish self-improvement. Paired cross-version evaluation tests measurement, consumption lineage identifies disputed supervision, and recovery experiments test whether a state remains useful after repair. Scores can improve while reference performance collapses, yet affected states can remain recoverable; progress claims should state which property was measured.

Reproducibility statement

The accompanying analysis contains aggregate per-update measurements, source hashes, the metric implementation, and figure-generation commands. Historical reference labels are checked by fresh execution of the grader. Appendix A gives the audit contract, experimental configuration, and unresolved provenance limitations. The fixed-prompt evaluation and the historical training experiments are reported separately.

AI USE STATEMENT

We used AI assistance in four ways. (1) Writing: to aid and polish the writing of the manuscript. (2) Retrieval and discovery: to find and organize related work. (3) Research ideation and execution: to help develop the experimental design and to implement and run the analysis and evaluation scripts. (4) Drafting: to draft sections of the paper, which the authors revised. Reported empirical values come from recorded experiment artifacts and executable analyses, not generated estimates. The authors verified the sources, claims, and experimental evidence and take full responsibility for the content of the paper.

ETHICS STATEMENT

The study uses controlled verifier faults on mathematical tasks to examine misleading progress signals. It does not deploy faulty evaluators to users or infer malicious intent from model behavior. Affected ancestry is explicitly not a judgment of model trustworthiness beyond the chosen reference and capture assumptions.

REFERENCES

Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in AI safety. arXiv:1606.06565, 2016.

Yuntao Bai et al. Constitutional AI: Harmlessness from AI feedback. arXiv:2212.08073, 2022.

Bowen Baker, Joost Huizinga, Leo Gao, Zehao Dou, Melody Y. Guan, Aleksander Madry, Wojciech Zaremba, Jakub Pachocki, and David Farhi. Monitoring reasoning models for misbehavior and the risks of promoting obfuscation. arXiv:2503.11926, 2025.

Peter Buneman, Sanjeev Khanna, and Wang-Chiew Tan. Why and where: A characterization of data provenance. In *International Conference on Database Theory*, 2001.

Mingguang Chen, Licheng Wang, and Bo Qu. Recursive self-improvement in AI: From bounded self-refinement to autonomous research loops. arXiv:2607.07663, 2026.

Zixiang Chen, Yihe Deng, Huizhuo Yuan, Kaixuan Ji, and Quanquan Gu. Self-play fine-tuning converts weak language models to strong language models. arXiv:2401.01335, 2024. URL https://arxiv.org/abs/2401.01335.

Paul F. Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*, 2017.

Thomas Coste, Usman Anwar, Robert Kirk, and David Krueger. Reward model ensembles help mitigate overoptimization. In *International Conference on Learning Representations*, 2024.

Carson Denison et al. Sycophancy to subterfuge: Investigating reward-tampering in large language models. arXiv:2406.10162, 2024.

Ruomeng Ding, Yifei Pang, He Sun, Yizhong Wang, Zhiwei Steven Wu, and Zhun Deng. Rubrics as an attack surface: Stealthy preference drift in LLM judges. arXiv:2602.13576, 2026. URL https://arxiv.org/abs/2602.13576.

Rotem Dror, Gili Baumer, Segev Shlomov, and Roi Reichart. The hitchhiker's guide to testing statistical significance in natural language processing. In *Annual Meeting of the Association for Computational Linguistics*, 2018.

Yi Duan et al. The last AI built by humans: Toward genuine recursive self-improvement. arXiv:2609.11873, 2026.

Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B. Hashimoto. Length-controlled AlpacaEval: A simple way to debias automatic evaluators. In *Conference on Language Modeling*, 2024.

Bradley Efron and Robert J. Tibshirani. *An Introduction to the Bootstrap*. Chapman & Hall, 1993.

João Gama, Indrė Žliobaitė, Albert Bifet, Mykola Pechenizkiy, and Abdelhamid Bouchachia. A survey on concept drift adaptation. *ACM Computing Surveys*, 46(4), 2014.

Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization. arXiv:2210.10760, 2022. URL https://arxiv.org/abs/2210.10760.

Rolando Garcia et al. Hindsight logging for model training. *Proceedings of the VLDB Endowment*, 14(4), 2021.

Daya Guo et al. DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. *Nature*, 645:633–638, 2025.

Lukas Helff, Quentin Delfosse, David Steinmann, Ruben Härle, Hikaru Shindo, Patrick Schramowski, Wolfgang Stammer, Kristian Kersting, and Felix Friedrich. LLMs gaming verifiers: RLVR can lead to reward hacking. arXiv:2604.15149, 2026. URL https://arxiv.org/abs/2604.15149.

Dan Hendrycks et al. Measuring mathematical problem solving with the MATH dataset. In *NeurIPS Datasets and Benchmarks Track*, 2021.

Jiaxin Huang, Shixiang Shane Gu, Le Hou, Yuexin Wu, Xuezhi Wang, Hongkun Yu, and Jiawei Han. Large language models can self-improve. In *Conference on Empirical Methods in Natural Language Processing*, 2023.

Yuzhen Huang, Weihao Zeng, Xingshan Zeng, Qi Zhu, and Junxian He. From accuracy to robustness: A study of rule- and model-based verifiers in mathematical reasoning. In *Conference on Empirical Methods in Natural Language Processing*, 2026.

Abigail Z. Jacobs and Hanna Wallach. Measurement and fairness. In *ACM Conference on Fairness, Accountability, and Transparency*, 2021.

Douwe Kiela et al. Dynabench: Rethinking benchmarking in NLP. In *Conference of the North American Chapter of the Association for Computational Linguistics*, 2021.

Philipp Koehn. Statistical significance tests for machine translation evaluation. In *Conference on Empirical Methods in Natural Language Processing*, 2004.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with PagedAttention. In *ACM Symposium on Operating Systems Principles*, 2023.

Nathan Lambert et al. Tulu 3: Pushing frontiers in open language model post-training. In *Conference on Language Modeling*, 2025.

Shuyue Stella Li, Rui Xin, Teng Xiao, Yike Wang, Rulin Shao, Zoey Hao, Melanie Sclar, Sewoong Oh, Faeze Brahman, Pang Wei Koh, and Yulia Tsvetkov. EvoLM: Self-evolving language models through co-evolved discriminative rubrics. arXiv:2605.03871, 2026.

Percy Liang et al. Holistic evaluation of language models. *Transactions on Machine Learning Research*, 2023.

Joy Jia Yin Lim, Xin Huang, Hao Peng, Yaxi Lu, Xin Cong, Zhong Zhang, Maosong Sun, and Yankai Lin. What is missing from AI post-training AI: An empirical analysis. arXiv:2608.19072, 2026.

Chris Lu, Cong Lu, Robert Tjarko Lange, Jakob Foerster, Jeff Clune, and David Ha. The AI scientist: Towards fully automated open-ended scientific discovery. arXiv:2408.06292, 2024.

David Manheim and Scott Garrabrant. Categorizing variants of Goodhart's law. arXiv:1803.04585, 2018.

Long Ouyang et al. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, 2022.

Alexander Pan, Kush Bhatia, and Jacob Steinhardt. The effects of reward misspecification: Mapping and mitigating misaligned models. In *International Conference on Learning Representations*, 2022.

Jane Pan, He He, Samuel R. Bowman, and Shi Feng. Spontaneous reward hacking in iterative self-refinement. arXiv:2407.04549, 2024. URL https://arxiv.org/abs/2407.04549.

Ben Rank, Hardik Bhatnagar, Ameya Prabhu, Shira Eisenberg, Karina Nguyen, Matthias Bethge, and Maksym Andriushchenko. Posttrainbench: Can LLM agents automate LLM post-training? arXiv:2603.08640, 2026.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. arXiv:1707.06347, 2017.

Shreya Shankar, J. D. Zamfirescu-Pereira, Björn Hartmann, Aditya G. Parameswaran, and Ian Arawjo. Who validates the validators? aligning LLM-assisted evaluation of LLM outputs with human preferences. arXiv:2404.12272, 2024. URL https://arxiv.org/abs/2404.12272.

Rulin Shao, Shuyue Stella Li, Rui Xin, Scott Geng, Yiping Wang, Sewoong Oh, Simon Shaolei Du, Nathan Lambert, Sewon Min, Ranjay Krishna, Yulia Tsvetkov, Hannaneh Hajishirzi, Pang Wei Koh, and Luke Zettlemoyer. Spurious rewards: Rethinking training signals in RLVR. arXiv:2506.10947, 2025.

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

Guangming Sheng et al. Hybridflow: A flexible and efficient RLHF framework. In *EuroSys*, 2025.

Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Nicolas Papernot, Ross Anderson, and Yarin Gal. AI models collapse when trained on recursively generated data. *Nature*, 631:755–759, 2024.

Prasann Singhal, Tanya Goyal, Jiacheng Xu, and Greg Durrett. A long way to go: Investigating length correlations in RLHF. In *Conference on Language Modeling*, 2024.

Joar Skalse, Nikolaus H. R. Howe, Dmitrii Krasheninnikov, and David Krueger. Defining and characterizing reward hacking. In *Advances in Neural Information Processing Systems*, 2022.

Manasi Vartak et al. ModelDB: A system for machine learning model management. In *Workshop on Human-In-the-Loop Data Analytics*, 2016.

Beining Wang, Weihang Su, Hongtao Tian, Hao Kong, Tao Yang, Ting Yao, Qingyi Pan, Yueyue Wu, Qingyao Ai, Min Zhang, and Yiqun Liu. Co-evolving LLM evaluators and policies via DynamicRubric. arXiv:2607.20083, 2026a.

Peiyi Wang et al. Large language models are not fair evaluators. In *Annual Meeting of the Association for Computational Linguistics*, 2024.

Xinyu Wang, Hanwei Wu, Jingwei Song, Shuyuan Zhang, Jiayi Zhang, Fanqi Kong, Tung Sum Thomas Kwok, Xiao-Wen Chang, Yuyu Luo, Chenglin Wu, and Bang Liu. Co-evolution of policy and internal reward for language agents. arXiv:2604.03098, 2026b.

Tianhao Wu, Weizhe Yuan, Olga Golovneva, Jing Xu, Yuandong Tian, Jiantao Jiao, Jason Weston, and Sainbayar Sukhbaatar. Meta-rewarding language models: Self-improving alignment with LLM-as-a-meta-judge. In *Conference on Empirical Methods in Natural Language Processing*, 2025.

An Yang et al. Qwen2.5-Math technical report: Toward mathematical expert model via self-improvement. arXiv:2409.12122, 2024.

Qiying Yu et al. DAPO: An open-source LLM reinforcement learning system at scale. arXiv:2503.14476, 2025.

Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Xian Li, Sainbayar Sukhbaatar, Jing Xu, and Jason Weston. Self-rewarding language models. In *International Conference on Machine Learning*, 2024.

Matei Zaharia, Mosharaf Chowdhury, Tathagata Das, Ankur Dave, Justin Ma, Murphy McCauley, Michael J. Franklin, Scott Shenker, and Ion Stoica. Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing. In *NSDI*, 2012.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah D. Goodman. STaR: Bootstrapping reasoning with reasoning. In *Advances in Neural Information Processing Systems*, 2022.

Chuyifei Zhang. When the reward suite is leaky: A preregistered causal contrast of natural verifier false positives in RLVR. arXiv:2607.11022, 2026.

Jenny Zhang, Shengran Hu, Cong Lu, Robert Lange, and Jeff Clune. Darwin Gödel Machine: Open-ended evolution of self-improving agents. In *International Conference on Learning Representations*, 2026a.

Lingzhe Zhang, Tong Jia, Yunpeng Zhai, Liancheng Fang, Kening Zheng, Hongyi Liu, Xiaosong Huang, Philip S. Yu, and Ying Li. Towards robust LLM post-training: Automatic failure management for reinforcement fine-tuning. arXiv:2605.04431, 2026b. URL https://arxiv.org/abs/2605.04431.

Yanli Zhao et al. PyTorch FSDP: Experiences on scaling fully sharded data parallel. *Proceedings of the VLDB Endowment*, 16(12), 2023.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-judge with MT-Bench and chatbot arena. arXiv:2306.05685, 2023. URL https://arxiv.org/abs/2306.05685.

A AUDIT PROTOCOL AND REPRODUCIBILITY DETAILS

A.1 SCOPE OF THE EVIDENCE

The two incident trajectories and six recovery runs predate this paper's new cross-version analysis. The retrospective contribution is their paired rescoring, common metric definitions, reference sensitivity, and separation of ancestry from recovery. The fixed-prompt evaluation generates new outputs from saved weights. It is not a new training run or independent replication of the injected fault. The public task data are mathematical questions; we introduce no human-subject annotations or synthetic task dataset.

A.2 REFERENCE SELECTION AND CALIBRATION

Before interpreting a result, specify the reference, common score scale, disagreement rule, audit cohorts, and endpoints. A repaired final-answer grader is appropriate for the demonstrated incident. For learned judges, store the judge prompt, decoding configuration, and repeated judgments or uncertainty. A changed preference objective should be reported as a changed objective, not retroactively labeled an implementation bug. A reference panel can expose sensitivity, but averaging several correlated judges does not create ground truth. These extensions are protocol recommendations, not evaluated results.

A.3 TRAINING CONFIGURATION

The incident and recovery scripts use 64 prompts per update and five rollouts per prompt, maximum prompt and response lengths of 512 and 1024 tokens, actor learning rate 10^{-6}, PPO (Schulman et al., 2017) minibatches of 32, and microbatches of two per GPU. GRPO uses a KL loss with coefficient 0.001, the `low_var_kl` form, and zero entropy coefficient. Actor parameters, optimizer state, and reference parameters use FSDP (Zhao et al., 2023) with CPU offloading. The rollout backend is vLLM (Kwon et al., 2023), tensor parallelism is one, and checkpoints are saved every ten updates. These parameters describe the available scripts and resolved C80 configuration; historical source reconstruction is not equivalent to a runtime-attested source snapshot for every legacy arm.

A.4 FIXED-PROMPT EVALUATION CONTRACT

We sample 256 indices without replacement from the 5,000-question MATH test split with NumPy's default random generator and seed 20260926, after excluding exact prompt matches to the training split. Indices are sorted before inference. The manifest stores the exact selected questions, task references, dataset hashes, and prompt strings. No reference result is consulted during selection. All saved actors are loaded from their single-rank FSDP weight files with strict parameter-key matching. Model weights are not updated.

To hold inference conditions fixed, every actor uses the same Instruct checkpoint tokenizer and chat template, a system message requesting step-by-step reasoning and a boxed final answer, and the original test prompt. This intentionally differs from reproducing each historical deployment configuration. Inference uses greedy decoding, FP16 weights, PyTorch SDPA, batches of 16, and at most 1024 new tokens. Both the Instruct end-of-turn token and end-of-text token terminate generation. Truncation is recorded explicitly. Checkpoint, script, tokenizer, grader, and output hashes accompany completed runs. The small frontend smoke run uses different settings and is excluded from every reported scientific result.

Paired bootstrap intervals (Efron & Tibshirani, 1993; Koehn, 2004; Dror et al., 2018) resample question indices with replacement, using 10,000 replicates and seed 20260926. The same resampled indices are applied to all evaluator columns and compared checkpoints. They quantify test-question variation conditional on the selected trained states; they do not quantify variation over retraining, historical seeds, or evaluator uncertainty. We do not use overlapping pointwise intervals as a substitute for an interval on the paired difference.

A.5 DEFINITIONS OF THE EVALUATOR COLUMNS

Table 3 defines the five evaluator columns used in every cross-evaluation matrix.

Table 3: Pass criteria of the evaluator columns.

Rule	Pass criterion
Reference	Final boxed answer matches the target after the existing MATH string-normalization function. No additional numeric fallback is used in fresh evaluation.
Exact box	Final extracted boxed answer equals the target after trimming surrounding whitespace; no algebraic or formatting normalization.
Box present	Literal `\boxed{` occurs in the response.
Short box	Box present and response has fewer than 400 characters.
Length pass	The continuous reward in Equation 6 is at least 0.5, equivalently box present and at most 750 characters.

The exact-box evaluator is stricter syntactically, not necessarily better aligned with mathematical correctness: equivalent fractions or formatting can fail it. Box presence, short-box, and length-pass are intentionally weak rules. High scores under them are evidence of criterion satisfaction, not independent evidence of reasoning ability. Only reference and length reward were used to train these incidents. In a true learned-evaluator sequence, the same protocol would require archived judge weights, prompts, tools, decoding settings, and repeated judgments.

B ADDITIONAL MEASUREMENT DERIVATIONS

B.1 A SYMMETRIC ALLOCATION OF EVALUATOR AND OUTPUT CHANGES

Let $M_{a,0}, M_{a,1}, M_{b,0}, M_{b,1}$ be the complete two-by-two crossed matrix. Averaging the two orders of changing output cohort and evaluator gives

$$B = \tfrac{1}{2}[(M_{b,0} - M_{a,0}) + (M_{b,1} - M_{a,1})], \tag{7}$$

$$E = \tfrac{1}{2}[(M_{a,1} - M_{a,0}) + (M_{b,1} - M_{b,0})], \tag{8}$$

$$B + E = M_{b,1} - M_{a,0}. \tag{9}$$

Cancellation proves the last equality. This is a symmetric descriptive allocation, not an identified causal decomposition of learning. It depends on the two chosen evaluators. A fixed-reference comparison instead answers which gain survives under a declared target criterion. Reporting the full matrix avoids hiding a large interaction behind either allocation.

B.2 RANKING REVERSAL VERSUS FALSE PROGRESS

For the same endpoints, a strict evaluator-dependent reversal occurs when $(M_{b,v} - M_{a,v})(M_{b,w} - M_{a,w}) < 0$. This relation is symmetric in v, w and requires no privileged reference. False progress is directional: it asserts that a positive apparent gain fails under a designated reference. A reversal alone cannot identify which evaluator is wrong. Both should be reported with a practically relevant margin and uncertainty when used for decisions rather than descriptive accounting.

B.3 WHAT AGGREGATE CURVES CANNOT IDENTIFY

Observed scores without historical outputs cannot determine how a later reference would grade those outputs: different sets of answers can receive the same old score and different new scores. Conversely, even identical reference-score curves can arise from different consumption histories. A discarded faulty reward and an optimizer-consumed faulty reward may produce the same telemetry but different ancestry. Thus measurement comparability requires paired evidence, while historical impact requires consumption information. These are information requirements, not claims that every application must use a particular storage system.

B.4 Why the diagonal is insufficient

Consider two rounds whose contemporaneous scores are 0.5 and 0.8. A compatible cross-matrix may have reference scores 0.5 and 0.8, or 0.5 and 0.2, while retaining the same contemporaneous scores. The apparent gain is 0.3 in both cases, but the reference gain changes sign. Without further assumptions linking evaluators, the diagonal cannot distinguish them. This is an illustrative construction, not an experimental result. Archived outputs permit the missing reference evaluations; a current-model evaluation cannot substitute for those historical observations.

B.5 Ancestry is not a parameter-effect claim

An update may consume a disputed score but receive zero net gradient, for example when all normalized advantages vanish. Other loss terms may still change parameters. Conversely, a signal can affect selection of an entire batch even if its own trajectory is excluded. The ancestry proposition therefore requires capturing every relevant control and data dependency, not merely direct example-to-gradient links. The conservative closure is well-defined for these records, but it does not prove damage or that removing one signal would restore an alternative parameter trajectory.

C Recovery provenance and additional results

Hybrid configuration. The primary run's early actor checkpoint has Base EOS configuration, whereas later checkpoints and recovery use Instruct configuration. An older run label that identifies the entire incident as a pure Base or pure Instruct run is insufficient. We use the term hybrid throughout. A common inference template in the new evaluation controls evaluation-time configuration; it cannot retroactively remove the training-time change.

C80 recovery audit. The affected C80 recovery completed 40 updates and 12,800 trajectories. The original post-run validator rejected a mismatch between frontend and compute-side spellings of the same scheduler job ID. A repaired validator subsequently produced a completion record. The original failure marker and revalidation note remain part of the evidence. For this paper, a separate read-only audit checks the normalized job identity, resolved configuration, checkpoint presence, raw row counts and unique IDs, input-ledger equality, and stored consumption identities. Fresh execution of the MATH grader agrees with all 12,800 raw recovery scores. The historical full checkpoint ledgers are not rehashed file by file; the new fixed-prompt evaluation separately hashes the model weights it loads. These checks do not modify the historical records. This is a retrospective audit, not a new independent training replicate, and does not establish exact parity of the legacy arms' data ordering or environment.

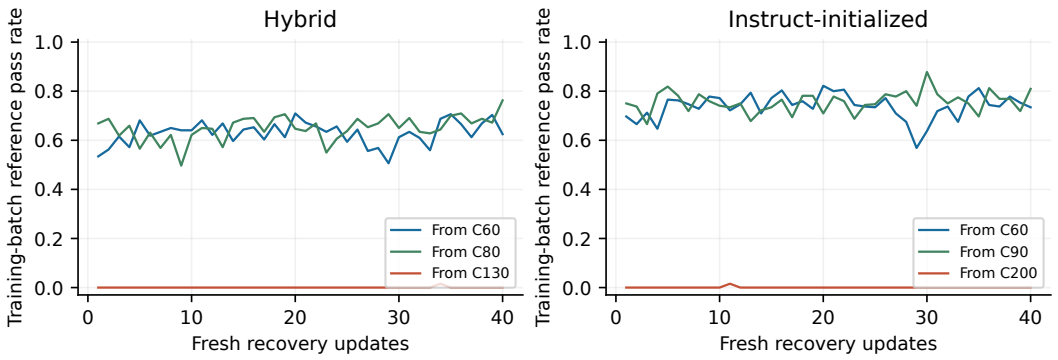


Figure 4: All recorded recovery trajectories. Each arm runs 40 fresh updates under the repaired reference. Curves measure the successive training batches; comparisons on common held-out questions are treated separately. Affected intermediate states and collapsed states exhibit different recovery behavior.

Audit-rule sensitivity. Table 4 compares the pass/fail and numeric-reward dispute predicates over all historical responses.

Incident	Responses	Pass/fail disputes	Numeric reward disputes
Hybrid	41,600	15,560	21,640
Instruct	64,000	26,535	43,808

Table 4: Pass/fail uses threshold 0.5; numeric reward disagreement uses absolute tolerance 10^{-6}. These are different predicates, even though they can produce the same conservative boundary in a sequential incident.

Operational cost. The earlier hybrid-incident audit measured a 91.49 MB trajectory-and-lineage store for 41,600 responses, compared with 70.34 MB of native rollout dumps. Its CPU diagnosis took 2.989 seconds without language-model resampling; fresh ingest took 14.51 seconds. These are inherited single-incident measurements, not timings of the new fixed-prompt evaluation. Three paired capture runs on one A100 reported a mean training-time change of -0.764% with descriptive 95% t interval $[-2.122\%, 0.595\%]$: no detected slowdown, not evidence of acceleration; the per-run timing records, launcher, and trainer instrumentation are in the supplement. These operational observations support feasibility but are not the scientific basis of the progress claims.

D HISTORICAL CROSS-EVALUATION AND EFFECTIVE LEARNING SIGNAL

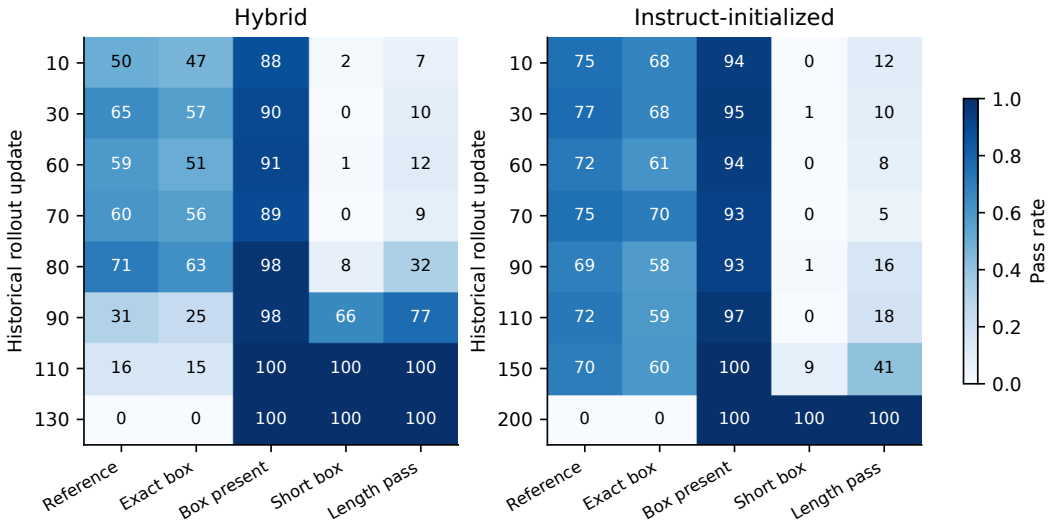


Figure 5: Cross-evaluator matrix on historical cohorts (cell labels are percentages). Rows are output cohorts, not held-out checkpoint evaluations. Reference and length-pass columns correspond to historical training evaluator versions; the three middle columns are additional diagnostic rules. A response can satisfy formatting and brevity while failing both answer references.

Table 5 counts, for every stored update, the prompt groups whose five training rewards are not all equal and hence carry a non-zero group-normalized advantage. It is computed from stored rollouts without resampling (`scripts/effective_signal.py`).

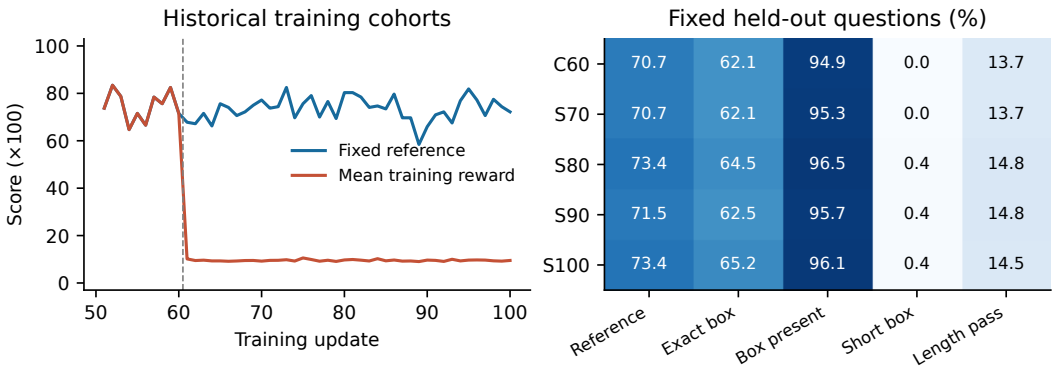


Figure 6: Thresholded short-box control. Left: historical reference pass rate and mean continuous training reward (both multiplied by 100); their numerical difference is not the pass-rate decomposition. Right: completed evaluations on the same 256 held-out questions. C60 precedes the evaluator change; St is the control checkpoint after update t. These checkpoints share a GPU backend within the control. Retained reference performance despite disputed ancestry does not establish behavioral equivalence.

Table 5: Groups with non-zero GRPO advantage. Updates with zero advantage have no such group; KL regularization and optimizer momentum may still change parameters.

Arm	Training evaluator	Updates	Live groups (%)	Updates with zero advantage
Hybrid, faulty updates 61–130	length	70	62.2	12
Hybrid, recovery from C60	reference	40	43.6	0
Hybrid, recovery from C80	reference	40	41.8	0
Hybrid, recovery from C130	reference	40	0.0	40
Instruct, faulty updates 61–200	length	140	78.4	6
Instruct, recovery from C60	reference	40	21.2	0
Instruct, recovery from C90	reference	40	20.4	0
Instruct, recovery from C200	reference	40	0.0	40
Short-box control, updates 61–100	short-box	40	15.2	0

The twelve updates with zero advantage in the hybrid faulty phase and six in the Instruct faulty phase occur after collapse: the sampled responses receive uniform proxy rewards within each group. Recovery from the collapsed states has zero reward advantage for all 40 updates in both incidents. This audit concerns the reward-driven surrogate; it does not measure parameter changes or prove that escape from collapse is impossible. Of the short-box control's 389 non-uniform groups, 376 contrast rewards 0 and 0.1 (box presence), while 13 contrast 0.1 and 1 (brevity). On held-out questions, S100 has 11 reference gains and four losses relative to C60, with 78 of 256 response texts identical. Similar aggregate accuracy can conceal changed outputs. Across all five evaluated control states the fixed-reference pass rate stays within 70.7–73.4% (C60 70.7, S70 70.7, S80 73.4, S90 71.5, S100 73.4%), and the faulty short-box rule passes at most 0.4% of responses.

An exploratory CPU check compares the first, middle, and last scalar of each stored parameter tensor before and after collapsed-state recovery. Of 1,017 probed coordinates, 853 change in the hybrid arm and 956 in the Instruct arm (`data/parameter_probe.json`). Thus stored parameters do change despite identical greedy FP16 outputs on the held-out questions. These probes neither estimate a full parameter norm nor identify the responsible loss term, and small stored changes can disappear under inference-time rounding.

E ARTIFACT ORGANIZATION

The supplementary package contains the Loom and RTD implementation, all VeRL changes as patches against public upstream VeRL, the launchers of every recorded run, the complete Loom stores of both incidents (every trajectory, signal, update, and checkpoint record), a script that regrades all stored responses and recomputes the reported ancestry closure, the analysis and evaluation scripts, the anonymized training configuration and evaluator implementations, raw completed held-out outputs, historical per-response scores, group reward records, recovery curves, the audit plan, hash records, and instructions to regenerate the figures and tables. CPU-only verification regrades every released held-out response and reconstructs historical aggregate measurements. Historical model checkpoints are too large for the supplement. Exact replay of their generation requires the original model artifacts; paired rescoring and reproduction of the included aggregate figures do not. Missing or partial generation outputs are never treated as zero scores, and checkpoint evaluation is marked complete only after every selected question has a stored response and an output hash.